EFFECTIVE CODING UNIT SIZE DECISION BASED ON MOTION HOMOGENEITY CLASSIFICATION FOR HEVC

Thanuja Mallikarachchi, Anil Fernando and Hemantha Kodikara Arachchi

Centre for Vision Speech and Signal Processing
University of Surrey
United Kingdom

ABSTRACT

Determining the best partitioning structure for a given Coding Tree Unit (CTU) is one of the most time consuming operations within the HEVC encoder. The brute force search through quad tree hierarchy has a significant impact on the encoding time especially on high definition (HD) videos. This paper presents a fast coding unit size decision-taking algorithm for inter prediction in HEVC. The proposed algorithm uses a motion homogeneity based classification approach utilizing RD cost as a feature vector. Simulation results show that the proposed algorithm achieves an average of 73.25% encoding time efficiency improvement with similar rate distortion performance compared to HEVC HM12.0 reference software.

Index Terms— Video Coding, HEVC, Inter Coding, CU Size, Optimization

1. INTRODUCTION

Cisco's Data Traffic forecast statistics show that 80-90% of the global Internet traffic will be video data by 2017, and a significant proportion of the above percentage will be high definition content [1]. Hence improved video compression techniques are required in order to handle this large volume of video data that will dominate consumer networks. High Efficiency Video Coding (HEVC), which was standardized in early 2013, intends to cater these upcoming video compression requirements with its added features and improved efficiency. HEVC is the latest video coding standard produced by Joint Collaborative Team on Video Coding (JCT-VC). It is a partnership between two prominent international organizations specifying video coding standards, namely ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG). [2]

This work was supported by EU FP7 project ACTION-TV (http://www.action-tv.net/), funded under European Commission's 7th Framework Program (Grant number: 611761).

While inheriting most of the features and methodologies from its predecessors, HEVC introduces a number of new features, which improve the coding efficiency. Similar to H.264/AVC, block based prediction and compression is the baseline for HEVC. However, a wider range of block sizes has been introduced [2]. In the main profile of HEVC, a Coding Tree Unit (CTU) is partitioned into multiple coding units of sizes ranging from 8×8 to 64×64. This flexible quad tree based partitioning structure in the standard is a main contributor for its improved rate-distortion performance [3]. Fig.1 shows partitioning of a 64×64 CTU into multiple Coding Units (CUs). A CU can have multiple prediction units (PU) and transform units (TU), which are used to maintain prediction and transform information respectively.

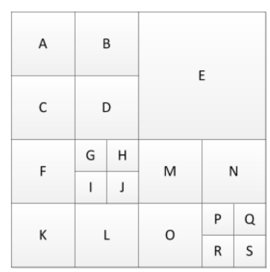

Fig. 1. An example partition structure of a 64×64 CTU.

HEVC supports multiple PU sizes that enhance inter and intra prediction coding efficiency. Fig.2 illustrates PU sizes that are supported in the inter prediction. It is vital to note that M/2×M/2 mode is limited for the smallest CU size which is 8×8 [3].

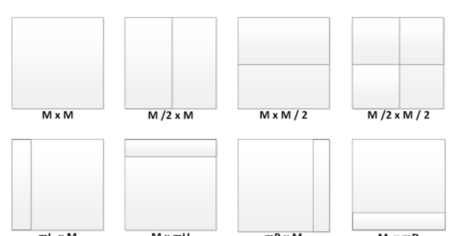

Fig. 2. PU sizes for inter prediction where *M = 8, 16, 32 or 64*.

Inter prediction in HEVC consumes a larger portion of the encoding time as the number of CU sizes and prediction modes have been increased. In addition, newly introduced merge mode, asymmetric partitions immensely contribute towards the compression efficiency while significantly increasing the computational complexity. Generally HEVC compatible encoder follows the rate-distortion optimization process to find the best prediction mode and the optimum CU size. Identifying the huge complexity that has been introduced, HEVC itself has incorporated several approaches to reduce the complexity. Early Skip Detection (ESD) mode utilizes the skip mode detection during the merge mode evaluation process and skips subsequent CU processing. Fast CBF (Coded Block Flag) mode skips processing of subsequent prediction modes if *luma*, and *chroma* CBF of the current CU is equal to zero. In addition, enabling of the fast search from the configuration file results in the encoder using diamond or square search patterns instead of the full search within the search range, in order to find the best match when performing motion estimation.

