STEADINESS ANALYSIS FOR OPTIMAL GOP SIZE SELECTION IN HEVC

Vigneswaran Poobalasingam and Ebroul Izquierdo

School of Electronic Engineering and Computer Science, Queen Mary University of London

ABSTRACT

High Efficiency Video Coding is the latest and most advanced video coding standard. It supports various group of pictures (GOP) sizes and types such as low delay and random access. The size of the GOP substantially influences the temporal coding process. Therefore, a suitable GOP selection strategy can have a significant impact in the compression efficiency. In this paper, a strategy for GOP selection is proposed. It is derived from a new, low complexity measure of the temporal "steadiness" in the video content. Contrasting conventional approaches the proposed technique does not relies on previous estimation of motion vectors or motion information in the video sequence. As such, it can be used for automated encoding parameter optimization at the start of the coding process. The proposed technique leads to improved encoding efficiency at a negligible computational cost, when compared with the standard coding settings. A comprehensive experimental evaluation confirm that on average -6.69%, BD-rate gain and 14.18% time savings can be achieved.

***Index Terms*—** High Efficiency Video Coding, Group of Pictures.

1. INTRODUCTION

The High Efficiency Video Coding (HEVC) standard [1], was ratified in January 2013. HEVC represents the latest generation of video compression standard achieving high compression efficiency to address the growing demand for high and Ultra-High definition video content [2]. HEVC reportedly achieves 50% higher coding compression efficiency over its predecessor H.264/MPEG4 Advanced Video Coding (AVC) [3] at equal perceptual visual quality level [4]. Similar to its predecessors, the HEVC encoder works by categorizing the pictures in a video sequence into three coding types, Intra, denoted as I frames, inter Predicted, denoted as P and Bi-directionally inter predicted, denoted as B. Each frame is processed using a 2-D decorrelation transform followed by quantization and entropy coding. This process is also known as a hybrid approach and it has been in use since the inception of video coding standardizations, e.g., MPEG1 and H.261 [1]. In parallel to the development of the standard, a reference software has been implemented too. It is referred to as the HEVC Test Model (HM).

The HEVC standard encodes slices in a sequence by dividing them into group of pictures (GOPs). The size of GOPs can be set as an encoding parameters before starting encoding process. Usually, the number of frames in a GOP represents a hard-limit for the delay at the decoder side, as typically all frames in the GOP need to be decoded before they can be displayed. HEVC was designed to support variable GOP size. This means that a bitstream formed of GOPs of different sizes is perfectly standard-compliant and decodable by a conventional HEVC decoder. Since, different GOP sizes undoubtedly leads to different coding performances in terms of bit rate gains and complexity, an optimal setting of the GOP size, leads to significant performance gains. Unfortunately, it is impossible to predict the best GOP size for a given video sequence without prior analysis or knowledge of the video content. Therefore, developing techniques for pre-processing and automatic optimization of this critical encoding parameter is critical for improved coding efficiency and performance gains. Further, observe that many typical HEVC encoder implementations assume a fixed GOP size for the encoding of the entire sequence.

To the best of our knowledge, there is no reported study in the literature on GOP selection for HEVC. However, there are a good number of reported studies for GOP size selection on previous video coding standards. Ascenso et al. [5] proposed a method to select the GOP size based on Hierarchical Clustering, which uses block statistics. Charles et al. [6] proposed a method to select the GOP size based on Return Channel Suppression in Wyner-Ziv Video Coding, which uses rate-distortion cost to select GOP size. Zatt et al. [7] proposed a method to select the GOP size based on the video content for efficient H.264/AVC encoding, which uses two dimensional entropy and pixel dissimilarity to detect scene change then select the GOP size that will fit within the scene. Jun-Ren et al. [8] proposed a method to select the GOP size using motion vectors and residuals to detect the scene changes and select the GOP size accordingly. These works assume that information about the motion activity or motion vectors is available.

