# EFFICIENT CODING UNIT SIZE SELECTION BASED ON TEXTURE ANALYSIS FOR HEVC INTRA PREDICTION

*Thanuja Mallikarachchi, Anil Fernando, Hemantha Kodikara Arachchi*

Centre for Vision Speech and Signal Processing
University of Surrey
United Kingdom
d.mallikarachchi@surrey.ac.uk, w.fernando@surrey.ac.uk, h.kodikaraarachchi@surrey.ac.uk

## 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 quadtree hierarchy has a significant impact on the encoding time of high definition (HD) videos. This paper presents a fast coding unit size decision-taking algorithm for intra prediction in HEVC. The proposed algorithm utilizes a low complex texture analysis technique based on the local range property of a pixel in a given neighborhood. Simulation results show that the proposed algorithm achieves an average of 72.24% encoding time efficiency improvement with similar rate distortion performance compared to HEVC reference software HM12.0 for HD videos.

***Index Terms—*** Video Coding, HEVC, Intra 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 amount of video data that will dominate consumer networks. High Efficiency Video Coding (HEVC), which was standardized in early 2013, intends to cater for 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].

While inheriting most of the features and methodologies from its predecessors, HEVC introduces a number of new features which improves 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 have been introduced. As the resolution of digital video increases to HD and beyond, the maximum macroblock size of 16×16 seemed inadequate to grasp features of high-resolution videos [2]. In the main profile of HEVC, a CTU is partitioned into multiple coding units of sizes ranging from 8×8 to 64×64. This flexible quadtree 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.

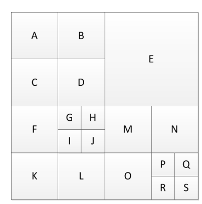



**Fig. 1. An example partition structure of a 64×64 CTU.**

HEVC supports multiple PU sizes that enhance inter and intra prediction processes. Fig.2 illustrates PU sizes that are supported for intra prediction. It should be noted that N×N mode is only available for the smallest CU size, which is 8×8[3].

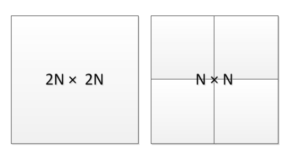



**Fig. 2. PU sizes for intra prediction where *N=4, 8, 16, 32.***

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).

Intra prediction in HEVC has evolved in many ways when compared to that in H.264/AVC. Apart from the introduction of hierarchical quadtree based partitioning structure, HEVC introduces 33 prediction angles for directional prediction as opposed to 8 directional modes in H.264. Therefore, HEVC supports 35 intra prediction modes along with DC (mode 0) and Planar (mode 1) modes. Due to the increased number of intra modes, a more efficient mode coding mechanism has been introduced to minimize bit rate overhead [4].

During the encoding process, the HEVC encoder evaluates combinations of coding unit partitions and prediction modes to determine the best coding unit size and prediction mode for a given CTU. As a result, increased number of prediction modes and partition block sizes in return increases the complexity of rate-distortion (RD) optimization process. The introduction of Rough Mode Decision (RMD) step, which initially evaluates the whole set of 35 modes and reduces the number of candidates for full RD process, has significantly reduced the processing time [4]. However, it can be noticed that processing of HD video is still time consuming due to brute force evaluation of the quadtree partition structure.

This paper introduces a fast and less complex coding unit size decision-taking algorithm for HEVC intra coding. The proposed algorithm utilizes texture complexity derived from local range values of a given block. Furthermore, it proposes an early termination criterion to prevent CUs splitting unnecessarily. Less computational power required for the texture analysis and savings from the early termination process of the proposed algorithm, altogether account for an average computational complexity saving of 72.24% for encoding HD videos, while maintaining a marginal impact on the coding efficiency.

The remainder of this paper is organized as follows. Section 2 provides an overview of previous work followed by Section 3, which details an illustrative overview of the proposed algorithm. Section 4 describes experimental results and finally, Section 5 concludes.