In addition, numerous attempts have been made in the recent literature to reduce the complexity of inter coding. Some of these attempts focus on improving motion estimation by reducing the number of search points, or by improving the sub-pixel motion estimation, which is also a complex task. Another branch of research focuses on determining the PU and CU size decision at an early stage. In [4], authors make use of Mean Square Error (MSE) and compare it with a threshold to decide whether to split the current CU. This method achieves a 34.83% time saving compared to the HM6.0 reference software. However this method requires the full evaluation of a certain depth level in order to make the comparison with a calculated threshold, which requires more time.

A Motion Vector Merging (MVM) approach is proposed in [5] to determine the best PU size. A 34% time reduction has been achieved with respect to the HM3.4 reference software, but this approach doesn't consider the CU size decision. An optical flow based approach is considered in [6], to identify the motion homogeneity. However performing optical flow calculation within the encoding process is a computationally expensive operation. Approaches in [7] and [8] utilizes neighboring and co-located CU information to decide on the unnecessary depth levels. These methods achieve a 45% and 30% average time saving respectively. However relying on the depth levels of neighboring CUs may result in invalid size decisions and there is probability to propagate these errors into subsequent CUs. In [7], CU level decision is made after finishing the mode decision in current depth. If the decision is made to split the CU further, the previous evaluation will be futile.

This paper introduces a fast and less complex CU size decision taking algorithm for HEVC inter coding. The proposed algorithm utilizes motion homogeneity and RD cost information to classify a CU to one of the predefined categories. The split probability for a CU is calculated using a simple nearest neighbor algorithm which is then used to make the split decision. This early termination prevents unnecessary CU evaluations resulting in an average of 73.25% saving of encoding time while maintaining a marginal impact on the rate-distortion performance.

The rest of the paper is organized as follows. Section 2 provides an illustrative overview of the proposed algorithm. Section 3 describes experimental results and finally Section 4 concludes the paper.

2. PROPOSED ALGORITHM

Considering the partitioning behavior of CUs with respect to inter prediction, it can be observed that blocks with similar motion tend to utilize large CUs while blocks with complex motion utilize smaller CUs [5][6][8][9]. Fig. 3 shows a typical partitioning structure for a particular frame of a video sequence with average motion complexities.

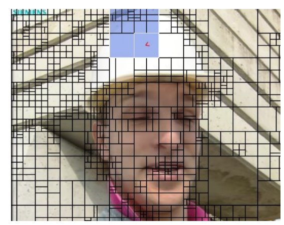

Fig. 3. A typical CU partitioning for 'foreman' sequence.

In order to identify the motion homogeneity of a given CU, inter N×N mode is initially evaluated for the CU. Thereafter, based on motion vector distribution of its constituent blocks, a classification approach is decided. Based on the analysis made on distinct video sequences for inter N×N mode for each CU, we have identified nine categories that are depicted in Fig. 4. Two motion vectors are considered to be similar when their horizontal and vertical components are equal and when they point to the same reference picture. Also Fig. 5 illustrates number of CUs that fall under each category for four distinct video sequences. From these data it can be seen that all nine categories have been utilized whereas category 0, 5 and 6 that corresponds to *'all four equal motion vectors*', '*three similar motion vectors with one that differs*' and '*all four unequal motion vectors*', being the most frequently used.

Fig. 4. Block classifications based on motion homogeneity. Similar motion vectors are identified with same index and color.

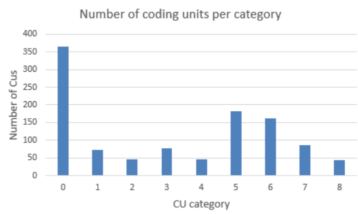

Fig. 5. Number of CUs fall for each category, when inter N×N mode is evaluated for 'foreman', 'highway', 'hall' and 'news' sequences.