In this paper, an algorithm is proposed for improved GOP selection without prior knowledge or estimation of motion vectors or motion activity in the scene. The proposed technique is based on inter-frame analysis of texture changes by encoding texture information into a single descriptor and measuring the perceived lack or amount of changes in this information from frame to frame. Thus, the proposed technique does not rely on previous estimation of motion vectors or motion information and can be used for automated encoding parameter optimization at the start of the coding process. In this paper, we call this perceived inter-frame lack of texture or information changes as "*steadiness*". The proposed technique leads to improved encoding efficiency, when compared with the standard coding settings. The analysis conducted in this work also include subjective classification of the testing corpus according to the perceived motion activity and its use to assess the performance of the proposed temporal steadiness as a measure to derive the optimum GOP size for improved coding.

2. BACKGROUND

During the development of the HEVC standard common test conditions (CTC) were devised, under which encoders should be tested to enable a fair comparison between different approaches. Among these conditions, several GOP structures are defined typically referred to as profiles. CTC define the following profiles:

All Intra (AI): Each frame is encoded as I frame. Because no inter picture prediction is used, it is suitable for higher bit rate applica-

tions. The quantization parameter (QP) offset is set to 0 and kept constant over the whole sequence

Random Access (RA): A hierarchical B structure is used. The coding efficiency achieved by the bidirectional hierarchical prediction structure is higher than the other configurations. It has however a larger delay due to the picture reordering.

Low Delay P picture (LDP): The first frame is encoded as *I* frame and the subsequent frames are encoded as P frames. Since reordering of frames is not allowed and only past frames are used for prediction, the coding delay in this configuration is very small.

Low Delay B picture (LDB): The first frame is encoded as I frame and subsequent frame are encoded as B frame. Moreover, since past I and B frame are used for prediction, a low coding delay, similar to LDP, but with higher coding efficiency (because of bi-prediction) is achieved.

Each profile includes a GOP table, namely a set of rules defining the GOP size, structure and the QP values to encode each frame in the GOP, and the reference picture set to use for each of the frames in the GOP. The QP is computed by means of a QP offset value as well as the slice type (I, P or B) and its temporal ID.

3. IMROVED AUTOMATED GOP SELECTION

For this study, a corpus of 42 video sequences of 1080x720 resolutions and different degrees of inter-frame activity was used. This is a fairly large corpus, when compared with standard data sets for testing video coding technology. The aim is to ensure that a good variety of videos is assessed to derive statistically significant conclusions. The corpus was first subjectively (or manually) classified, and then encoded with five different GOP types. The standard Bjøntegaard model is used to calculate the coding efficiency by calculating the Bjøntegaard-Delta bit-rate (BD-BR) and using it to confirm the subjective classification of the videos according to the perceived steadiness and to benchmark the proposed approach.

3.1. Subjective classification

The subjective classification of the video sequences according to their inter-frame activity was performed as follows. The corpus was first divided into two groups: "*single*" and "*multi-scene*" video sequences. Single scene sequences have a single continuous camera view shot, whereas multi scene has more than one camera view and scene changes. Next, each one of these two groups were divided into two sub-groups according to subjectively perceived amount of local motion. Sequences that have visually noticeable change of information were classified as "*active*". For example, when both background and foreground of the video change or small object move fast. On the other hand, a sequence where only a portion of the frame has visually noticeable changes was classified as "*static*", e.g., when at least on average 50 percentage of the frame is static and the other part of the scene has small local or global activity.

Each of these four sub groups is further divided into *high (H), medium (M) and low (L)* classes by visually estimating the average amount of changes throughout the sequence. For example, a single-active video with a fast moving objects from the start to end, it is grouped in the *H-single-active* category. If it only has fast moving for around a quarter of its length and then it displays slow moving object, then it is classified as *L-single-active*. The remaining videos are classified as *M-single-active*. The final subjective classification leads to a total of 12 different groups. Table I shows the classification of all the 42 videos according to this subjective assessment.

