Final Report on

Fast Intra Coding Based on Reference Samples Similarity in HEVC

A PROJECT UNDER THE GUIDANCE OF

DR. K. R. RAO

COURSE: EE5359 - MULTIMEDIA PROCESSING, SPRING 2016

SUBMITTED BY:

Nikita Thakur (1001102923)

DEPARTMENT OF ELECTRICAL ENGINEERING

UNIVERSITY OF TEXAS AT ARLINGTON
List of Acronyms and Abbreviations:

AHG: Ad Hoc Group.
AVC: Advanced Video Coding.
ANRS:
CABAC: Context Adaptive Binary Arithmetic Coding.
CTU: Coding Tree Unit.
CTB: Coding Tree Block.
CU: Coding Unit.
DCT: Discrete Cosine Transform.
DST: Discrete Sine Transform.
DVD: Digital Video Disk.
HD: High Definition.
HDR: High Dynamic Range.
HEVC: High Efficiency Video Coding.
HM: HEVC Test Model.
ITU: International Telecommunication Union.
JCT: Joint Collaborative Team.
JCT-VC: Joint Collaborative Team on Video Coding.
JPEG: Joint Photographic Experts Group.
KTA: Key Technical Areas
MPEG: Moving Picture Experts Group.
MPM: Most Probable Mode
MVC: Multiview Video Coding.
PB: Prediction Block.
PU: Prediction Unit
RD: Rate Distortion
RMD:
RDO: Rate Distortion Optimization
SATD: Sum of Absolute Transformed Differences
SCC: Screen Content Coding.
TB: Transform Block.
TU: Transform Unit.
UHD: Ultra High Definition
VCEG: Visual Coding Experts Group
WCG: Wide Color Gamu
1. Introduction:

Digital Video Coding is used to compress the information so it can be stored or transmitted occupying minimal space. This process is done in a video encoder and the main goal is to optimize coding efficiency. Coding efficiency is the ability to minimize the bit rate necessary for representation of video content while maintaining a given level of video quality. The two previous video coding standards, the H.262/MPEG-2 Video [3] and H.264/MPEG-4 AVC (Advanced Video Coding) [11], have had a significant influence in a variety of products that are frequently present in our daily lives. Nevertheless, the growing popularity of HD video, the emergence of beyond-HD formats (4K and 8K), the need of higher resolution 3D or multiview, and the increased desire for higher quality and resolutions in mobile applications, are generating emergency needs for coding efficiency superior to H.264/MPEG-4 AVC’s capabilities. In addition, more than 50% of the current network traffic is video targeted to mobile devices and tablet-PCs; this growth in traffic and the transmission needs for video-on-demand services, are imposing harsh challenges in today’s networks.

For these reasons and mainly due to the increased need for higher compression of video content, the H.265/HEVC (High Efficiency Video Coding) [1] standard has been developed. This Recommendation addresses all existing applications of H.264/MPEG-4 AVC and also many new applications such as those aforementioned. It has been designed to pay special attention to three key issues: increased video resolution, ease of integration of the transport system and increased use of parallel processing architectures. Thanks to many new improvements and an optimized coding scheme, the HEVC standardization effort is capable to improve the compression efficiency by reportedly reducing by an average 50% the bit-rate for the same objective and perceptual video quality with respect to its predecessor H.264/MPEG-4 AVC [2]. This comes at the cost of a generally much higher computational complexity with respect to previous standards.
2. Encoder and Decoder in HEVC [21]:
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 as shown in Fig. 2.1:
- 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 as shown in Fig. 2.2:
- 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

Figure 2.1: Block Diagram of HEVC Encoder [21]
3. Picture Partitioning [3]:

The previous standards split the pictures in block-shaped regions called Macroblocks and Blocks. Nowadays we have high-resolution video content, so the use of larger blocks is advantageous for encoding. To support this wide variety of blocks sizes in efficient manner HEVC pictures are divided into so-called coding tree Units (CTUs) as shown in Fig.3.1[3]. Depending on the stream parameters, the CTUs in a video sequence can have the size: 64×64, 32×32, or 16×16 as shown in Fig.3.2[3].