## 2. RELATED WORK

The literature reports a significant amount of researches to optimize the CU partitioning task by utilizing diverse characteristics and features that are encountered along the encoding process. A branch of research utilizes information such as RD costs, prediction residuals and distribution of RD costs of neighboring blocks to determine an early termination point for the CTU evaluation. Statistical distribution of RD costs has been utilized in [5] to determine whether to split a given CU or not. If RD cost of the CU is less than a predefined threshold, it is not split any further. This early-termination algorithm has achieved a 24% time saving compared to HM5.2rc1 software. Depth level of the current tree-block is predicted using spatially adjacent tree-blocks in [6]. This method has achieved a 21% computational complexity saving on average. Variation of mode costs calculated within RMD process is utilized in [7] to make CU splitting decisions. If the cost exceeds a predefined threshold, the CU is split to the next depth level straightaway without processing the current level. This technique has demonstrated about 32% encoding time saving for *all intra main* configuration in HM7.0 software. The evaluation of a CU to obtain initial cost information in order to make a decision on further splitting, consumes more time in these approaches. There is a clear tendency to split less homogeneous areas into smaller blocks while keeping homogeneous regions in large blocks. Proposed methods in [8] and [9] estimate texture complexity using gradient information. In [8], horizontal and vertical Sobel kernels are used to calculate gradient information and partition decision is taken based on standard deviations of gradient values. Once the decision is made, a fine tuning process is carried out to determine whether partitioned coding units can be merged together. This is achieved by comparing the sum of RD costs of constituent blocks with the cost of coding unit. In [9], texture complexity calculated from directional gradients are compared with an adaptive threshold based on quantization parameter (QP) and prediction unit (PU) size and less complex CUs are merged together to generate the final partition structure. A content adaptive threshold determination mechanism has been proposed in [10], which reports a 28.8% time saving for HD videos. Kernel based gradient calculation is a time consuming operation and hence a low complex texture analysis method, which doesn't exceed the computational cost of the RD optimization cycle, is required to decide CU splitting at an early stage.

## 3. PROPOSED ALGORITHM

Considering the partitioning behavior of intra-coded CTUs in HEVC, it can be observed that, the RD cost can be minimized by partitioning homogeneous regions of an image into larger CUs and less homogeneous regions into smaller CUs. Fig.3 below depicts partition structures formed by HM12.0 [11] encoder with *all intra main* configuration for two sequences.

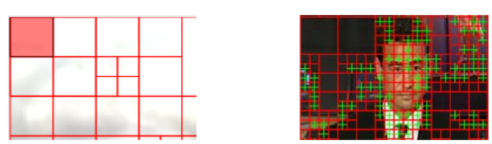


**Fig.3. Homogeneous image regions (L) have larger CU partitions while non-homogeneous regions (R) have smaller CU partitions.**

Homogeneous regions of an image segment can be identified by performing a texture analysis on the region of interest. In the methods proposed in [8] and [9], gradient information of the coding unit image segment has been utilized to derive texture complexity. However, a texture analysis method, which is less computationally expensive and which could be used as a measure of homogeneity is required to make an analysis of the coding unit under consideration. This analysis method should not cause an additional overhead once integrated into the encoder such that time taken for a CU texture analysis is less than time taken for full RD optimization cycle to decide on the best CU size.

Here, we propose a simple texture analysis method based on the local range, which is the variation of a pixel with respect to its local neighborhood [12]. It can be experimentally shown that local range *mean* values and their *variance*, of a pixel in a pre-defined neighborhood provide an indication about the texture homogeneity.

Local range of a pixel is calculated by considering the maximum and minimum pixel values of a given neighborhood.

$$LR_{CU(x,y)} = Max_p - Min_p \quad (1)$$

$$Max_p = Max\left\{ I_{CU}\left( x-\frac{w}{2} : x+\frac{w}{2} ; y-\frac{w}{2} : y+\frac{w}{2} \right) \right\} \quad (2)$$

$$Min_p = Min\left\{ I_{CU}\left( x-\frac{w}{2} : x+\frac{w}{2} ; y-\frac{w}{2} : y+\frac{w}{2} \right) \right\} \quad (3)$$

where $I_{CU}$ is the image segment of current CU, *(x, y)* is the location of the current pixel in the CU image segment and *w* is the size of the neighborhood surrounding current pixel. $LR_{CU}$ is the local range matrix generated for the current coding unit. It was observed that when '*w*' increases, both PSNR and bit rate increases. Therefore, based on an empirical optimization study, *w* was set to 3, which resulted in a better RD performance.

