EE5359 PROJECT PROPOSAL

ANALYSIS OF ADAPTIVE GOP ALGORITHMS FOR EFFICIENT HEVC COMPRESSION

Abhijith Jagannath
ID: 1000874642
Objective:

The objective of this project is to analyze the advantages of adoptive GOP structure on the HEVC. The analysis will be carried out and different performance metrics like Compression Ratio, MSE, PSNR, BD Rate, SSIM and video quality will be evaluated for high resolution videos at various bitrates.

Introduction:

The HEVC is the latest video standard. It outperforms the other standards, providing the best video compression. For broadcasting video services, a group of pictures (GOP) with a proper intra frame insertion should be designed for practical considerations. However, the fixed GOP size (FGS) will use more bits to encode the intra frame and the inter frames, which consist of scene changes. To improve the coding performance, GOP size should be adapted properly such that intra frame can be encoded in better locations.

HEVC:

High Efficiency Video Coding (HEVC) standard is the most recent joint video standardization project of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG), currently under development in a collaboration known as the Joint Collaborative Team on Video Coding (JCT-VC). The first edition of the HEVC standard has been finalized in January 2013, resulting in an aligned text that will be published by both ITU-T and ISO/IEC. In ISO/IEC, the HEVC standard will become MPEG-H Part 2 and in ITU-T it is likely to become ITU-T Recommendation H.265 [1].

The video coding layer of HEVC employs the same “hybrid” approach (inter-/intra-picture prediction and 2D transform coding) used in all video compression standards since H.261. Figure 1.1 depicts the block diagram of a hybrid video encoder, which can create a bit stream conforming to the HEVC standard. An encoding algorithm producing an HEVC compliant bit
stream would typically proceed as follows. Each picture is split into block-shaped regions, with the exact block partitioning being conveyed to the decoder. The first picture of a video sequence is coded using only intra-picture prediction. For all remaining pictures of a sequence or between random access points, inter-picture temporally-predictive coding modes are typically used for most blocks. 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. The encoder and decoder generate identical inter prediction signals by applying motion compensation (MC) using the MV and mode decision data, which are transmitted as side information.

Fig 1.1 - HEVC Encoder Block Diagram [1]

The residual signal of the intra or inter prediction, which is the difference between the original block and its prediction, is transformed by a linear spatial transform. The transform coefficients are then scaled, quantized,
entropy coded and transmitted together with the prediction information. The encoder duplicates the decoder processing loop such that both will generate identical predictions for subsequent data. Therefore, the quantized transform coefficients are constructed by inverse scaling and are then inverse transformed to duplicate the decoded approximation of the residual signal. The residual is then added to the prediction, and the result of that addition may then be fed into one or two loop filters to smooth out artifacts induced by the block-wise processing and quantization. The final picture representation (which is the duplicate of the output of the decoder) is stored in a decoded picture buffer (DPB) to be used for the prediction of subsequent pictures. In general, the order of the 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 (bit stream order) and the output order (display order) for a decoder.

**GOP Structures:**

Generally the classic GOP structures used in most of the encoders are as shown in the figure below.

![Classical GOP Structure](image)

Fig 1.2 – Classical GOP Structure [2]
Fig 1.3 - Hierarchical GOP structure [2]

Fig 1.4 - IPPP... with multiple reference pictures [2]
**Adaptive GOP:**

The encoders mostly use fixed group of pictures (GOP) size to encode video sequences. The GOP size can achieve different values, but once target size for GOP is selected, it is applied to whole coded sequence. While fixed GOP structures are easy to implement, they prevent encoders from adapting to temporal variations in video sequences and thus prevent encoders from improving coding efficiency by electing the frame type of each frame adaptively. The transitions between shots are the regions, where static GOP structures achieved poor performance. Generally, if the video frames with smaller video content variance are coded as intra frames, we will waste a lot of bits in video coding. Conversely, if two shots changed and frames are coded using inter frames, it will also become inefficient. This can be solved by using adaptive GOP structure with positioning I frames to the places of shot changes.

Adaptive GOP structure (AGS) [5] is a new technique that can be used for enhancing the coding performance of the scalable extension of HEVC. The AGS scheme adaptively changes the sizes of GOP structure according to the temporal characteristics of a video sequence to improve the coding efficiency.