Coding Tree Unit (CTU) is therefore a coding logical unit, which is in turn encoded into an HEVC bit-stream. It consists of three blocks, namely luma (Y), that covers a square picture area of LxL samples of the luma component, and two chroma components (Cb and Cr), that cover (L/2)x(L/2) samples of each of the two chroma components, and associated syntax elements, as shown in Fig.3.3. Each block is called Coding Tree Block (CTB).
Syntax elements describe properties of different types of units of a coded block of pixels and how the video sequence can be reconstructed at the decoder. This includes the method of prediction (e.g. inter or intra prediction, intra prediction mode, and motion vectors) and other parameters [9].

Each CTB has the same size (LxL) as the CTU (64×64, 32×32, or 16×16). However, CTB could be too big to decide whether we should perform inter-picture prediction or intra-picture prediction. So each CTB can be split recursively in a quad-tree structure, from the same size as CTB to as small as 8×8. Each block resulting from this partitioning is called Coding Block (CB) as shown in Fig.3.4 and becomes the decision making point of prediction type (inter or intra prediction) [12]. Fig.3.4 illustrates an example of 64×64 CTBs split into CBs.

The prediction type along with other parameters is coded in Coding Unit (CU). So CU is the basic unit of prediction in HEVC, each of which is predicted from previously coded data. And the CU consists of three CBs (Y, C_b and C_r), and associated syntax elements, as shown in Fig.3.5 [12].
CBs could still be too large to store motion vectors (inter-picture (temporal) prediction) or intra-picture (spatial) prediction mode. Therefore, Prediction Block (PB) was introduced. Each CB can be split into PBs differently depending on the temporal and/or spatial predictability.

4. Prediction:
Frames of video are coded using intra or inter prediction:

Intra-frame prediction:
In the spatial domain, the redundancy means that pixels (samples) that are close to one another in the same frame or field are usually highly correlated. This means that the appearance of samples in an image is often similar to their adjacent neighbor samples; this is called the spatial redundancy or intra-frame correlation. This redundant information in the spatial domain can be exploited to compress the image. Note that when using this kind of compression, each picture is compressed without referring to other pictures in the video sequence. This technique is called Intra-frame prediction and it is designed to minimize the duplication of data in each picture (spatial-domain redundancy) [7]. It consists in forming a prediction frame and subtracting this prediction from the current frame.
Several methods can be used to remove this redundant information in the spatial domain. Typically, the values of the prediction samples are constructed by combining their adjacent neighbor samples (reference samples) by means of several techniques. In some cases, considerable prediction accuracy can be obtained by means of efficient intra-prediction techniques.

**Inter frame prediction:**
In the temporal domain, redundancy means that successive frames in time order are usually highly correlated; therefore, parts of the scene are repeated in time with little or no changes. This type of redundancy is called temporal redundancy or inter-frame correlation [13].

It is clear then that the video can be represented more efficiently by coding only the changes in the video content, rather than coding each entire picture repeatedly. This technique is called Inter-frame prediction; it is designed to minimize the temporal-domain redundancy and at the same time improve coding efficiency to achieve video compression [7].

To remove the redundant information in the temporal domain typically motion compensated prediction or inter prediction methods are used. Motion compensation (MC) consists of
constructing a prediction of the current video frame from one or more previous or future encoded frames (reference frames) by compensating differences between the current frame and the reference frame. To achieve this, the motion or trajectory between successive blocks of the image is estimated. The information regarding motion vectors (describes how the motion was compensated) and residuals from the previous frames are coded and sent to the decoder.

5. Intra Prediction:

The Intra-picture prediction uses the previously decoded boundary samples from spatially neighboring TBs in order to predict a new prediction block PB. So the first picture of a video sequence and the first picture at each clean random access point into a video sequence are coded using only intra-picture prediction [1].