In general, the Local Range (LR) value is smaller in homogenous regions while it becomes larger when the texture is visually complex. Fig. 4 shows *standard deviation* and *mean* values of LR samples of two image blocks with different characteristics. It is evident that both mean and standard deviation of LR values are large in coding units with complex textures.

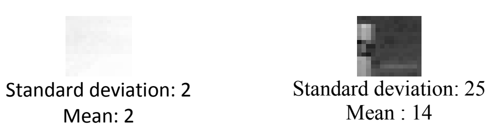



**Fig. 4. Homogeneous region (L) has a lower local range mean value while complex region (R) has a high value for mean and variation.**

With further analysis of the LR values and partition information, it can be observed, that HEVC encoder partitions regions with low LR values to higher block sizes while regions with high LR values are partitioned to blocks with smaller sizes. Fig. 5 and Fig. 6 illustrate mean and standard deviation variations with respect to partition decisions of the encoder. These statistics are obtained by comparing local range values and partition decisions made by the HM12.0 encoder with respect to multiple video sequences with different texture conditions.

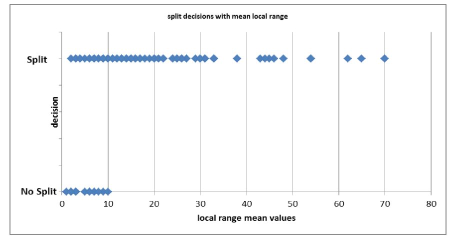



**Fig. 5. Mean values of local range of a 2N×2N CU and split decisions.**

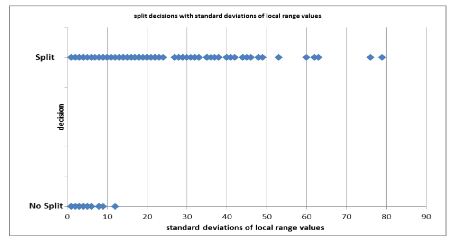



**Fig. 6. Standard deviations of local range of 2N×2N CUs and split decisions.**

These graphs depict that if the mean and standard deviation of local range values of a given CU are large, it tends to be partitioned further. Therefore, a content dependent threshold setting is performed by considering both mean and standard deviation of local range values in a 2N×2N CU as well as standard deviations of local range values in constituent blocks.

Considering the abovementioned observations, we propose the algorithm illustrated in Fig. 7 to make CU size decision prior to encoding each CU.

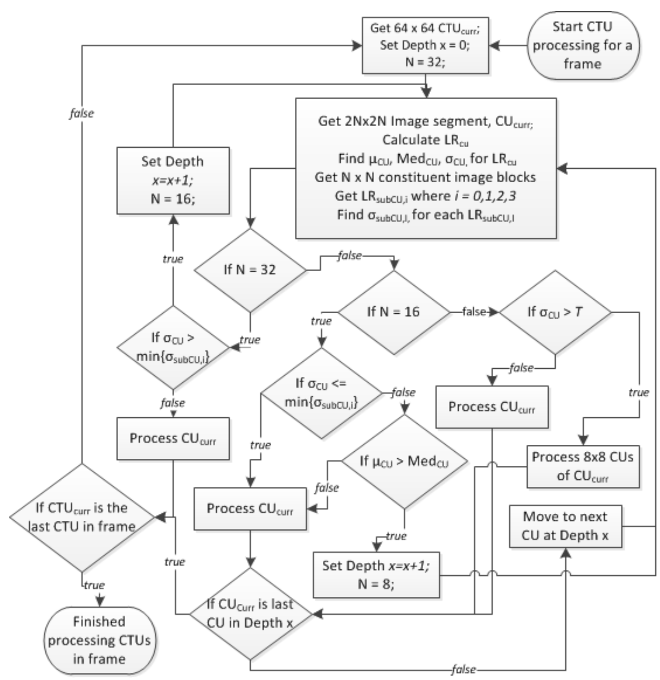



**Fig. 7. CU size determination algorithm using local range and its properties.**

In the algorithm elaborated in Fig. 7, $\mu_{CU}$ and $\sigma_{CU}$ are the mean and standard deviation of LR values of current CU, respectively, $\sigma_{subCU,i}$ is the standard deviation of LR values in constituent blocks such that $i= 0,1,2,3$ and $med_{CU}$ is the median of LR values in current CU block. $CU_{curr}$, is the image segment corresponding to current CU and $CTU_{curr}$ is the current 64×64 CTU. After extensive analysis of partitions with respect to diverse video sequences, in this paper we have set, $T = (QP/16) + 11$. We let the decision making of whether to use N×N or 2N×2N PU size for 8×8 CUs, to the rate-distortion optimization due to the lack of information available in 8×8 CUs for LR analysis.

