PERFORMANCE COMPARISON OF HEVC intra, JPEG, JPEG 2000, JPEG XR, JPEG LS and VP9 intra

A PROJECT UNDER THE GUIDANCE OF

DR. K. R. RAO

COURSE: EE5359 - MULTIMEDIA PROCESSING, SPRING 2016

SUBMITTED BY:

SWAROOP KRISHNA RAO

swaroop.krishnarao@mavs.uta.edu

1001256012

DEPARTMENT OF ELECTRICAL ENGINEERING

UNIVERSITY OF TEXAS AT ARLINGTON
# Table of Contents:

1. Objective ........................................................................................................................................... 5

2. Overview of the Compression schemes ............................................................................................ 5

   2.1. HEVC ........................................................................................................................................... 5
      
      2.1.1. Encoder and Decoder in HEVC .......................................................................................... 5
      
      2.1.2. Features of HEVC ............................................................................................................. 7
          
          2.1.2.1. Partitioning .................................................................................................................. 7
          
          2.1.2.2. Prediction .................................................................................................................... 7
          
          2.1.2.3. Transform and Quantization ....................................................................................... 8
          
          2.1.2.4. Entropy coding ............................................................................................................ 8
      
      2.2. JPEG ......................................................................................................................................... 9
      
      2.3. JPEG2000 ................................................................................................................................ 10
      
      2.4. JPEG XR ................................................................................................................................... 11
      
      2.5. JPEG LS ................................................................................................................................... 13
      
      2.6. VP9 .......................................................................................................................................... 14
          
          2.6.1. Encoder and Decoder in VP9 .......................................................................................... 15
          
          2.6.2. Features of VP9 .............................................................................................................. 16
              
              2.6.2.1. Prediction Block Sizes ............................................................................................... 16
              
              2.6.2.2. Prediction Modes ....................................................................................................... 16
              
              2.6.2.3. Transform and quantization ....................................................................................... 17
              
              2.6.2.4. Entropy coding ............................................................................................................ 17
      
3. Comparison Metrics ........................................................................................................................... 18

   3.1 Peak Signal to Noise Ratio ........................................................................................................... 18
   
   3.2 Structural Similarity Index ......................................................................................................... 19
   
   3.3 Implementation Complexity ....................................................................................................... 19
List of Acronyms and Abbreviations:

ADST: Asymmetric Discrete Sine Transform.
AVC: Advanced Video Coding.
CABAC: Context Adaptive Binary Arithmetic Coding.
CAVLC: Context Adaptive Variable Length Coding.
CSVT: Circuits and Systems for Video Technology.
CTB: Coding Tree Block.
CTU: Coding Tree Unit.
CU: Coding Unit.
DBF: De-blocking Filter.
DCT: Discrete Cosine Transform.
DST: Discrete Sine Transform.
DPB: Decoded Picture Buffer.
EBCOT: Embedded Block Coding with Optimized Truncation.
EZW: Embedded Zero-tree Wavelet.
HD: High Definition.
HEVC: High Efficiency Video Coding.
HM: HEVC Test Model.
HP: High Profile.
ITU-T: International Telecommunication Union-Telecommunication Standardization Sector.
JCT: Joint Collaborative Team.
JCT-VC: Joint Collaborative Team on Video Coding.
JM: H.264 Test Model.
JPEG: Joint Photographic Experts Group.
JPEG-LS: JPEG Lossless.
JPEG-XR: JPEG extended range.
KTA: Key Technical Areas (H.264 based exploration software of VCEG).
LOCO: Low Complexity Lossless Compression.
MC: Motion Compensation.
ME: Motion Estimation.
MJPEG: Motion JPEG.
MPEG: Moving Picture Experts Group.
MSE: Mean Square Error.
NGOVD: Next Generation open Video.
PB: Prediction Block.
PCS: Picture Coding Symposium.
PSNR: Peak Signal to Noise Ratio.
PU: Prediction Unit.
QP: Quantization Parameter.
RD: Rate Distortion.
SAO: Sample Adaptive Offset.
SPIE: Society of Photo-Optical and Instrumentation Engineers.
SSIM: Structural Similarity Index.
TB: Transform Block.
TU: Transform Unit.
VCEG: Video Coding Experts Group.
VLC: Variable Length Coding.
1. Objective:
The objective of this project is to study coding standards HEVC [1] [26] [27] [28], JPEG [32], JPEG 2000 [34], JPEG XR [35], JPEG LS [36] and VP9 [3] [4] and understand various techniques in image coding such as prediction, transform, quantization and coding. A performance comparison of these codecs based on two metrics such as PSNR [19] and SSIM [5] [14] [24] is carried out.