Several improvements have been introduced in HEVC in the intra prediction module:

- Due to the larger size of the pictures, the range of supported coding block sizes has been increased.
- A plane mode that guarantees continuity at block boundaries is desired.
- The number of directional orientations has been increased.
- For intra mode coding, efficient coding techniques to transmit the mode for each block are needed due to the increased number of intra modes.
- HEVC supports a large variety of block sizes, so it needs consistency across all block sizes.

HEVC employs 35 different intra modes to predict a PB: 33 Angular modes, Planar mode, and DC mode. Table. 1 shows the mode name with their corresponding intra prediction mode index as by the convention used throughout the standard [11].

<table>
<thead>
<tr>
<th>Mode Name</th>
<th>Intra Prediction Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planar</td>
<td>0</td>
</tr>
<tr>
<td>DC</td>
<td>1</td>
</tr>
<tr>
<td>Angular</td>
<td>2...34</td>
</tr>
</tbody>
</table>

Table 1: Specification of Intra Prediction modes and associated index [11].

At the end, the video encoder will choose the intra prediction mode that provides the best Rate-Distortion performance.
5.1. PB partitioning:
The CB can be split into size of MxM or M/2xM/2, as shown in Fig. 5.1. The first one means that the CB is not split, so the PB has the same size as the CB. It is possible to use it in all CUs. The second partitioning means that the CB is split into four equally-sized PBs. This can only be used in the smallest 8x8 CUs. In this case, a flag is used to select which partitioning is used in the CU. Each resulting PB has its own intra prediction mode [1]. The prediction blocks size range from 4x4 to 64x64.

![Figure 5.1: Prediction Block for Intra Prediction [1]](image)

5.2. Reference Samples:

5.2.1. Reference Sample Generation:
The intra sample prediction in HEVC is performed by extrapolating sample values from the reconstructed reference samples as defined by the selected intra prediction mode. Compared to the H.264/AVC, HEVC introduces a reference sample substitution process which allows HEVC to use the complete set of intra prediction modes regardless of the availability of the neighboring reference samples. In addition, there is an adaptive filtering process that can pre-filter the reference samples according to the intra prediction mode, block size and directionality to increase the diversity of the available predictors.

5.2.2. Reference Sample Substitution:
Some or all of the reference samples may not be available for prediction due to several reasons. For example, samples outside of the picture, slice or tile are considered unavailable for prediction. In addition, when constrained intra prediction is enabled, reference samples belonging to inter-predicted PUs are omitted in order to avoid error propagation from potentially erroneously received and reconstructed prior pictures. As opposed to H.264/AVC which allows only DC
prediction to be used in these cases, HEVC allows the use of all its prediction modes after substituting the non-available reference samples.

For the extreme case with none of the reference samples available, all the reference samples are substituted by a nominal average sample value for a given bit depth (e.g., 128 for 8-bit data). If there is at least one reference sample marked as available for intra prediction, the unavailable reference samples are substituted by using the available reference samples. The unavailable reference samples are substituted by scanning the reference samples in clock-wise direction and using the latest available sample value for the unavailable ones. More specifically, the process is defined as follows:

1. When $p[-1][2N-1]$ is not available, it is substituted by the first encountered available reference sample when scanning the samples in the order of $p[-1][2N-2], \ldots , p[-1][-1]$, followed by $p[0][-1], \ldots , p[2N-1][-1]$.
2. All non-available reference samples of $p[-1][y]$ with $yD2N -2 : : : -1$ are substituted by the reference sample below $p[-1][y+1]$.
3. All non-available reference samples of $p[x][-1]$ with $xD0 : : : 2N -1$ are substituted by the reference sample left $p[x-1][-1]$.

Figure 5.2 shows an example of reference sample substitution.