## 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 plain to high complex texture environments. QP values were set to 22, 27, 32 and 37 and all the frames were encoded in intra mode using the *all intra main* configuration file provided with the HM 12.0 reference software. The frame rates of HD and CIF sequences were set to 60 fps and 25 fps respectively. All simulations were carried out on an Intel core i3 machine with 4GB RAM.

Performance of the proposed algorithm is compared against three different state-of-the-art algorithms. The first algorithm is the algorithm implemented in HM12.0 reference software [11], based on rate-distortion optimization. The second algorithm is the gradient based intra prediction method discussed in Satish *et al.* [7]. In this algorithm, the current coding unit of 2N×2N is filtered with two Sobel kernels (vertical and horizontal) to obtain gradient information. Standard deviation of gradient values ($\sigma_{CU}$) are compared with standard deviation of gradient values ($\sigma_{subCU,i=0,1,2,3}$) of constituent sub-blocks. If $\sigma_{CU} > mean(\sigma_{subCU,i})$, the current CU is split. The "*two pass search*" for 32×32 blocks is disabled (i.e., if 64×64 block is marked for split, no further search is performed at 64×64 level) and simulations are performed with *all intra main* configuration.

The third reference algorithm is the RD cost analysis method described by Kim *et al.* [5]. The RD cost of a coding unit is compared with a predefined threshold to make the decision on further splitting of the CU.

Fig. 9 and Fig. 10, respectively, compare the RD and encoding time performance of the proposed algorithm with the reference algorithms. As it can be observed in the figures, the gradient based algorithm for texture analysis is more computationally expensive than the proposed local range value based texture analysis. Moreover, RD cost analysis based method shows a better encoding time performance than gradient based analysis. However, the RD cost for a given CU needs to be calculated before making a decision. Since this operation is computationally expensive, the proposed algorithm outperforms RD cost analysis based method. The impact on encoding time is reflected in the encoding time performance graphs for simulated video sequence. It is understandable that this time saving could also yield opportunities even for already optimized encoders to achieve higher encoding time performance with respect to intra coding.

Fig. 8 shows partition structures obtained by both HM12.0 encoder and proposed algorithm for Poznan Street HD sequence. It can be seen that the proposed algorithm has almost achieved a similar partition structure to HM software. As a result, the proposed algorithm has achieved a comparable RD efficiency. Being able to achieve a partition structure similar to which is being generated by brute force RD optimization, is the key factor behind the time saving that is visible here. The decision to split a coding unit is taken prior to full evaluation of the CU, thereby making our algorithm more efficient while achieving similar rate-distortion performance.

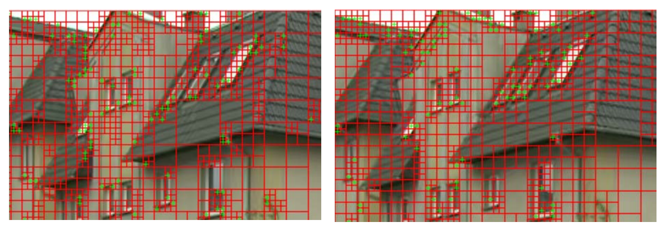


**Fig. 8. Partition structures formed during encoding process. The left figure corresponds to partition structure formed during HM12.0 encoder while the right corresponds to the proposed algorithm.**

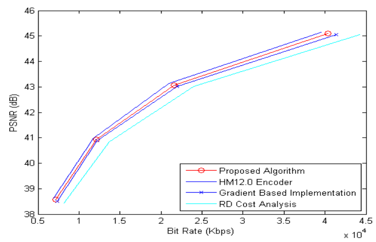
Furthermore, it can be observed that partition structure plays an important role in encoding time statistics. If a homogeneous block is not split any further, a considerable encoding time can be saved, as no further analysis on lower levels is required. As a result, greater encoding time saving can be observed for sequences with less complex textures. Moreover, the proposed algorithm has shown that significant encoding time improvement can be achieved even when CTUs are partitioned into smaller size CUs due to complex textures, since CUs of higher depth levels are not evaluated in this scenario.