3.2. Steadiness analysis of video sequences

The MPEG-7 Homogeneous Texture Descriptor (HTD) was selected as basis for the proposed steadiness measure. HTD is known to have good discriminative properties encapsulating well texture information form small areas in a video frame. As described in [9] and [10], first the mean (f_{av}) and standard deviation (f_{sd}) of all luma samples are computed. Subsequently, the frame is transformed using Fourier transform and expressed in the polar coordinates. After that, Gabor filter is applied to strengthen the image directional information. The outcome of Gabor filter is denoted as $H_i(\omega, \theta) = G_{sr}(\omega, \theta) \times F(\omega, \theta)$, where i is the number of sub bands generated by dividing the frequency domain, $i \in \{1, 2... 30\}$.

Finally, 30 energy coefficients [*e1, e2... e30*] and 30 energy deviations [*d1, d2... d30*] are computed. The HTD of a given block in a frame is computed as a 62-component feature vector:

$$HTD = [f_{av}, f_{sd}, e1, e2... e30, d1, d2... d30].$$

The similarity between two feature vectors HTD_i *and* HTD_j is usually computed using the distance (1) below. Here, α is a weight value to compensate scale differences between *HTD* components.

$$D_T(HTD_i, HTD_j) = |f_{av\_i} - f_{av\_j}| + |f_{sd\_i} - f_{sd\_j}| + \sum \left|\frac{e_i - e_j}{\alpha}\right| + \sqrt{\sum (d_i^2 - d_j^2)} \quad (1)$$

Observe that (1) is a combination of continuous L_1 and L_2 norms and it has been confirmed to achieve good discrimination power when comparing two blocks in two images or for judging image similarity in visual information retrieval. In our application, we need to divide each frame in small blocks of fix size and extract the HTD descriptor for each block. To measure the steadiness between two consecutive frames we split each frame into blocks of small size and then we estimate the difference between two blocks in two consecutive frames at the same position. Since we need a single value for the steadiness of the whole block, we add up all the resulting differences over the whole frame. The results of this process using (1) lead us to conclude that it fails to provide a good measure for the target steadiness in video sequences.

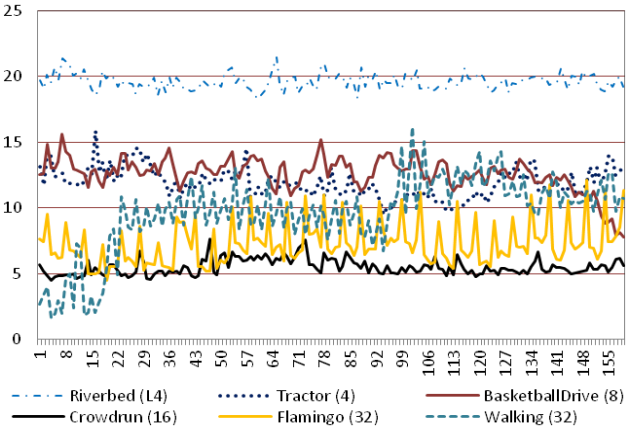

Fig. 1: Plots for "steadiness" of few selected sequences using (1).

The main reason for this is the fact that we are interested on measuring steadiness over the whole frame at once, regardless the intensity of steadiness in isolated or large image regions. Basically, (1) provides a high granularity measure, while we need a more Coarse metric. As a simple example, let's assume we have an image of dimension 1280x1280, which is split into 100 blocks of dimension 128x128. If just 10% of the image, i.e., 10 blocks, have extreme large HTD differences *L* according to (1), while the remaining 90 block are completely steady, then the overall steadiness

measure will be *10L*. On the other hand, a different video may have random motion activity everywhere, giving HTD differences smaller than (but close to) *L/10* for each block. This would be still a reasonable large amount of activity but in this case it appears everywhere in the frame. Here, the overall steadiness will be measured as smaller than *10L*. Clearly, this result is completely wrong since we want the first case to show higher steadiness than the second case. This analysis has been cemented with a thorough experimental evaluation, which rules out the use of (1) to measure steadiness in the sense addressed in this work. Fig. 1 displays "steadiness plots" using (1) for six different sequences. Here, the optimum GOP size is given in brackets after the sequence name. Clearly, these GOP sizes mix randomly and obviously, it is impossible to discriminate them using this measure.