![Reference Sample Substitution Process](image)

Figure 5.2: An example of reference sample substitution process. Non-available reference samples are marked as grey: (a) reference samples before the substitution process (b) reference samples after the substitution process [3]

### 5.3. Filtering Process of Reference Samples:

The reference samples used by HEVC intra prediction are conditionally filtered by a smoothing filter similarly to what was done in 8x8 intra prediction of H.264/AVC. The intention of this processing is to improve visual appearance of the prediction block by avoiding steps in the values of reference samples that could potentially generate unwanted directional edges to the prediction block. For the optimal usage of the smoothing filter, the decision to apply the filter is done based on the selected intra prediction mode and size of the prediction block.

- Two types of filtering process of reference samples
• Fig.5.3 shows a three-tap filtering using two neighboring reference samples. Reference sample X is replaced by the filtered value using A, X and B while reference sample Y is replaced by the filtered value using C, Y and D.

• Fig.5.3 shows a strong intra smoothing process using corner reference samples. Reference sample X is replaced by a linearly filtered value using A and B while Y is replaced by a linearly filtered value using B and C [3].

5.4. Angular Prediction:
Angular intra prediction in HEVC is designed to efficiently model different directional structures typically present in video and image content. The set of available prediction directions has been selected to provide a good trade-off between encoding complexity and coding efficiency for typical video material. The sample prediction process itself is designed to have low computational requirements and to be consistent across different block sizes and prediction directions. This has been found especially important as the number of block sizes and prediction directions supported by HEVC intra coding far exceeds those of previous video codecs, such as H.264/AVC. In HEVC there are four effective intra prediction block sizes ranging from 4x4 to 32x32 samples, each of which supports 33 distinct prediction directions.

Angle Definitions:
HEVC defines a set of 33 angular prediction directions at 1/32 sample accuracy as illustrated in Fig. 15. In natural imagery, horizontal and vertical patterns typically occur more frequently than patterns with other directionalities. Small differences for displacement parameters for modes close to horizontal and vertical directions take advantage of that phenomenon and provide more accurate prediction for nearly horizontal and vertical patterns. The displacement parameter differences
become larger closer to diagonal directions to reduce the density of prediction modes for less frequently occurring patterns.

Table 2 provides the exact mapping from indicated intra prediction mode to angular parameter $A$ [3]. That parameter defines the angularity of the selected prediction mode (how many 1/32 sample grid units each row of samples is displaced with respect to the previous row).

<table>
<thead>
<tr>
<th>Mode</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>
<th>16</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>32</td>
<td>26</td>
<td>21</td>
<td>17</td>
<td>13</td>
<td>9</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>-5</td>
<td>-9</td>
<td>-13</td>
<td>-17</td>
<td>-21</td>
<td>-26</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mode</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
<th>31</th>
<th>32</th>
<th>33</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>-32</td>
<td>-26</td>
<td>-21</td>
<td>-17</td>
<td>-13</td>
<td>-9</td>
<td>-5</td>
<td>-2</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>13</td>
<td>17</td>
<td>21</td>
<td>26</td>
</tr>
</tbody>
</table>

Table 2: Angular parameter $A$ defines the directionality of each angular intra prediction mode [3]

![Figure 5.4](image)

Figure 5.4: Angle definitions of angular intra prediction in HEVC numbered from 2 to 34 and the associated displacement parameters. $H$ and $V$ are used to indicate the horizontal and vertical directionality, respectively, while the numeric part of the identifier refers to the sample position displacements in 1/32 fractions of sample grid positions [3]

For each octant, eight angles are defined with associated displacement parameters, as shown in Fig.5.4. When getting closer to diagonal directions, the displacement parameter become larger in order to reduce the density of prediction modes for less occurring directions. For modes close to horizontal and vertical directions, the displacement becomes smaller in order to provide more accurate prediction for nearly horizontal and vertical patterns.