Table 1 summarizes the results with respect to the time saving ($\Delta T$), BD Rate [13], Δ PSNR, and Δ Bit Rate. $\Delta T$ has been obtained as follows.

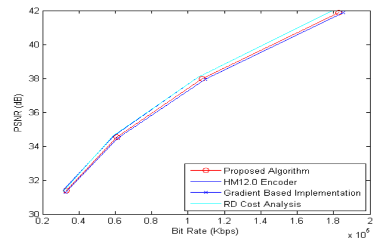
$$\Delta T = \frac{T_{ORG} - T_{PROP}}{T_{ORG}} \times 100 \quad (4)$$

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

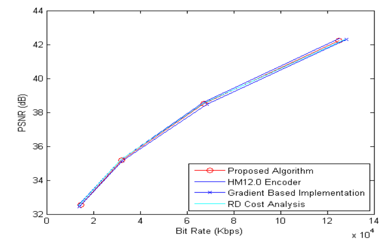
These objective results depict that the proposed algorithm achieves a significant time saving with respect to the HM12.0 encoder as well as other methods, with a negligible rate-distortion performance loss. Moreover, visual examinations show that the proposed algorithm has no visual quality impact on the reconstructed video sequences.



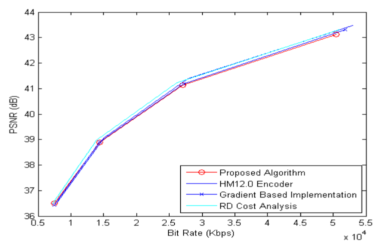

(a) HD Café sequence



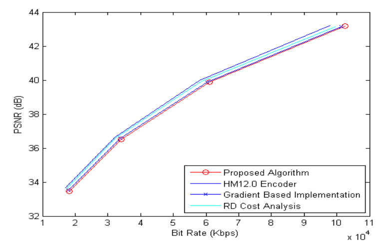

(b) HD Beergarden sequence



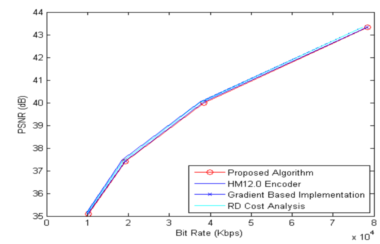

(c) HD Dancer sequence (Synthetic)



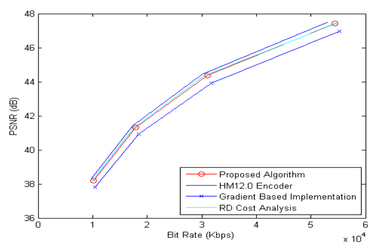

(d) HD GT Fly sequence (Synthetic)



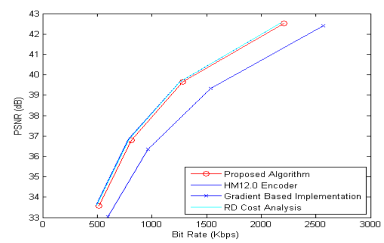

(e) HD Musicians sequence



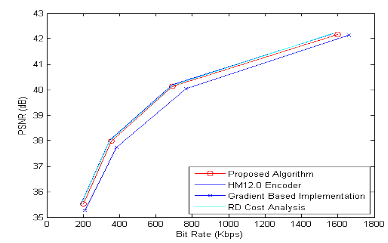

(f) HD Poznan Street sequence




(g) HD 24h_WaterSplash_0 sequence

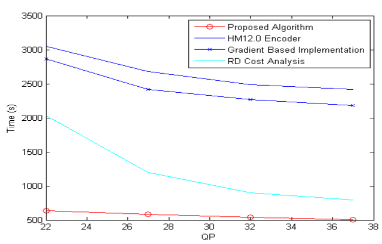



(h) CIF Hall sequence

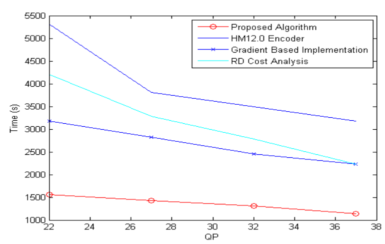



(i) CIF Highway sequence

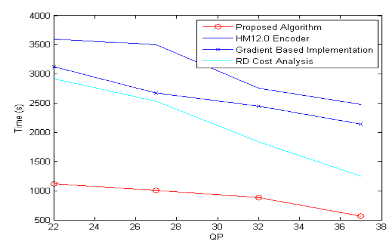
**Fig. 9. Rate-distortion performance curves for video sequences.**