2. Overview of the Compression schemes:

2.1. HEVC [1] [26]:

High Efficiency Video Coding (HEVC) [1] is an international standard for video compression developed by a working group of ISO/IEC MPEG (Moving Picture Experts Group) and ITU-T VCEG (Video Coding Experts Group) [1]. The main goal of HEVC standard is to significantly improve compression performance compared to existing standards (such as H.264/Advanced Video Coding [2]) in the range of 50% bit rate reduction at similar visual quality [1].

HEVC is designed to address existing applications of H.264/MPEG-4 AVC [2] and to focus on two key issues: increased video resolution and increased use of parallel processing architectures [1]. It primarily targets consumer applications as pixel formats are limited to 4:2:0 8-bit and 4:2:0 10-bit. The next revision of the standard, finalized in 2014, enables new use-cases with the support of additional pixel formats such as 4:2:2 and 4:4:4 and bit depth higher than 10-bit [12] as shown in Fig. 1 [15], embedded bit-stream scalability, 3D video [11] and multiview video [30].

2.1.1. Encoder and Decoder in HEVC [13]:

Source video, consisting of a sequence of video frames, is encoded or compressed by a video encoder to create a compressed video bit stream. The compressed bit stream is stored or transmitted. A video decoder decompresses the bit stream to create a sequence of decoded frames. The video encoder performs the following steps:

- Partitioning each picture into multiple units
• Predicting each unit using inter or intra prediction, and subtracting the prediction from the unit
• Transforming and quantizing the residual (the difference between the original picture unit and the prediction)
• Entropy encoding transform output, prediction information, mode information and headers

The video decoder performs the following steps:
• Entropy decoding and extracting the elements of the coded sequence
• Rescaling and inverting the transform stage
• Predicting each unit and adding the prediction to the output of the inverse transform
• Reconstructing a decoded video image

Figs. 2 [11] and 3 [15] represent the detailed block diagrams of HEVC encoder and decoder respectively:

![Figure 2: Block Diagram of HEVC Encoder [11]](image-url)
2.1.2. Features of HEVC:

2.1.2.1. Partitioning [13]:
HEVC supports highly flexible partitioning of a video sequence. Each frame of the sequence is split up into rectangular or square regions (units or blocks) [13], each of which is predicted from previously coded data. After prediction, any residual information is transformed and entropy encoded.

Each coded video frame, or picture, is partitioned into tiles and/or slices, which are further partitioned into coding tree units (CTUs). The CTU is the basic unit of coding, analogous to the macro-block in earlier standards, and can be up to 64x64 pixels in size.

A coding tree unit can be subdivided into square regions known as coding units (CUs) using a quad-tree structure. Each CU is predicted using inter or intra prediction and transformed using one or more transform units, as shown in Fig.4 [13].

Figure 3: Block diagram of HEVC Decoder [15]
2.1.2.2. Prediction [1]:
Frames of video are coded using intra or inter prediction:

Intra prediction: Each PU is predicted from neighboring image data in the same picture, using DC prediction (an average value for the PU), planar prediction (fitting a plane surface to the PU) or directional prediction (extrapolating from neighboring data), as shown in Fig.5 [1].

Inter prediction: Each PU is predicted from image data in one or two reference pictures (before or after the current picture in display order), using motion compensated prediction.

Figure 5: Modes and angular intra prediction directions in HEVC [1]
2.1.2.3. Transform and Quantization [13]:
Any residual data remaining after prediction is transformed using a block transform based on the Discrete Cosine Transform (DCT) [9]. Only 4x4 luma are applied to residual data in each CU as shown in Fig. 6 [13].

![Figure 6: CTU showing range of transform (TU) sizes [13]](image)

2.1.2.4. Entropy coding:
A coded HEVC bit stream consists of quantized transform coefficients, prediction information such as prediction modes and motion vectors, partitioning information and other header data. All of these elements are encoded using Context Adaptive Binary Arithmetic Coding (CABAC) [10] similar to H.264/AVC [3].