In order to calculate the value of each sample of the PB, the angular mode extrapolates the samples from the reference samples, depending on the directional orientation in order to achieve lower
complexity. When the direction selected is between 2 and 17, the samples located in the above row (red samples and may be green samples, as shown in Fig. 5.5 (a)) are projected as additional samples located in the left column, extending the left reference column. When the direction selected is between 18 and 34, the samples located at the left column (blue samples and may be orange samples, as shown in Fig. 5.5 (b)) are projected as samples located in the above row, extending the top reference row. In both cases the samples projected would have negative indexes [3].

![Diagram](image.png)

Figure 5.5: Example of diagonal orientation [3]

5.4.1. Sample Prediction for Angular Prediction Mode [3]: Predicted sample values p[x][y] are obtained by projecting the location of the sample p[x][y] to the reference sample array applying the selected prediction direction and interpolating a value for the sample at 1/32 sample position accuracy [3].

Prediction for horizontal modes (modes 2-17) is given by:
Sample Prediction for vertical modes (modes 18–34) is given by:

\[ p[x][y] = (32 - f) \cdot \text{ref}[y + i + 1] + f \cdot \text{ref}[y + i + 2] + 16 \gg 5 \]

where \( i \) is the projected integer displacement on row \( y \) (for vertical modes) or column \( x \) (for horizontal modes) and calculated as a function of angular parameter \( A \) as follows:

\[ i = ((x + 1) \cdot A) \gg 5, \text{ for horizontal modes} \]
\[ i = ((y + 1) \cdot A) \gg 5, \text{ for vertical modes} \]

\( f \) represents the fractional part of the projected displacement on the same row or column and is calculated as:

\[ f = ((x + 1) \cdot A) \& 31, \text{ for horizontal modes} \]
\[ f = ((y + 1) \cdot A) \& 31, \text{ for vertical modes} \]

5.5. DC Prediction:
This mode is also similar to the DC mode in H.264/MPEG-4 AVC. It is efficient to predict plane areas of smoothly-varying content in the image, but gives a coarse prediction on the content of higher frequency components and as such it is not efficient for finely textured areas. The value of each sample of the PB is an average of the reference samples. As explained before, for this case the reference samples will be the boundary samples of the top and left neighboring TBs.

5.6. Planar Prediction:
This mode in HEVC is similar to the planar mode in H.264/MPEG-4 AVC, and is known as mode 0. In H.264/MPEG-4 AVC this method is a plane prediction mode for textured images, and may introduce discontinuities along the block boundaries. Conversely, in HEVC this mode was improved in order to preserve continuities along the block edges.

Planar mode is essentially defined as an average value of two linear predictions using four corner reference samples. This mode is implemented as follows, with reference to Fig.5.6; the sample \( X \)
is the first sample predicted as an average of the samples D and E, then the right column samples (blue samples) are predicted using bilinear interpolation between samples in D and X, and the bottom row samples (orange samples) are predicted using bilinear interpolation between samples in E and X. The remaining samples are predicted as the averages of bilinear interpolations between boundaries samples and previously coded samples [14].

![Planar intra prediction mode](image)

**Figure 5.6: Planar intra prediction mode [14]**

### 5.7. Smoothing Filter:
HEVC uses a smoothing filter in order to reduce the discontinuities introduced by the intra-prediction modes. This is applied to the boundary samples, namely the first prediction row and column for DC mode, or the first prediction row for pure horizontal prediction, or the first prediction column for pure vertical prediction. The smoothing filter consists of a two-tap finite impulse response filter for DC prediction or a gradient-based smoothing filter for horizontal (mode 10) and vertical (mode 26) prediction. Due to the fact that chroma components tend to be already smooth, this filter is not used in this case. Prediction boundary smoothing is only applied to luma component.

The smoothing filter is applied to the reference samples depending on the size of the blocks and the directionalities of the prediction. Thanks to using this filter, contouring artefacts caused by boundaries in the reference samples may be drastically reduced.