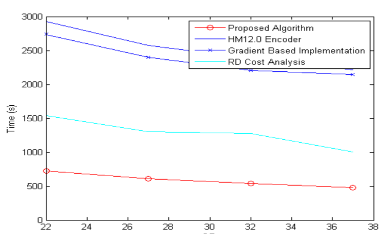

(a) HD Café sequence



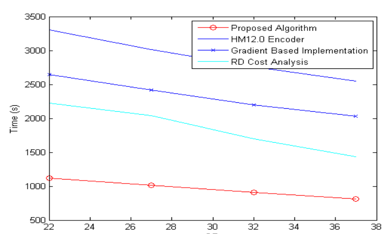

(b) HD Beergarden sequence



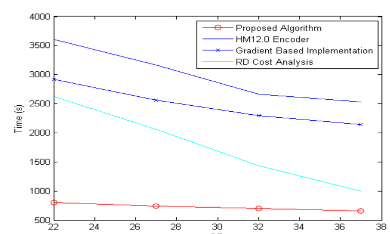

(c) HD Dancer sequence (Synthetic)



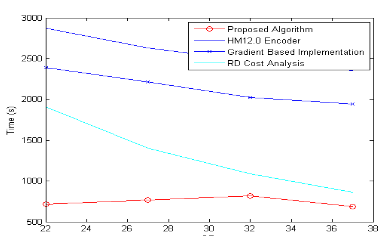

(d) HD GT Fly sequence (Synthetic)