**Methods Proposed:**

1. **Adaptive GOP Based on Motion Information**
   
   Generally, if the video frames with smaller video content variance (VCV) [5] are coded as intra frames, we will waste a lot of bits in video coding. Conversely, if two scene changed frames are coded using inter frames, it will also become inefficient. With FGS, the coding performance of HEVC in the above two cases will not be effective.
Usually, the VCV of two consecutive frames could be evaluated by computing the total difference between the two frames. Intuitively, the VCV can also be expressed using the motion vectors obtained after motion estimation. If the motion vector is zero, the video is actually a still image. Conceptually, small motion information means small VCV. In this paper, we utilize the motion information to design an AGD algorithm. The motion information will not introduce any computation during mode decision. The sum of the absolute motion vectors (SAMV) in the current frame in a 4X4 blocks size after normalization is expressed using

\[ SAMV = \sum_{i=0}^{N-1} \sum_{k=0}^{M-1} \left[ \sqrt{X_{sms_x}} + \sqrt{Y_{sms_y}} \right] \times \rho_k, \]  

where \( N \) is the total number of macro-blocks in a frame, \( M \) is the total number of sub-macro-blocks in each macro-block, \( X \) and \( Y \) are the horizontal and vertical components of the motion vector after motion estimation, and the subscripts, \( sms_x \) and \( sms_y \) denote the width (x) and height (y) of the \( k^{th} \) sub-macroblock size (sms), respectively. In (1), \( \rho_k \), which is treated as normalization factor for none 4X4 sub-macroblocks, is given by

\[ \rho_k = \frac{sms_x \times sms_y}{16} \]
2. SCD Based on Motion and Residual Information

To improve the coding performance, Scene Change Detection (SCD) is very important for many entertainment videos. Many advanced video applications also require the SCD algorithm to discriminate video content. The proposed AGD should further combine the SCD to avoid the inter frame coding of scene-changed frames in some movie sequences. In this section, we also use the motion information to achieve an efficient SCD method. With the existing motion information, again, the SCD method will not take any extra computation. Here, we suggest that the SCD needs to monitor the variation in $SAMV_t$ and $SAMV_{t-1}$ and detect the discrepancy of motion residuals at the $t^{th}$ and $(t-1)^{th}$ frames. If the $t^{th}$ frame has scene change, we observe that $SAMV_t$ will extremely small than $SAMV_{t-1}$ since the scene change frames are uncorrelated such that the most of the macro-blocks will be coded with intra coding modes. At the same time, the scene change will result in large sum of absolute transformed differences ($SATD$). Hence, the SCD method is proposed that could simply perform the motion VCV ratio (MVR) test depicted by

$$\frac{SAMV_t + \epsilon}{SAMV_{t-1} + \epsilon} \leq sp_1$$

(3)

and the motion residual ratio (MRR) test given by

$$\frac{SATD_t}{SATD_{t-1}} \geq sp_2$$

(4)

where $sp_1$ and $sp_2$, which are statistical parameters are set to 0.15 and 1.3 respectively. The first VCV ratio test will detect most of the scene changes. In (4), we need to add a small constant $\epsilon$ to avoid singularity problem since the smaller VCV will lead to the $SAMV_t = 0$. However, the true still image sequence may have small $SAMV_t$. To avoid the static case, the motion
residual ratio test can robust the SAMVt method in successfully detect the scene change.

**Scope of this Project:**

- In this project the validity of adaptive GOP structure will be analised over HEVC encoder.
- The analysis will be done by implementing one of the algorithms in the latest HM-9.2 code base [7].
- The Performance of HEVC with Adaptive GOP and different profile specific GOP structures will be compared using performance metrics like Compression Ratio, MSE, PSNR, BD Rate, SSIM and video quality will be evaluated for high resolution videos at various bitrates.
Acronyms

AGS: Adaptive GOP Structure
B-Frame: Bidirectionally Interpolated Frame
CAVLC: Context Adaptive Variable Length Coding
CIF: Common Intermediate Format
DIP: Direct Intra Prediction
DPB: Decoded Picture Buffer
FGS: Fixed GOP size
HEVC: High Efficiency Video Coding
I-Frame: Intra Frame
ITU-T: International Telecommunication Union
MSE: Mean Square Error
MRR: Motion Residual Ratio
PSNR: Peak Signal to Noise Ratio
QCIF: Quarter Common Intermediate Format
SMPTE: Society of Motion Picture and Television Engineers
SSIM: Structural Similarity Index
SCD: Scene Change Detection
SATD: Sum of Absolute Transform Differences
SAMV: Sum of Absolute Motion Vectors
VCV: Video Content Variance
References:


7. HM Software:

   https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-9.2-dev/