### 6. Intra Mode Coding:

The process of Intra coding is shown in Fig. 6.1. In the rough mode decision (RMD) stage, the current block is predicted with mode 0 firstly. Then the differences between the original block and the prediction are obtained.

Afterward Hadamard transform is applied to the difference block, and finally the sum of the absolute transformed difference (SATD) is calculated. After the process is repeated for all the 35 intra modes, the N modes with the minimum SATD are selected to be candidates
for RDO. N is determined according to PU sizes as eight for PUs of size 4×4 and 8×8, and three for PUs of other sizes. The MPM will also be added in the N modes if they are not included yet.

Fig 6.1: Process of Intra coding
8. Fast Intra Coding Based on Reference Samples Similarity in HEVC Algorithm [36]:

![Diagram of sample prediction](image)

Fig 7.1: An example of sample prediction for angular prediction Modes [36].

It can be seen from Fig. 7.2 that not all the samples in the reference array are needed for the prediction. The array of needed reference samples (ANRS) varies with different angular prediction modes. If the ANRS are similar, then they would generate similar predictions. So, we don’t have to apply the RMD and RDO to all the modes which have similar ANRSs. Only one of them is needed. Even though other modes are discarded, it will not result in too much degradation of quality and BD-rate.

![Diagram of block and reference samples](image)

Fig. 7.2. A block and the reference samples [36]
Fig. 7.2 shows an original block and the reference samples. Fig. 7.3 shows the ANRSs for the block. It can be seen that the ANRSs in the red rectangle are similar. These similar ANRSs would generate similar prediction. Only one of the angular modes needs to be evaluated in the RMD stage. In the proposed algorithm, only the first one (intra prediction mode 25) is kept and others are discarded.

Mean and variance are used to judge whether two ANRSs are similar. If the means of two ANRSs are close enough and the variances of both ANRSs are small enough, the two ANRSs are similar. The condition is defined as follows [36]:

\[
\text{Mean}(\text{ANRS}[i]) - \text{Mean}(\text{ANRS}[j]) < \text{MeanThreshold}
\]

\[
\&\& \text{Variance(ANRS}[i]) < \text{VarianceThreshold} \quad (11)
\]

\[
\&\& \text{Variance(ANRS}[i]) < \text{VarianceThreshold}
\]

where ANRS[i] means the ANRS for intra mode i (i > 1 and i < 35).
6.1 DC and Planar Prediction [36]
The cases of DC and planar mode are different from angular modes. For DC mode, the ANRS used in prediction is shown in the red rectangles in Fig. 5.3. Only the samples which are on the left and above of the block are calculated for mean value. The variance always equals 0 in this case, since the prediction is determined by mean only.
For the planar mode, the ANRS used in prediction is shown in the red rectangles in Fig. 5.4. The samples in the bottom row are copies of the bottom-left reference sample and the samples in the right column are copies of top-right reference sample. The mean and variance are calculated for the samples in the red rectangles.

6.2 The Process of Skipping Intra Modes [36]
As is shown in Fig. 5.5, the proposed algorithm is based on rough mode decision (RMD). The algorithm processes each mode from mode 0 to mode 34. Before predicting a block using a certain mode, the mean and variance of the ANRS are calculated. If the ANRS of the current mode is
Similar to that of a preceding mode, the current mode is discarded. So, the Hadamard cost is calculated only for a subset of all the intra modes. The encoding time is thereby reduced.

1. Fig. 7.6. The flowchart of Fast Intra Coding Based on Reference Samples Similarity in HEVC Algorithm [36]
8. Implementation of the Test sequences:

Fig.8.1 Command window display running HM code for BQMall test sequence
Fig. 8.2 Command window display with output for BQ Mall test sequence
Fig. 8.3 Command window display running Hm code for Racehorses test sequence
Fig. 8.4 Command window display showing output for Racehorses test sequence
Fig. 8.5 Command window display running Hm code for Kimono test sequence
Fig. 8.6 Command window display showing output for Kimono test sequence
9. Algorithm Implementation in the HM code