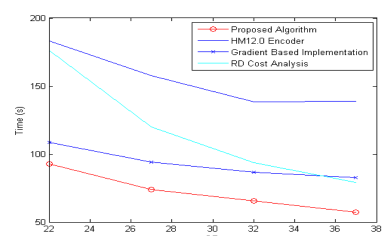

(e) HD Musicians sequence



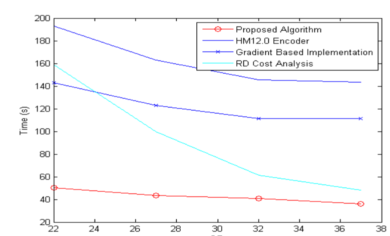

(f) HD Poznan Street sequence




(g) HD 24h_WaterSplash_0 sequence




(h) CIF Hall sequence




(i) CIF Highway sequence

**Fig. 10. Encoding time performance curves for video sequences.**

**Table 1.** Performance of the proposed algorithm with respect to HM12.0 encoder in *all intra main* configuration

| Sequence | Proposed Algorithm | | | | Gradient Based Algorithm | | | | RD Cost Based Algorithm | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *ΔT%* | BD Rate % | Δ PSNR | Δ Bit Rate % | *ΔT%* | BD Rate % | Δ PSNR | Δ Bit Rate % | *ΔT%* | BD Rate % | Δ PSNR | Δ Bit Rate % |
| Beergarden HD | 64.89 | 3.7 | -0.06 | 2.9 | 31.52 | 6.72 | -0.13 | 3.85 | 21.25 | 1.78 | -0.02 | -0.07 |
| Café HD | 78.74 | 4.3 | -0.04 | 3.02 | 9.66 | 8.12 | -0.10 | 5.25 | 54.9 | 20.03 | -0.13 | 15.15 |
| Musicians HD | 66.9 | 9.9 | -0.1 | 4.7 | 20.08 | 7.16 | -0.08 | 4.73 | 36.79 | 2.09 | -0.04 | 1.01 |
| GT_Fly 1088p | 76.82 | 5.4 | -0.2 | -4.6 | 5.95 | 1.5 | -0.16 | -3.9 | 49.39 | -0.83 | -0.13 | -6.8 |
| Dancer 1088p | 71.93 | 2.1 | -0.05 | 2.3 | 16.94 | 4.65 | -0.08 | 2.53 | 33.76 | 3 | -0.03 | 0.81 |
| Watersplash HD | 70.88 | 5.3 | -0.07 | 3.42 | 17.02 | 15.99 | -0.49 | 5.81 | 49.9 | 3.82 | -0.11 | 1.81 |
| PoznanStreet HD | 75.54 | 5.9 | -0.05 | 3.06 | 16.7 | 4.265 | -0.01 | 2.64 | 42.1 | 2.13 | -0.02 | 0.36 |
| **Average** | **72.24** | **5.22** | **-0.08** | **2.11** | **16.83** | **6.91** | **-0.15** | **2.98** | **41.15** | **4.57** | **-0.06** | **1.75** |
| | | | | | | | | | | | | |
| News CIF | 51.0 | 3.3 | -0.08 | 2.26 | 15.8 | 16.57 | -0.29 | 11.59 | 27.96 | 1.07 | -0.02 | 0.36 |
| Hall CIF | 53.38 | 4.2 | -0.04 | 2.54 | 39.68 | 30.51 | -0.39 | 20.69 | 25.73 | 0.85 | -0.01 | 0.77 |
| Bridge-far CIF | 72.286 | 6.9 | -0.03 | 2.72 | 13.93 | 3.52 | -0.02 | 1.8 | 39.9 | 1.18 | 0 | 0.16 |
| Highway CIF | 73.517 | 2.4 | -0.03 | 2.21 | 24.19 | 18.12 | -0.18 | 9.75 | 45.15 | 0.75 | -0.01 | 0.17 |
| Stefan CIF | 42.74 | 4.6 | -0.02 | 4.23 | 46.54 | 4.97 | -0.13 | 3.39 | 14.99 | 0.19 | 0 | 0.06 |
| **Average** | **58.58** | **4.28** | **-0.04** | **2.79** | **28.02** | **14.7** | **-0.20** | **8.76** | **22.77** | **0.61** | **0.00** | **0.304** |

## 4. CONCLUSION

In this paper, we propose a fast CU size selection algorithm for HEVC intra prediction. The proposed algorithm utilizes the local range (LR) values and their variance of a pixel in a neighborhood to derive the texture homogeneity of a coding unit. Experimental results show that CUs with larger LR means and variances tend to be partitioned into smaller sizes while smaller LR values causes not to split the CU. Due to the low complexity of texture analysis and decision is made prior to encoding a particular CU, the proposed algorithm can provide an average time saving of 72.24% for HD sequences with a minimal impact on the PSNR and bit rate.

Future work will focus on utilizing RD cost and partition decisions of neighboring coding units to improve this process while extending a similar methodology for inter prediction.

## 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.-R. 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] J. Lainema, F. Bossen, *Member*, *IEEE*, W.-J. Han, *Member, IEEE*, J. Min, and K. Ugur, "Intra coding of HEVC standard", *IEEE Transactions On Circuits and Systems For Video Technology.*, vol. 22, no. 12, pp. 1792 – 1801, Dec. 2012.

[5] J. Kim, Y. Choe, and Y.-G. Kim, "Fast coding unit size decision algorithm for intra coding in HEVC", *IEEE International Conference on Consumer Electronics (ICCE).*, pp.637 – 638, Jan. 2013.

[6] L. Shen, Z. Zhang, P. An, "Fast CU Size Decision and Mode Decision Algorithm for HEVC Intra Prediction", *IEEE Transactions on Consumer Electronics.*, vol. 59, no.1, pp. 207-213, 2013.

[7] H. Zhang, Z. Ma, "Early Termination Schemes for Fast Intra Mode Decision in High Efficiency Video Coding", *IEEE International Symposium on Circuits and Systems (ISCAS).,* pp.45-48, May. 2013.

[8] S. Lokkoju, D.Reddy, "Fast coding unit partition search", *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT).,* pp. 315-319, Dec. 2012.

[9] Y. Zhang, Z. Li, B. Li, "Gradient based fast decision for intra prediction", *IEEE Visual Communications and Image Processing (VCIP).*, pp. 1-6, Nov. 2012.

[10] G. Tian, S. Goto, "Content Adaptive Prediction Unit Size Decision Algorithm for HEVC Intra Coding", *Picture Coding Symposium (PCS).*, pp-405-408, May 2012.

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

[12] http://www.mathworks.co.uk/help/images/analyzing-the-texture-of-an-image.html

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