In the proposed method, each sequence is subjected to a training phase while encoding the first four inter frames. During this phase, statistics on CU split decisions and rate-distortion costs are collected into the following two arrays along with split and non-split information for each of the categories based on rate-distortion cost as the feature vector. The two 2D arrays are denoted as follows.

$$blockSplit[cat_i][c_k] \quad (1)$$

$$blockNSplit[cat_i][c_k] \quad (2)$$

where $i=0, 1, \dots, 6$ and $k=0, 1, 2, \dots, 29$. cat_i indicates the category index corresponding to the motion vectors of the constituent blocks and c_k indicates rate-distortion cost. Fig. 6 shows the range of rate-distortion costs that four distinct sequences exhibit for inter N×N prediction mode.

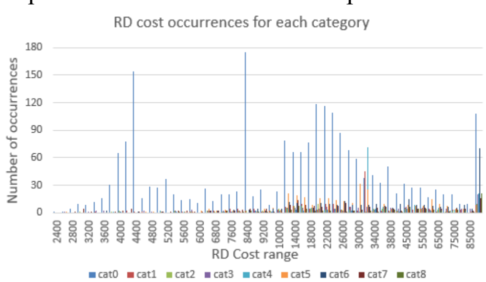

Fig 6. Rate-distortion cost histogram for inter N×N mode.

Based on the empirical analysis which was performed on various sequences, in this paper we consider rate-distortion cost levels ranging from 2500 to 100000 with a gap of 100 between each. All costs beyond 100000 are aggregated into the 100000 cost level. The initial statistical data gathering process is illustrated below.

1. Find motion vectors of constituent blocks by performing inter N×N prediction.
2. Calculate the RD cost for this mode, RD_c
3. Figure out the category based on classifications mentioned in fig. 4, cat_j
4. RD_c (RD cost) is quantized to the nearest c_k, RD_{ck}
5. If decision is to split,
   $blockSplit[cat_j][RD_{ck}]{+}{+}$,
6. If decision is not to split,
   $blockNSplit[cat_j][RD_{ck}]{+}{+}$,

Initial training phase is terminated after processing the first four inter frames. From the next inter frame, following steps are followed to obtain a decision for the current CU.

1. Find the motion vectors of constituent blocks by performing inter N×N prediction.
2. Calculate RD cost for this mode, RD_c
3. Figure out the category based on the classifications mentioned in fig. 4, cat_m
4. RD_c (RD Cost) is quantized to the nearest c_k, RD_{ck}
5. Find out the split probability,

$$prob(split) = \frac{blockSplit[cat_m][RD_{ck}]}{(blockSplit[cat_m][RD_{ck}] + blockNSplit[cat_m][RD_{ck}])} \quad (3)$$

6. If $prob(split) < 0.5$, decision is taken not to split
7. If $prob(split) = 0$, a new split probability is calculated using a simple nearest neighbor method. For this split and non-split counts of surrounding costs within same category are considered.

$$prob(splitNN) = \frac{\sum_{ck=k-2}^{k+2} blockSplit[cat_m][RD_{ck}]}{\left(\sum_{ck=k-2}^{k+2} blockSplit[cat_m][RD_{ck}] + \sum_{ck=k-2}^{k+2} blockNSplit[cat_m][RD_{ck}]\right)} \quad (4)$$

8. If $prob(splitNN) < 0.5$, decision is taken not to split
9. If any of the above conditions are not satisfied, decision is taken to split the current CU.

3. EXPERIMENTAL RESULTS

Simulations were conducted on a range of HD and CIF video sequences of natural and synthetic content. Video sequences have been selected such that they span across simple to high complex motion compositions. QP values were set to 22, 27, 32 and 37 and all the frames were encoded in *low delay P main* configuration in HM 12.0 encoder software [10]. The frame rates of HD and CIF sequences are 60 fps and 25 fps respectively. All simulations were carried out on an Intel core i5 machine with 8GB RAM.