The two thresholds are defined in TComPrediction.h and used in the function "TComPrediction::xPredIntraAngWithStatistics" in TComPrediction.cpp. This is the function where my algorithm is mostly implemented. This function is used in the function TComPrediction::predIntraAngWithStatistics which is used in the function TEncSearch::estIntraPredLumaQT.
/// Scale the mean and variance of rank.
float fmean = 0;
int fsize = 0;
float fvariance = 0;
int fu4variance = 0;
int fu8variance = 0;

if (intra4angled == 0) {
    for (int x = 0; x < width; ++x)
        fmean += rank[x] / fsize;
    fvariance += pow(rank[x] / fsize - fmean, 2); //fvariance = variance / float(width);
    fvariance = fvariance / float(width);
    fmean = fmean;
} else if (intra4anged > 0) {
    // The maximum index used in rank is used when x == width - 1 and y == height - 1;
    int delta = intra4anged * height;
    int delta2 = delta << 1;
}
10. Experiments and Results

The experiment is conducted in “All Intra – main”. Quantization parameter values 22, 27, 32 and 37 are used for each video sequence. The Fast Intra Coding Based on Reference Samples Similarity in HEVC Algorithm is implemented in HM 16.5, and the performance of the Fast Intra Coding Based on Reference Samples Similarity in HEVC Algorithm is compared with original HM 16.5. To compare the performance, Bjøntegaard Delta bit rate (BD rate) and Bjøntegaard’s Delta PSNR-Y (BDPSNR-Y) are calculated, and the encoding time change is given by:

\[
\text{Time Change} = \frac{T(\text{Original}) - T(\text{Proposed})}{T(\text{Original})} \times 100
\]

Where T (Proposed) means the encoding time of the Fast Intra Coding Based on Reference Samples Similarity in HEVC Algorithm and T (Original) is the encoding time of original HM16.5 [24]. The programs are performed on 64-bit Windows 8 operating system with Intel(R) Core(TM) i7- CPU and 12 GB RAM.

To compare the performances of the Fast Intra Coding Based on Reference Samples Similarity in HEVC Algorithm the test sequences in Table 3 are tested. Table 4,5,6 and 7 show the experimental result of the Fast Intra Coding Based on Reference Samples Similarity in HEVC Algorithm, when the mean threshold equals 3 and the variance threshold equals 100 and it is compared with respect to the bitrate PSNR and Time change.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Test Sequence</th>
<th># of frames</th>
<th>fps</th>
</tr>
</thead>
<tbody>
<tr>
<td>832x480</td>
<td>BQMall</td>
<td>100</td>
<td>60</td>
</tr>
<tr>
<td>1280x720</td>
<td>KristenAndSara</td>
<td>100</td>
<td>60</td>
</tr>
<tr>
<td>1920x1080</td>
<td>Kimono</td>
<td>100</td>
<td>60</td>
</tr>
</tbody>
</table>
10.1 Encoder Complexity Reduction

The following test results (figures 10.1 to 10.3) show the difference in encoding time of the original HM16.5 [24] and the Fast Intra Coding Based on Reference Samples Similarity in HEVC Algorithm for different quantization parameter (QP) values.

**BQMall-100 Frames**

**Mean Threshold =3 and variance Threshold=100**

![Graph showing encoding time vs. quantization parameter for BQMall]

Figure 10.1.1 Encoding time vs. quantization parameter for BQMall
Mean Threshold=2 Variance Threshold=100

Figure 10.1.2 Encoding time vs. quantization parameter for BQMall

Mean Threshold =1 and Variance Threshold= 100
Figure 10.1.3 Encoding time vs. quantization parameter for BQMall