2.2. JPEG:
JPEG is the first ISO/ITU-T standard for continuous tone still images [32]. It allows lossy and lossless coding of still images. JPEG gives good compression results for lossy compression with the least complexity. There are several modes defined for JPEG including baseline, progressive and hierarchical. The baseline mode, which supports lossy compression alone, is most popular. Average compression ratio of 15:1 is achieved using lossy coding with the help of DCT-block based compression. Lossless coding is made possible with predictive coding compression techniques which include differential coding, run length coding and Huffman coding. JPEG employs quantization with HVS weighting. Zig-zag scanning is performed on quantized coefficients since it allows entropy coding to be performed in the order from low frequency to high frequency coefficients as shown in Fig.7(a) and Fig.7(b) [32].
The process flow of JPEG baseline (lossy) algorithm is shown in the Figs. 8 and 9 [32].

The process flow starts with the color conversion for color images followed by 8x8 block based DCT (process flow starts here for gray scale images), quantization, zig-zag scanning, and entropy coding using Huffman tables in the encoding process and vice versa for decoding process.
Different quantization matrices are used for luminance and chrominance components. Quality factor ‘Q’ is set using quantization tables and different kinds of artifacts in varied ranges are observed.

2.3. JPEG2000:

JPEG 2000 [34] is image compression standard which supports lossy and lossless compression of gray scale or color images. In addition to the compression capability, JPEG 2000 supports excellent low bit rate performance without sacrificing the performance at high bit rate, region of interest coding, EBCOT (Embedded Block Coding with Optimized Truncation) [34] which overcomes the limitations of EZW (embedded zero-tree wavelet coding) [34] which are random access to specific regions of the image, error resilience. It also supports flexible file format and progressive decoding of the image to allow from lossless to lossy by fidelity and resolution. It is a transform based framework, uses wavelet based decomposition. Wavelet transform has 3dB improvement over DCT based compression. Lossless compression is the result of transform, entropy coding. Consider non-scalable, single layer mode since scalability feature leads to adverse effect on rate-distortion performance. Also we disable tiling mode because it also lowers rate-distortion performance. Tiling allows the image be partitioned into non-overlapped rectangular tiles to be encoded independently. The encoder and decoder of JPEG 2000 is shown in Fig.10 [34]. The tiling, DC level shifting, color transformation, Discrete Wavelet Transform (DWT) of each image component is shown in Fig.11 [34].

![JPEG 2000 Encoder Diagram](image1)

![JPEG 2000 Decoder Diagram](image2)

Figure 10(a): Block diagram of JPEG 2000 encoder [34]

Figure 10(b): Block diagram of JPEG 2000 decoder [34]
2.4. JPEG XR:

JPEG XR [35], a coded file format is designed mainly for storage of continuous-tone photographic content. It supports wide range of color formats including n-channel encodings using fixed and floating point numerical representations, bit depth varieties giving a way for wide range of data compression scenarios. The ultimate goal is to support wide range of color encodings, maintain forward compatibility with existing formats and keep device implementation simple. It also aims at providing same algorithm for lossless as well as lossy compression.

HD photo format [35] is a new file format standardized using JPEG-XR. Just like JPEG-2000, Microsoft HD photo works on advanced features like lossy-lossless compression, bit-rate scalability, editing, region-of-interest decoding, integer implementation without division etc. on top of compression capability. HD photo minimizes objectionable spatial artifacts preserving high frequency detail and outperforms other lossy compression technologies in this regard.

HD photo is a block-based image coder similar to traditional image-coding paradigm: color conversion, transform, coefficient scanning, scalar quantization and entropy coding. Main blocks of HD photo include transformation stage and the coefficient-encoding stage. HD photo employs a reversible integer-to-integer-mapping lapped bi-orthogonal transform (LBT) [35] as its decorrelation engine. The reversible property of the algorithm supports both lossy and lossless compression. Thus, it simplifies the overall implementation of the system. HD photo’s encoder contains many adaptive elements: adaptive coefficient scanning, flexible quantization, inter-block coefficient prediction, adaptive VLC table switching, etc. JPEG XR supports a number of advanced pixel formats in order to avoid limitations and complexities of conversions between different unsigned integer representations. This feature allows flexible approach to numerical encoding of image data. This results in low-complexity implementations in the encoder and decoder.
The encoder and decoder block diagrams of JPEG XR are shown in the Figs. 12 and 13 [38], respectively.