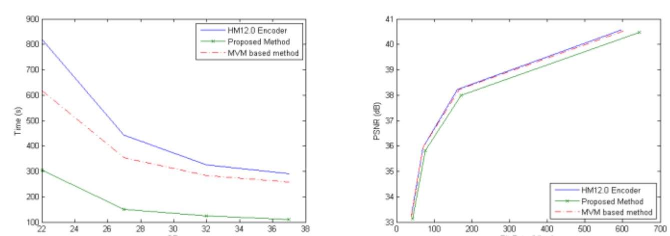

Fig 7. Encoding time and rate-distortion performance for 'Hall' CIF video sequence.

Fig.7 and Fig. 8 demonstrate rate-distortion performances and encoding time performances with respect to HM12.0 reference software [10] and MVM based approach discussed in Sampaio *et al.* [5] for two CIF sequences and Fig. 9 shows the same performance graphs for a HD sequence.

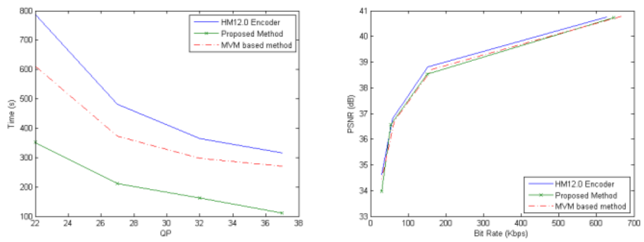

Fig. 8 . Encoding time and rate-distortion performance for 'Highway' CIF video sequence.

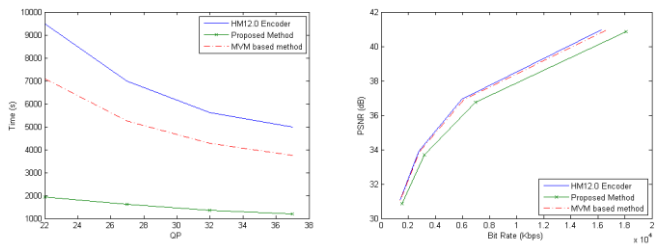

Fig 9 . Encoding time and rate-distortion performance for 'Beergarden' HD video sequence.

These graphs clearly show that the proposed algorithm can achieve a significant performance gain with respect to encoding time with minimal impact on rate-distortion performance. These results elaborate, that by performing a pre-evaluation of the CU and making the CU size decision, could contribute immensely towards encoding time improvement.

In order to validate the decision making accuracy of the proposed algorithm, the probabilities of making the same split decision by the proposed algorithm with respect to that of the HM12.0 reference software [10], is calculated. The ratio of number of times the same decision is made and the total number of split decisions, is analyzed as follows.

$$P(R) = \frac{C_{RD}}{C_{TD}} \tag{5}$$

where *P(R)* is the probability of making the same decision as the HM12.0 reference software [10], C_{RD} is the number of time the same decision is made and C_{TD} is the total number of decisions made. The probabilities given in Table 1 are for 50 frames in respective sequences and it can also be observed that by increasing the number of training frames and by inserting training frames at certain intervals will increase the split decision accuracy significantly.

Table 1. Probabilities of making the same split decision as the HM reference software

QP	**22**	**27**	**32**	**37**
News	0.66	0.7012	0.7517	0.8680
Hall	0.634	0.693	0.734	0.812
Highway	0.6298	0.6554	0.7801	0.8155
Foreman	0.6846	0.667	0.7281	0.7695

Table 2 summarizes the results with respect to average time saving (*ΔT*), BD Rate increase [11], average Δ PSNR, and average Δ Bit Rate for *low delay P main* configuration with respect to HM12.0 reference software. *ΔT, ΔPSNR*, and *ΔBit Rate* have been obtained as follows.

$$\Delta T = \frac{T_{ORG} - T_{PROP}}{T_{ORG}} \times 100 \tag{6}$$

where T_{ORG}, is the encoding time of HM12.0 encoder and T_{PROP}, is the encoding time achieved with the proposed algorithm.

$$\Delta PSNR = PSNR_{PROP} - PSNR_{ORG} \tag{7}$$

$$\Delta BitRate = \frac{BitRate_{PROP} - BitRate_{ORG}}{BitRate_{ORG}} \times 100 \tag{8}$$

These objective results depict that the proposed algorithm achieves a significant time saving with respect to the HM12.0 encoder, with a negligible rate-distortion performance loss. Moreover, visual examinations show that the proposed algorithm has no visual quality impact on the reconstructed video sequences when compared with that of the reference software. Therefore it is evident that the proposed method is capable of achieving a higher encoding time save with respect to the HM 12.0 reference software, when methods in current literature achieve a maximum of 45% time save with respect to the previous HM versions.