**BQMall-100 Frames**

Figure 10.1.4 Encoding time vs. quantization parameter for KristenandSara

**KristenAndSara-100 Frames**

Figure 10.1.5 Encoding time vs. quantization parameter for Kimono

**Kimono-100 Frames**
10.2 BD-PSNR

To objectively evaluate the coding efficiency of video codecs, Bjøntegaard Delta PSNR (BD-PSNR) was proposed [38]. Based on the rate-distortion (R-D) curve fitting, BD-PSNR is able to provide a good evaluation of the R-D performance [36]. BD-PSNR is a curve fitting metric based on rate and distortion of the video sequence. However, this does not take into account the complexity of the encoder, but the BD metric tells a lot about the quality of the video sequence [30] [31]. Ideally, BD-PSNR should increase and BD-bitrate should decrease. The following results show a plot of BD-PSNR versus the quantization parameter (QP). It can be observed from figures 10.2.1 to 10.2.3 that there is a slight drop in PSNR using BD metrics for the Fast Intra Coding Based on Reference Samples Similarity in HEVC Algorithm for the HM16.5 [24] in the range of 0.09dB to 0.5 dB
Fig. 10.2.1 BD-PSNR vs. quantization parameter for BQMall
**Kimono-100 Frames**

Fig.10.2.2 BD-PSNR vs. quantization parameter for Kimono

**KristenAndSara-100 Frames**

Fig.10.2.3 BD-PSNR vs. quantization parameter for KristenandSara

10.3 BD-bitrate

BD-bitrate is a metric similar to the BD-PSNR metric which determines the quality of encoded video sequence. It can be observed from figures 10.3.1 to 10.3.3 that there is a slight increase in

**BQMall-100 Frames**

![BQMall-100 Frames graph](image)

Fig.10.3.1 BD-bitrate vs. quantization parameter for BQMall

**Kimono-100 Frames**

![Kimono-100 Frames graph](image)

Fig.10.3.2 BD-bitrate vs. quantization parameter for Kimono

**KristenAndSara-100 Frames**

![KristenAndSara-100 Frames graph](image)
Fig. 10.3.3 BD-bitrate vs. quantization parameter for KristenandSara

10.4 Percentage Decrease in Encoding Time

Figures 10.4.1 to 10.4.3 show 13-30% decrease in encoding time which shows the decrease in the complexity of the encoder with Fast Intra Coding Based on Reference Samples Similarity in HEVC Algorithm as compared to the original HM16.5 algorithm [24].

**BQMall-100 Frames**

Fig. 10. 4.1 % improvement in encoding time vs. quantization parameter for BQMall
Kimono-100 Frames

Fig. 10.4.2 % improvement in encoding time vs. quantization parameter for Kimono

KristenAndSara-100 Frames

Fig. 10.4.3 % improvement in encoding time vs. quantization parameter for KristenAndSara
11. Conclusions
The results of comparative experiments demonstrate that the Fast Intra Coding Based on Reference Samples Similarity in HEVC Algorithm can effectively reduce the computational complexity (encoding time) by 14-30% on average as compared to the HM 16.5 encoder [24], while only incurring a slight drop in the PSNR and a slight increase in the bitrate for different values of the quantization parameter. The experimental results show that when the mean threshold equals 3 and the variance threshold equals 100, 14-30% encoding time can be reduced on average with acceptable loss of quality.

ACKNOWLEDGEMENT
I would sincerely like to thank Dr. K. R. Rao and Tuan Ho for their constant support and guidance throughout the duration of my project.
12. References:


[23] HM C++ Code: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/


[27] VCEG & JCT documents available from http://wftp3.itu.int/av-arch in the video-site and jvt-site folders


[38] Tutorials:


This tutorial is for personal use only. Password: a2FazmgNK

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

Tut8. Please find the links to YouTube videos on the tutorial - HEVC/H.265 Video Coding
Standard including the Range Extensions Scalable Extensions and Multiview Extensions below:
https://www.youtube.com/watch?v=TLNkK5C1KN8

[39] Detailed Overview of HEVC/H.265 by Shevach Riabtsev
https://app.box.com/s/rxxzr1al1nh7709yvih