Figure 12: Block diagram of JPEG XR encoder [38]

Figure 13: Block diagram of JPEG XR decoder [38]

2.5. JPEG LS [36]:

Hewlett Packard proposed a simpler predictive coder for low complexity [36]. LOCO-I (Low Complexity Lossless Compression for Images) [36] is a lossless compression algorithm for continuous-tone images which combines the simplicity of Huffman coding with the compression potential of context models. Lossless image compression schemes often consist of two distinct and independent components: modeling and coding. The modeling part can be formulated as an inductive inference problem, in which an image is observed pixel by pixel in some pre-defined order (e.g. raster-scan). The block diagram of JPEG-LS is shown in Fig.14 [37].
2.6. VP9 [3][4]:

VP9 is an open and royalty free video compression standard being developed by Google. VP9 had earlier development names of Next Generation Open Video (NGOV) and VP-Next [3] [4]. VP9 is a successor to VP8. Development of VP9 started in Q3 2011. One of the goals of VP9 is to reduce the bit rate by 50% compared to VP8 while having the same video quality [16]. Also VP9 aims to improve it to the point where it would have better compression efficiency than High Efficiency Video Coding. VP9 expands techniques used in H.264/AVC and VP8 and is very likely to replace AVC at least in the YouTube video service [17]. Google is working on a new compression algorithm, an open-source codec dubbed VP10 [3], which will reduce the bit rate by 50 percent compared to VP9.

2.6.1. Encoder and Decoder in VP9 [18]:

A large part of the advances made by VP9 over its predecessors is natural progression from current generation video codecs to the next. Figs. 15 and 16 represent block diagrams of encoder and decoder of VP9 respectively [18].

![Block diagram of JPEG-LS](image-url)
Figure 15: Encoder block diagram for VP9 [18]

Figure 16: Decoder block diagram for VP9 [18]
2.6.2. Features of VP9:

2.6.2.1. Prediction Block Sizes:
A large part of the coding efficiency improvements achieved in VP9 can be attributed to incorporation of larger prediction block sizes [4] [17]. VP9 introduces super-blocks (SB) of size up to 64x64 and allows breakdown using recursive decomposition all the way down to 4x4.

Each sub-block may be further split into prediction blocks and transform blocks. Intra-prediction in VP9 is still performed on square regions thus rectangular prediction blocks represent two square prediction blocks with the same prediction mode.

Giving an analogy to HEVC [1], prediction spreading 2N x 2N, N x N, 2N x N or N x 2N is available where 2N x 2N is the size of the block being split. It is worth mentioning that 4x4 prediction blocks are determined within corresponding 8x8 blocks as a group, unlike other prediction sizes when prediction data is stored per each prediction block. Like in HEVC, a sub-block can be split into transform blocks in a quad-tree structure down to the smallest 4x4 block. The allowed sizes are 32x32, 32x16, 16x16, 8x16, 8x8 and 4x4.

Fig. 17 shows an example of partitioning of a 64x64 Super-block [4] [17].

![Figure 17: Example partitioning of a 64x64 Super-block in VP9](image)

2.6.2.2. Prediction Modes:

**Intra-prediction Modes [4]:**
VP9 supports a set of 10 Intra prediction modes for block sizes ranging from 4x4 up to 32x32: DC_PRED (DC prediction), TM_PRED (True-motion prediction), H_PRED (Horizontal prediction), V_PRED (Vertical prediction), and 6 oblique directional prediction modes: D27, D153, D135, D117, D63, D45 corresponding approximately to angles 27, 153, 135, 117, 63, and 45 degrees (counter-clockwise measured against the horizontal axis). The horizontal, vertical and oblique directional prediction modes involve copying (or estimating) pixel values from
surrounding blocks into the current block along the angle specified by the prediction mode. Fig.18 shows angular Intra-prediction modes in VP9 [4].

**Inter Prediction Modes [4]:**

VP9 supports a set of 4 inter prediction modes for block sizes ranging from 4x4 up to 64x64 pixels: NEARESTMV, NEARMV, ZEROMV, and NEWMV.

