COMPLEXITY REDUCTION FOR HEVC INTRAFRAME LUMA MODE DECISION USING IMAGE STATISTICS AND NEURAL NETWORKS.

DILIP PRASANNA KUMAR

1000786997

UNDER GUIDANCE OF DR. RAO

UNIVERSITY OF TEXAS AT ARLINGTON.

DEPT. OF ELECTRICAL ENGINEERING
1. **INTRODUCTION**

HEVC or the High Efficiency Video Coding standard is a new standard being developed as a joint project by ITU-T VCEG and ISO/IEC MPEG, working together in a partnership known as the Joint Collaborative Team on Video Coding (JCT-VC) [1]. It has been designed to address all existing applications of H.264/MPEG-4 AVC and to particularly focus on two key issues: increased video resolution and increased use of parallel processing architecture [2].

HEVC aims to achieve 50% compression over H.264 while maintaining a similar visual quality [2]. The compression is achieved at the cost of increased encoder complexity. It supports visually lossless compression and lossless compression, pictures from QVGA (320x240) to ultraHD (8k x 4k). It allows random access, fast channel switching, “trick modes” and also intra only coding modes.

HEVC employs a traditional hybrid coding model employing temporal and spatial predictions, spatial transforms, quantization, entropy coding and in loop filtering. The block diagram for a typical HEVC encoder is shown in figure 1.

![HEVC encoder block diagram](image)
HEVC encoder is functionally similar to the older video coding standards such as H.264 and AVC. The notable differences are the new coding tree unit instead of the macroblock, new in loop filtering techniques and a single entropy coding method—Context Adaptive Binary Arithmetic Coding (CABAC). However, modifications have been made to almost all aspects of the encoder and these differences contribute to the increased compression achieved by HEVC [1].

2. CODING TREE UNIT IN HEVC

The significant change is the new coding tree unit instead of a macroblock. In principle, the quadtree coding structure in HEVC is described by means of “blocks” and “units”. A block defines an array of samples and sizes thereof whereas a unit encapsulates one luma and corresponding chroma blocks together with syntax needed to code these. Consequently, a Coding Tree Unit (CTU) includes coding tree blocks (CTB) and syntax specifying coding data and further subdivision. This subdivision results in coding unit (CU) leaves with coding blocks (CB). Each CU incorporates more entities for the purpose of prediction, so called prediction units (PU), and of transform, so called transform units (TU). Similarly, each CB is split into prediction blocks (PB) and transform blocks (TB) [3]. Figure 2 shows the coding units for a frame from the sequence Traffic.

Figure 2: HEVC decoder block diagram. [12]
The coding units are traversed in a Z-scan order as shown in figure 4. The sizes of the luma CTB can be chosen as 16x16, 32x32 or 64x64 samples [2]. The larger sizes typically enable better compression [4].

![HEVC Z-scan order for traversing the coding units. Figure adapted from the documentation accompanying the HM 8.0 source code [1][2].](image-url)
HEVC introduces 33 directional modes and a planar and a DC mode for intra prediction for luma samples [Fig 5]. The angles are intentionally designed to provide denser coverage for near-horizontal and near-vertical angles and coarser coverage for near-diagonal angles [1]. The angles are ±0, ±2, ±5, ±9, ±17, ±21, ±26 and ±32 in degrees, from the horizontal and vertical directions. In addition to this, HEVC supports two alternative prediction methods, planar and DC, to target regions which have strong directional edges.