Following the previous analysis, a different measure was derived. To allow for variable granularity the *HTD* vector values are first normalized:

$$C_Q=(C - C_{min})/(C_{max} - C_{min})Q_L$$

where C_Q is the normalized vector value, Q_L is a fixed scaling or quantization factor, C is the original vector value, C_{min} and C_{max} are minimum and maximum vector values. Next, C_Q is further quantized by rounding it to the next integer value. Thus, the final normalized and quantized HTD vector values are given as:

$$\hat{C}_Q = \text{round } (C_Q) \quad (2)$$

Using (2), we obtain $\hat{H} = (\hat{C}_{Q1}, \hat{C}_{Q2} \ldots \hat{C}_{Q62})$. To measure the steadiness between two consecutive frames we split each frame into blocks of size 128x128. For each block, the $\hat{H}$ vector is calculated. Then, the difference between two blocks, B_i and B_j, in two consecutive frames and at the same position is estimated using:

$$D(B_i, B_j) = V_L - d(\hat{H}_i, \hat{H}_j), \quad (3)$$

where V_L is the length of the vector, i.e., 62, and d represents the *Hamming Distance* between the two vectors $\hat{H}_i$ and $\hat{H}_j$. Observe that (3) is a sort of "inverted" Hamming distance in which the zeros (rather than the non-zeros) are counted.

Finally, for two consecutive frames, the values of (3) in each block are summed up. The resulting value Z, as given by (4), is then regarded as a measure for the steadiness of the underlying subsequent frames. Here, n is total number of blocks in the frame. In the sequel we call the values Z given in (4) as "zero-counts".

$$Z = \sum_{k=0}^{n-1} D(B_{i,k}, B_{j,k}). \quad (4)$$

3.3. Automated GOP Selection Using Steadiness

Using the results of (4) and comprehensive empirical observations of the behavior of this measure over the whole test corpus, an automatic strategy for optimal GOP selection was derived. This strategy is based on a basic analysis of the behavior of the mean μ and variance σ^2 of obtained zero-counts Z for all sequences in the test corpus. The first observation is that optimal *GOPL4, GOP4 and* GOP8 tends to cluster within the first quartile of all estimated means in the observation data, while GOP16 and GOP32 tend to cluster above the first quartile. Therefore, assuming that λ is the point of separation between the first 25% of data then: *if $\mu < \lambda$ then the optimal GOP size is with high probability GOPL4, GOP4 or GOP8. Otherwise, the optimal GOP size is with high probability GOP16 or GOP32.* This empirical observation sets the first corner stone for the strategy for automated GOP selection. Furthermore, if we assume that the zero-counts obtained by (4) are normalized to the unit and the observed data obeys a uniform density function, then *$\lambda=0.25$*. Note that this critical assumption, i.e., that the mean of the zero-counts obey a uniform distribution, is reasonable, since for a given random video it is impossible to predict its corresponding steadiness, without any prior analysis or knowledge on the actual video.

Another important aspect is that the means μ of the optimum zero-counts for GOPL4, GOP4, GOP8 and GOP16 distribute over intervals of same length but with GOPL4 and GOP4, as well as, GOP8 and GOP16 presenting significant overlaps. On the other hand, it appears that the corresponding variances significantly differ in each case and therefore σ^2 can be used to better discriminate these four groups of GOPs. These observations lead to the strategy outlined below:

Pseudo-algorithm for automatic GOP selection
if ($\mu < \lambda/3$) then GOPL4
else if ($\mu < 2\lambda/3 \wedge \sigma^2 > \varepsilon$) then GOP4
else if ($\mu < \lambda \wedge 3\sigma^2 \in [\lambda, 2\lambda]$) then GOP8
else if ($\mu < 4\lambda/3$) then GOP16
else if ($\mu >= 4\lambda/3$) then GOP32
else GOP8

Here, the value of the parameter ε serves to separate GOPL4 and GOP4. Observe that this separation is delicate since there is no clear relation between steadiness, or amount of local motion, and these two GOP sizes. However, empirical observations again led us to conclude that a very small number in the variance of the data σ2 can be used to separate GOPL4 and GOP4. This number is set to ε=0.01 in the proposed algorithm assuming again that the zero-counts are normalized to the unit.