**2.6.2.3. Transform and quantization [4]:**

The residuals after subtraction of predicted pixel values are subjected to transformation and quantization. Transform blocks can be 32x32, 16x16, 8x8 or 4x4 pixels. Like most other coding standards, these transforms are an integer approximation of the DCT [9].

For intra coded blocks either or both the vertical and horizontal transform pass can be DST (discrete sine transform) instead. This is with respect to the specific characteristics of the residual signal of intra blocks. In addition, VP9 introduces support for a new transform type, the Asymmetric Discrete Sine Transform (ADST), which can be used in combination with specific intra-prediction modes. Intra-prediction modes that predict from a left edge can use the 1-D ADST in the horizontal direction, combined with a 1-D DCT in the vertical direction.

Similarly, the residual signal resulting from intra-prediction modes that predict from the top edge can employ a vertical 1-D ADST transform combined with a horizontal 1-D DCT transform. Intra-prediction modes that predict from both edges such as the True Motion mode and some diagonal intra-prediction modes use the 1-D ADST in both horizontal and vertical directions.

**2.6.2.4. Entropy coding:**

VP9 uses 8-bit arithmetic coding engine from VP8 known as bool-coder [4]. Unlike AVC or HEVC, the probabilities of VP9 bool-coder do not change adaptively within a frame. VP9 makes use of forward context updates through the use of flags in the frame header that signal modifications of the coding contexts at the start of each frame. These probabilities are stored in
what is known as a frame context. The decoder maintains four of these contexts, and each frame specifies which one to use in bitstream.

3. Comparison Metrics:

3.1. Peak Signal to Noise Ratio [19]:
Peak signal-to-noise ratio (PSNR) [19] is an expression for the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation. Because many signals have a very wide dynamic range (ratio between the largest and smallest possible values of a changeable quantity), the PSNR is usually expressed in terms of the logarithmic dB scale.

PSNR is most commonly used to measure the quality of reconstruction of lossy compression codecs. The signal in this case is the original data, and the noise is the error introduced by compression. When comparing compression codecs, PSNR is an approximation to human perception of reconstruction quality. Although a higher PSNR generally indicates that the reconstruction is of higher quality, in some cases it may not. One has to be extremely careful with the range of validity of this metric; it is only conclusively valid when it is used to compare results from the same codec (or codec type) and same content.

PSNR is defined via the mean squared error (MSE). Given a noise-free m x n monochrome image f and its noisy approximation g, MSE is defined as:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (f(i,j) - g(i,j))^2$$

f represents the data of original image
g represents the data of degraded image
m represents the numbers of rows of pixels of the images and i represents the index of that row
n represents the number of columns of pixels of the image and j represents the index of that column

The PSNR in dB is defined as:

$$PSNR = 20 \log_{10} \left( \frac{MAX_f}{\sqrt{MSE}} \right)$$

MAX_f is the maximum signal value that exists in the original image

The formula used to calculate Y’CBCr PSNR for 4:2:0 format is

$$PSNR_{Y'CBCr} = \frac{6 \times PSNR_{Y'} + PSNR_{Cb} + PSNR_{Cr}}{8}$$
3.2. Structural Similarity Index [5][14][24]:
The structural similarity index is a method for measuring the similarity between two images. The SSIM index is a full reference metric; in other words, the measuring of image quality based on an initial uncompressed or distortion-free image as reference. SSIM is designed to improve on traditional methods like peak signal-to-noise ratio (PSNR) and mean squared error (MSE), which have proven to be inconsistent with human eye perception.
The difference with respect to other techniques such as MSE or PSNR is that these approaches estimate perceived errors; on the other hand, SSIM considers image degradation as perceived change in structural information. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene.
The SSIM metric is calculated on various windows of an image. The measure between two windows x and y of common size N×N is:

\[
SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)}
\]

Where
- \(\mu_x\) the average of x;
- \(\mu_y\) the average of y;
- \(\sigma_x^2\) the variance of x;
- \(\sigma_y^2\) the variance of y;
- \(\sigma_{xy}\) the covariance of x and y;
- \(c_1 = (k_1L)^2\), \(c_2 = (k_2L)^2\) two variables to stabilize the division with weak denominator;
- \(L\) the dynamic range of the pixel-values (typically this is \(2^{\text{#bits per pixel}} - 1\));
- \(k_1=0.01\), \(k_2=0.03\) by default.