The prediction process of the Intra_Angular modes can involve extrapolating samples from the projected reference sample location according to a given directionality. To remove sample-by-sample switching between the reference row and column buffers, all extrapolations in a PB refer to a single reference row or column depending on the mode number [1]. The reference samples used for intra prediction are sometimes filtered by a 3 tap smoothing filter \([1 2 1]/4\) smoothing filter, in a manner similar to what was used for 8×8 intra prediction in H.264/MPEG-4 AVC. However, HEVC applies this smoothing operation more adaptively according to the directionality and the block size. As in H.264/MPEG-4 AVC [2], the smoothing filter is not applied for 4×4 blocks. For 8×8 blocks, only the diagonal directions 2, 18, or 34, use the reference sample smoothing. For 16×16 blocks, the reference samples are filtered for most directions except the near-horizontal and near-vertical directions, (directions in the range of 9 to 11 and 25 to 27). For 32×32 blocks, all directions except the exactly-horizontal and exactly-vertical directions use the smoothing filter.
The Intra Planar mode also uses the smoothing filter when the block size is equal or greater than 8×8, and the smoothing is not used (or useful) for the Intra DC case. HEVC applies this smoothing operation adaptively according to the directionality and block size. [2].

Figure 6: Reference samples used with mode 29. [1]
Intra prediction in HEVC is incredibly complex because of the large number of possible modes. This is further complicated by the quad tree structure: the encoder must decide both the quad tree depth and the prediction direction for each coding unit. A brute force search for the best intra mode will consist of evaluating 735 modes per LCU\(^1\).

![Recursive Coding Tree Search](image)

**Figure 7:** The recursive coding tree search employed by the HM encoder. [2].

Evaluating each mode consists of generating a prediction, subtracting the prediction from the original image and obtaining the residual. This is then used to compute the cost for rate distortion optimization.

The reference software HM8.0 employs this brute force method, and it has been shown that in the all intra mode, a quarter of the total time is spent on rate-distortion optimization and an additional 16% for intra prediction [3].

The problem being considered is to evaluate a method to reduce the number of modes that need to be evaluated for each coding unit and generate data regarding its performance. It is hoped that similar attempts by other researchers will generate sufficient data so that the best method can eventually be determined by comparing such results of all proposed methods.

---

\(^1\) 35 modes + \((4 \times (35 \text{ modes} + (4 \times 35 \text{ modes}))\) = 735 modes
5. **HYPOTHESIS**

The hypothesis is that the optimum coding unit size, and hence the quad tree depth, can be determined as a function of the variance of the pixel values within a coding unit and the correlation to the reference pixels. Also, the statistics of the most frequently used tree depth around the neighboring CUs can be helpful in determining the order in which the possible modes can be tested.

Further, once the coding tree depth has been determined by analyzing the statistics of the image, the intra mode direction within each prediction unit can be viewed as classification based on image features. Thus an artificial neural network classifier can be trained to identify the most probable intra mode direction. [13]

6. **PROPOSED WORK**

To verify the hypothesis, the following work is proposed:

1. Obtain a large set of sample pixels and corresponding intra mode decisions by modifying the HM 8.0 software [1][2], to log the relevant data and then compressing the test sequences using the modified encoder.

2. Develop a method to interpret the logged data to determine the image statistics. Use this to obtain the statistics for the data samples.

3. Search for a relation between the image statistics and the code tree depth. Specifically, determine if the variance within an image, and the correlation to the reference pixels can be used to obtain the quad tree depth.

4. Develop a MATLAB code to determine the quad tree depth using the developed method using variance and correlation and use it to determine quad tree depths with reference samples. Compare the results with the results of brute force search employed by HM 8.0

5. Train a neural network to identify the most possible intra mode direction. Collect the training data required from the modified HM 8.0 encoder.

6. Develop a MATLAB code to use the neural network to identify the intra mode for reference samples. Compare these with the results of HM 8.0 encoder.

7. Study the implementation complexity and verify that the proposed method is less complex than the brute force method. Also, compare the results with other complexity reduction techniques [6][7][8].

8. Implement the above model in HM 8.0 and measure the gain in encoding time. Also, measure the increase in BD-bitrate and BD-PSNR [15][16][17]. Compare the results with the results achieved by other techniques [6][7][8].

9. Use the data to conclude if the proposed solution is a feasible solution to the complexity of the HEVC encoder. Determine the limitations and possible improvements to the proposed solution.
REFERENCES


Available at http://wfpt3.itu.int/av-arch/video-site/0906_LG/VCEG-AL23.zip