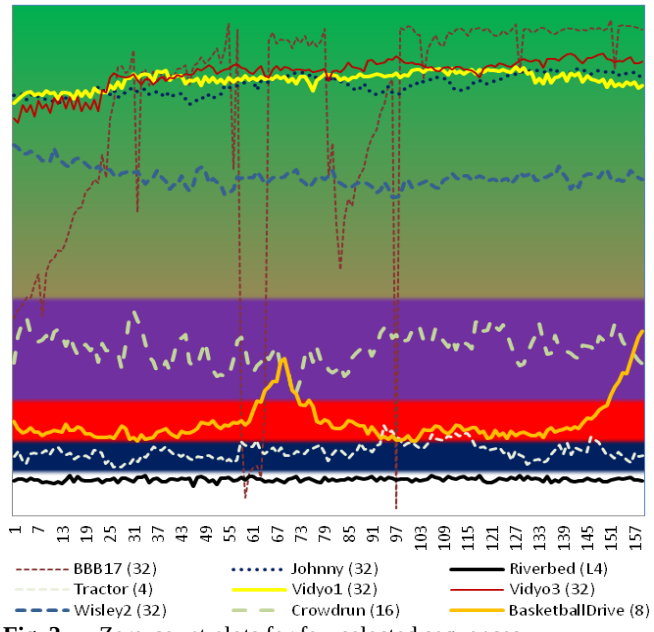

Fig. 2. Zero-count plots for few selected sequences.

Finally, the GOP separation is completed by exploiting the condition $3\sigma^2 \in [\lambda, 2\lambda]$, since in the majority of cases large GOPs exhibit

large variances too. This last observation brings the final element needed to derive the proposed algorithm.

4. RESULTS

The proposed algorithm was implemented and tested using the HM code version 12.0 and the corpus of 42 HD test sets from JCT-VC, SVT, MoCA [11], Peach [12], and hdrsamples.com [13]. All tests run using 2.4 GHz Intel Westmere (E5645) machines with 24 GB of RAM. The experiments were performed according to the JCT-VC CTC [14]. Results are presented in terms of luma BD-rates, where negative BD-rate values represent efficiency gain with respect to the anchors and positive time saving values represent speed-up gain with respect to the anchor. All the video sequences were encoded with four different QP values (22, 27, 32, and 37) and for five different GOP types, namely low delay 4, random access 4, 8, 16, and 32. The BD-rate was calculated using PSNR value of the luma components of each four GOP types except random access 8 (RA8). RA8 was used as an anchor against all others to calculate the BD-rates differences against this anchor.

In addition, the total encoding time was computed for each of the 4 tested QPs according to the formula $\Delta T_i = ((T_A - T_M)/T_A)100$, where T_A denotes the total encoding time for the anchor setting (RA8) and T_M denotes the total encoding time for the alternative testes setting. Finally, the mean value of ΔT_i for all 4 QP test points was computed to obtain the average encoding speed-up ΔT. It is important to note that the IP values for the performed test are set as follows: IP value 32 for video sequences with frame rates 24, 25 and 30. IP value 64 for video sequences with frame rates 50 or 60.

The proposed HTD based algorithm produces the average BD-rate gains of -6.69%, -15.22%, -14.36% for Y, U and V components respectively, and time savings of 14.18% (see Table I). It is important to note that the above time saving value includes the HTD processing time of 160 frames of the corpus. This was calculated by running RA-Main 8 with HTD processing enabled to processes fist 160 frames only. Then, encoding time for RA-Main 8 with HTD was subtracted from encoding time for RA-Main 8 only to obtain the difference, which corresponds, to the HTD processing overhead. Finally, these times were added to the encoding times for different GOP types.