3.3. Implementation Complexity:
The computational time of HEVC intra, JPEG, JPEG 2000, JPEG XR, JPEG LS and VP9 intra encoders will be compared and this serves as an indication of implementation complexity.

4. Profiles used for comparison:
Test sequences in 4:2:0 YUV sampling format are encoded using HM 16.9 [20][25], JPEG baseline reference software [39], JasPer (version 1.900.1) [40], HD Photo Device Porting Kit (version 1.0) [41], HP LOCO-I [42] and VPX encoder from the WebM Project [21] test models for HEVC, JPEG, JPEG 2000, JPEG XR, JPEG LS and VP9 respectively, are used for comparison in this project.
5. **Test Sequences:**
The following sequences [22] are used to study and compare the codecs (shown in Figs. 19-22).

Figure 19: Johnny_1280x720_60.yuv [22]

Figure 20: Jockey_1920x1080.yuv [22]
6. Test Platform:
The following properties are used
Processor: Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz
Installed Memory (RAM): 6.00 GB
System Type: 64-bit operating system, x-64 based processor
7. Configuration of Tools:

7.1. Configuration of HM 16.9:
This section describes configuration settings and command line parameters required for encoding a video test sequence in HM 16.9 in intra mode.

Main all-intra profile settings:

IntraPeriod: 1 # Period of I-Frame (-1 = only first)
GOPSize: 1 # GOP Size (number of B slice = GOPSize-1)
QP: 32 # Quantization parameter (0-51)

Command line parameters for using HM 16.9 encoder:

TAppEncoder [-h] [-c config.cfg] [--parameter=value]
where,
-h Prints parameter usage
-c Defines configuration file to use. Multiple configuration files may be used with repeated –c options.
--parameter=value Assigns value to a given parameter.

7.2. Configuration of JPEG:
The command line arguments for JPEG-baseline software [39] are as follows:
Input image can be in pnm or ppm
Encoder: cjpeg --quality N inputfile.pnm outputfile.jpg
where quality factor N denotes the scale quantization tables to adjust image quality. Quality factor varies from 0 (worst) to 100 (best); default is 75.
Decoder: djpeg --outfile outputfilename.pnm --fileformat inputfile.jpg

7.3. Configuration of JPEG 2000:
The command line arguments for JPEG-2000 software [40] are as follows:
Encoder:
jasper --input inputfilename.bmp --output outputfilename.jp2 --output-format jp2 --O rate=0.01
where rate specify target rate as a positive real number. ‘rate’=1 corresponds to no compression.

7.4. Configuration of JPEG XR:
For Microsoft HD Photo [41], all options are set to their default values with the only control coming from the quality factor setting:
- No tiling
- One-level of overlap in the transformation stage
- No color space sub-sampling
- Spatial bit-stream order
- All sub-bands are included without any skipping
The command line arguments for JPEG XR software [41] are as follows:
**Encoder:** wmpencapp –i input.bmp –o output.wdp –q [ ]

where, quality factor ‘q’ leads to lowering of PSNR resulting in lossy compression. q=0 is the case of lossless compression and wmpencapp command line converts certain uncompressed file formats into equivalent HD photo files.

**Decoder:** wmpdecapp –i input.wdp –o output.bmp –c [ ]

where wmpdecapp command line converts HD photo files to different uncompressed file formats

### 7.5. Configuration of JPEG LS:

The settings for JPEG-LS software [42] are as follows at the encoder. Decoder settings need not be changed from default as they follow the encoder settings.

- Images should be in ppm or pgm format.
- Line interleaved mode is considered in the project.
- Error value is varied from 1 to 60. Error value of zero corresponds to no compression.
- T1, T2, T3 are thresholds. While giving the settings the following condition need to be met. Error value+1<T1<T2<T3.
- Default RESET value of 64 is considered in the project

**Encoder:** locoe [flags] [-i infile] [-o outfile]

**Decoder:** locod [flags] [-i infile] [-o outfile]

### 7.6. Configuration of VP9:

This section describes command line parameters required for encoding a video test sequence in VP9.