Table 2. Performance of the proposed algorithm with respect to HM 12.0 reference software (low delay P main configuration)

Sequence	***ΔT%***	**BD rate%**	**Δ PSNR**	**Δ Bit Rate%**
Beergarden 1080p	77.139	18.35	-0.005	0.13
Café 1080p	76.288	26.65	-0.004	0.19
Musicians 1080p	76.5	21.46	-0.004	0.18
GT_Fly 1088p	75.15	22.82	-0.01	0.12
Average	**76.26**	**22.32**	**-0.023**	**0.155**
News CIF	83.67	7.88	-0.03	0.04
Hall CIF	79.13	10.7	-0.02	0.05
Highway CIF	61.78	20.72	-0.16	0.12
Foreman CIF	56.382	29.68	-0.132	0.12
Average	**70.24**	**17.24**	**-0.08**	**0.082**

4. CONCLUSION

In this paper, we bring forward a fast CU size selection algorithm for HEVC inter prediction. The proposed algorithm utilizes the motion homogeneity and attempts to classify a particular CU to a predefined category and make a decision on the CU splitting based on previous training results. Due to the early decision making made prior to the evaluation of a particular CU, the proposed algorithm can provide an average time saving of 73.25% with a negligible impact on the PSNR and bit rate. As the future work, we will focus on utilizing features other than rate-distortion cost for the classification process and further improve rate-distortion performance while maintaining the time saving that has been achieved.

5. REFERENCES

[1] http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_ White_Paper.html.

[2] G. J. Sullivan, J. Ohm,W.-J.Han and T.Wiegand, "Overview of the high efficiency video coding (HEVC) standard", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[3] I.-K. Kim, J. Min, T. Lee, W.-J. Han and J.-H. Park, "Block Partitioning Structures in HEVC", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1697 – 1706, Dec. 2012.

[4] Q. Yu, X. Zhang , S. Wang and S. Ma, "Early termination of coding unit splitting for HEVC", *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC)*., pp. 1 – 4, Dec. 2012.

[5] F. Sampaio, S. Bampi, M. Grellert, L. Agostini and J. Mattos, "Motion Vectors Merging: Low Complexity Prediction Unit Decision Heuristic for the Inter-prediction of HEVC Encoders", *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 657 – 662, July 2012.

[6] J. Xiong, H. Li, Q. Wu and F. Meng, "A Fast HEVC Inter CU Selection Method Based on Pyramid Motion Divergence", *IEEE Transactions on Multimedia*, vol.16, pp. 559-564, Feb 2014.

[7] J. Leng, L. Sun, T. Ikenaga and S. Sakaida, "Content Based Hierarchical Fast Coding Unit Decision Algorithm for HEVC", *International Conference on Multimedia and Signal Processing (CMSP)*., vol. 11, pp. 56-59, May 2011.

[8] L. Shen, Z. Liu, X. Zhang, W. Zhao and Z. Zhang, "An Effective CU Size Decision Method for HEVC Encoders", *IEEE Transactions on Multimedia*., vol. 15, pp. 465 – 470, Feb. 2013.

[9] Z. Liu, L. Shen and Z. Zhang, "An Efficient Intermode Decision Algorithm Based on Motion Homogeneity for H.264/AVC", *IEEE Transactions on Circuits and Systems for Video Technology*., pp. 128 – 132, Jan. 2009.

[10] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-12.0/.

[11] G. Bjontegarrd, "Calculation of average PSNR differences between RD curves," ITU-T SC16/Q6 13th VCEG meeting, No.VCEG-M33, Austin, TX, Apr. 2001.