This proposed algorithm selects the optimal configuration in most of the cases, delivering substantial BD-rate gains while reducing the computational cost.

5. CONCLUSIONS

The proposed method proves that selecting the GOP type for video sequence will improve the compression efficiency of HEVC in terms of quality as well as speed. Even though, the proposed algorithm produced a significant gain in BD-rate and time saving, it is using fixed GOP type for the entire sequence by first analyzing it using HTD. Future work includes varying the GOP type within the video sequence to obtain even higher coding gains.

6. ACKNOWLEDGMENT

The authors gratefully acknowledge their colleagues Saverio G. Blasi, R. Weerakkody, A. Gabriellini, M. Naccari. & I. Zupancic for their contributions.

TABLE I. PROPOSED METHOD RESULTS WITH SUBJECTIVE CLASSIFICATION AND THEIR HTD *Z* MEAN AND VARIANCE OF ALL THE TEST SEQUENCES.

Subjective classification			**42 test sequences info**	**Proposed method (algorithm) results**				
Scene type	Motion	Amount of motion	Sequence name	Mean	Variance	GOP type	**BD [%]**	**Time saving [%]**
Single	Active	H	Riverbed	7.05	0.15	L4	**-4.73**	**-23.46**
			Tractor	12.58	2.42	4	**-1.43**	**-0.42**
		M	BQTerrace	37.28	161.61	32	**-6.3**	**30.1**
			ParkJoy	17.99	24.48	16	**-0.42**	**7.42**
			ParkRun	69.84	2.80	32	**-0.12**	**39.08**
			Station	30.02	31.33	16	**-3.43**	**-1.56**
			BasketballDrive	18.24	16.22	8	**0**	**-3.15**
			Stockholm	74.09	0.58	32	**-11.66**	**28.9**
			Sunflower	18.67	46.23	16	**-1.23**	**1.72**
			BBB24	39.24	31.70	32	**-13.76**	**21.14**
			DucksTakeOff	24.03	18.43	16	**5.12**	**8.5**
		L	ChristmasTree	29.14	3.88	16	**-2.69**	**5.83**
			CrowdRun	32.80	9.66	16	**-0.14**	**7.36**
			Mobcal	80.01	29.49	32	**-8.33**	**27.11**
			ParkScene	35.84	3.03	32	**-9.19**	**19.99**
			Walking	22.94	151.08	16	**-2.54**	**6.05**
			Wisley2	66.44	3.57	32	**-11.79**	**32.38**
			RushHour	21.07	2.72	16	**-0.09**	**-1.05**
			Flamingo	37.84	12.99	32	**-4.58**	**12.98**
	Static	H	Cactus	42.36	3.33	32	**-4.18**	**27.04**
		M	FourPeople	83.87	2.27	32	**-12.22**	**19.91**
			Meerkat	43.60	76.84	32	**-4.98**	**20.9**
			Vidyo4	81.86	4.46	32	**-12.51**	**19.87**
			Train	56.84	191.39	32	**-5.92**	**21.16**
			Highway	59.91	58.48	32	**-10.51**	**16.44**
		L	KristenAndSara	82.52	3.39	32	**-10.41**	**18.44**
			Vidyo1	85.48	2.38	32	**-11.13**	**18.62**
			Johnny	84.47	3.59	32	**-10.77**	**18.14**
			Vidyo3	86.74	9.20	32	**-10.35**	**17.4**
Multi-scene	Active	H	LoP1	40.38	15.80	32	**-3.64**	**27.06**
			LoP2	32.78	26.38	16	**-2.68**	**10.15**
			LoP3	24.07	19.40	16	**-3.72**	**13.32**
			Exodus1	23.78	34.59	16	**0.01**	**2.69**
			Tennis	10.27	60.49	4	**-2.63**	**-0.07**
		M	Exodus3	47.29	118.83	32	**0.93**	**20.39**
		L	BBB2	50.12	25.55	32	**-19.83**	**19.65**
			Kimono1	17.93	70.97	16	**-0.71**	**1.66**
	Static	H	BBB10	52.32	459.15	32	**-15.18**	**18.1**
			Exodus2	52.63	113.54	32	**-16.37**	**13.67**
		M	BBB5	88.09	55.11	32	**-19.74**	**15.78**
			BBB17	79.52	519.73	32	**-21.08**	**14.82**
		L	BBB19	60.73	752.07	32	**-6.11**	**21.67**
Average							-6.69	14.18