**Sample command line parameters for VP9:**

```
swaroop@swaroop-VirtualBox:~/libvpx-pub/libvpx/build/linuxbuild$ vpxenc PeopleOnStreet_2560x1600_30_crop.yuv -o pos.22.webm --codec=vp9 --i420 --width=2560 --height=1600 --passes=50 -t 0 --good --cpu-used=0 --end-usage=q --limit=1 --fps=30000/1001 --verbose --psnr --lag-in-frames=25 --kf-max-dist=1 --min-q=32 --max-q=32
```
8. Results:
Figs. 23 – 30 illustrate the YUV PSNR Vs Frame Index and SSIM Vs Frame Index for test sequences Johnny_1280x720_60.yuv, Jockey_1920x1080.yuv, Bosphorus_3840x2160.yuv and PeopleOnStreet_2560_1600_30_crop.yuv.

![Figure 23: PSNR Vs Frame Index plot for Johnny_1280x720_60.yuv](image1)

![Figure 24: SSIM Vs Frame Index plot for Johnny_1280x720_60.yuv](image2)
Figure 25: PSNR Vs Frame Index plot for Jockey_1920x1080.yuv

Figure 26: SSIM Vs Frame Index plot for Jockey_1920x1080.yuv
Figure 27: PSNR Vs Frame Index plot for PeopleOnStreet_2560_1600_30_crop.yuv

Figure 28: SSIM Vs Frame Index plot for PeopleOnStreet_2560_1600_30_crop.yuv
Figure 29: PSNR Vs Frame Index plot for Bosphorus_3840x2160.yuv

Figure 30: SSIM Vs Frame Index plot for Bosphorus_3840x2160.yuv
Fig. 31 shows the rank of the codecs.

![Codec Rank](image)

Figure 31: Rank of the codecs

Figs. 32 – 35 illustrates the Encoding time (sec) for the test sequences.

![Comparison of Encoding Time for Johnny_1280x720_60.yuv](image)

Figure 32: Comparison of encoding time (secs) for Johnny_1280x720_60.yuv
Figure 33: Comparison of encoding time (secs) for Jockey_1920x1080.yuv

Figure 34: Comparison of encoding time (secs) for PeopleOnStreet_2560_1600_30_crop.yuv
9. Conclusions:
In this project, compression performance was analyzed for HEVC Intra, JPEG, JPEG 2000, JPEG XR, JPEG LS and VP9 Intra. Both objective and subjective methodologies for the comparison of various still image coding techniques were used. While PSNR gives an objective quality metrics of image, SSIM gives subjective quality which takes human visual perception (HVS) into account. From the results shown in Figs. 23-30, the conclusion that can be drawn is that the performance of HEVC Intra frame for 4:2:0 image is better compared to all other codecs. VP9 Intra coding performs better than JPEG, JPEG 2000, JPEG LS and JPEG XR. JPEG produced worst quality images when compared to other codecs. From Figs. 32-35, it can be seen that the encoding time for HEVC is significantly greater than other codecs and indicates that HEVC encoder is more complex.
**Future work:**

JPEG XT [48] is a new standardization effort which targets the extension of the JPEG features mainly by providing backward and forward compatibility with the JPEG legacy format. It also enables support for High Dynamic Range Imaging, lossless and near-lossless coding, and alpha channel coding.

Google is working on a new compression algorithm, an open-source codec dubbed VP10 [3], which will reduce the bit rate by 50 percent compared to VP9. It is also anticipated that VP10 will bring a gamut of enhancements when compared to its predecessor VP9, such as speedier frame rates, improved dynamic range for shadows and highlights, as well as a bigger color range.

Therefore, as an extension to this project one can also compare JPEG XT and VP10.

**ACKNOWLEDGEMENT**

I would sincerely like to thank Dr. K. R. Rao for his constant support and guidance throughout the duration of my project.

I would also like to thank Tuan Ho and MPL members for assisting me in my project.
References:


[21] Chromium® open-source browser project, VP9 source code, Online: http://git.chromium.org/gitweb/?p=webm/libvpx.git;a=tree;f=vp9;hb=aaf61dfbcab414bfacc3171501be17d191ff8506


[38] JPEG-XR encoder and decoder block diagrams: http://www.microsoft.com/whdc/xps/wmphotoeula.mspx


This tutorial is for personal use only.

Password: a2FazmgNK

https://datacloud.hhi.fraunhofer.de/owncloud/public.php?service=files&t=8edc97d26d46d4458a9c1a17964bf881