a. Zero in the BD and Timesaving means no gain against GOP8.
b. BBB is Big Buck Bunny, an animated cartoon and number 2 is second batch of 552 frames.
c. Lop is Life of Pi draft is a movie trailer version and number 1 is first batch of 395 frames.

7. REFERENCES

[1] Sullivan, G.J.; Ohm, J.; Woo-Jin Han; Wiegand, T., "Overview of the High Efficiency Video Coding (HEVC) Standard," Circuits

and Systems for Video Technology, IEEE Transactions on, vol. 22, no. 12, pp. 1649,1668, Dec. 2012.

[2] Cisco Visual Networking Index Predicts Annual Internet Traffic to Grow More Than 20 Percent (reaching 1.6 Zettabytes) by 2018 [Online]: http://newsroom.cisco.com/release/1426270 (Last access: 19/06/2014).

3] Wiegand, T.; Sullivan, G.J.; Bjontegaard, G.; Luthra, A, "Overview of the H.264/AVC video coding standard," Circuits and Systems for Video Technology, IEEE Transactions on , vol.13, no.7, pp.560,576, July 2003.

[4] Ohm, J.; Sullivan, G.J.; Schwarz, H.; Thiow Keng Tan; Wiegand, T., "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)," Circuits and Systems for Video Technology, IEEE Transactions on, vol.22, no.12, pp.1669,1684, Dec. 2012.

[5] Ascenso, J.; Brites, C.; Pereira, F., "Content Adaptive Wyner-ZIV Video Coding Driven by Motion Activity," Image Processing, 2006 IEEE International Conference on, vol., no., pp.605,608, 8-11 Oct. 2006.

[6] Charles Yaacoub; Joumana Farah,; Béatrice Pesquet-Popescu, "New Adaptive Algorithms for GOP Size Control with Return Channel Suppression in Wyner-Ziv Video Coding" International Journal of Digital Multimedia Broadcasting Volume 2009 (2009), Article ID 319021, 11 pages http://www.hindawi.com/journals/ijdmb/2009/319021/

[7] Zatt, B.; Porto, M.; Scharcanski, J.; Bampi, S., "Gop structure adaptive to the video content for efficient H.264/AVC encoding," Image Processing (ICIP), 2010 17th IEEE International Conference on, vol., no., pp.3053,3056, 26-29 Sept. 2010.

[8] Jun-Ren Ding.; Ji-Kun Lin.; Jar-Ferr Yang.; "Motion-based Adaptive GOP Algorithms for Efficient H.264/AVC Compression" Institute of Computer and Communication Engineering Department of Electrical Engineering National Cheng Kung University, Tainan, Taiwan 701.

[9] H. Shao, J. Ji, Y. Kang and H. Zhao, "Application Research of Homogeneous Texture Descriptor in Content-Based Image Retrieval," Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on, pp. 1-4, 19-20 Dec 2009.

[10] M. K. H. K. K. B. M. a. J. K. Yong Man Ro, "MPEG-7 Homogeneous Texture Descriptor," ETRI Journal, vol. 23, pp. 41-51, June. 2001.

[11] MoCA homepage
http://ls.wim.uni-mannheim.de/de/pi4/research/projects/retargeting/test-sequences/

[12] http://hdrsamples.com/

[13] https://peach.blender.org/download/

[14] F. Bossen, "Common test conditions and software reference configurations" JCTVC-L1100, April 2013.