



# HEVC optimization based on human perception for real-time environments

D. G. Fernández<sup>1</sup> · Guillermo Botella<sup>1</sup> · Alberto A. Del Barrio<sup>1</sup>  · Carlos García<sup>1</sup> · Manuel Prieto-Matías<sup>1</sup> · Christos Grecos<sup>2</sup>

Received: 4 July 2018 / Revised: 16 October 2018 / Accepted: 7 December 2018 /

Published online: 18 December 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

## Abstract

High-Efficiency Video Coding (HEVC) is the new emerging video coding standard of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). The HEVC standard provides a significant improvement in compression efficiency in comparison with existing standards such as H264/AVC by means of greater complexity. In this paper we will examine several HEVC optimizations based on image analysis to reduce its huge CPU, resource and memory expensive encoding process. The proposed algorithms optimize the HEVC quad-tree partitioning procedure, intra/inter prediction and mode decision by means of H264-based methods and spatial and temporal homogeneity analysis which is directly applied to the original video. The validation process of these approaches was conducted by taking into account the human visual system (HVS). The adopted solution makes it possible to perform HEVC real time encoding for HD sequences on a low cost processor with negligible quality loss. Moreover, the frames pre-processing leverages the logic units and embedded hardware available on an Intel GPU, so the execution time of these stages are negligible for the encoding processor.

**Keywords** HEVC · CU size decision · Spatial homogeneity · Temporal homogeneity · HVS metrics · GPU · Mode decision · Intra prediction · Inter prediction · Texture analysis

## 1 Introduction

Nowadays, HEVC, aka H.265 [50, 51], is a cutting-edge technology that is attracting much attention in the world of digital video compression. The next-generation HEVC/H.265 compression standard requires half the bitrate that the previous H.264 standard. However, it requires a big computational effort, high power consumption and, consequently, a high economic cost to meet the real time constraints of the high definition (HD) and ultra-high definition (UHD) television resolutions. The HEVC video compression standard has been developed by the Joint

---

✉ Alberto A. Del Barrio  
abarriog@ucm.es

Collaborative Team on Video Coding (JCT-VC), a cooperation among the ITU-T VCEG and ISO/IEC MPEG organizations [50].

An HEVC block diagram is depicted in Fig. 1. The coding structure is very similar to that of the H264 video compression standard architecture, but the new HEVC quad-tree coding block partitioning scheme provides the capability of selecting the most appropriate size for the prediction, coding and transform units. The input frame is subsequently divided into Coding Tree Units (CTUs), which is a similar concept to the H264 macroblock, but instead of a fixed size of  $16 \times 16$  pixels, HEVC allows the use of a CTU size equal to  $64 \times 64$ ,  $32 \times 32$  or  $16 \times 16$  pixels.

In the quad-tree partitioning procedure a CTU can be subdivided into square blocks known as Coding Units (CUs). The CUs represent an image block which is configured to use the same prediction mode (intra, inter or skip) and it is the first level of division in the quad-tree structure. The quad-tree partitioning recursive splitting process starts from the CTU level and it finishes when no further splitting is necessary or when the minimum CU size is reached. The quad-tree subdivision allows the use of multiples CUs to adapt it to the content of the input video. The size of a CU ranges from the CTU size to a minimum of  $8 \times 8$  pixels [3].

The prediction units (PUs) define portions of the image sharing the same prediction information. Intra PUs can only exploit spatial redundancy using the pixel’s neighbourhood by means of DC prediction, planar prediction or 33 angular predictions [31, 50]. Inter PUs are predicted by searching the motion of the current PU in relation to the prior encoded frames. A motion estimation process is required to define these motion parameters as well as an intra prediction decision method to exploit both temporal and spatial redundancies. Figure 2 depicts the quad-tree partitioning structure.

A detailed description of the HEVC standard may be found in our earlier papers [9, 11].

After a thorough review of the related literature, it is possible to conclude that regions within the image which possess a homogeneous texture should not be split in order to maintain the lowest Rate Distortion (RD) cost [15, 29, 32, 37, 49, 52, 53, 68, 70]. Henceforth, we refer to these image areas as spatially homogeneous regions.

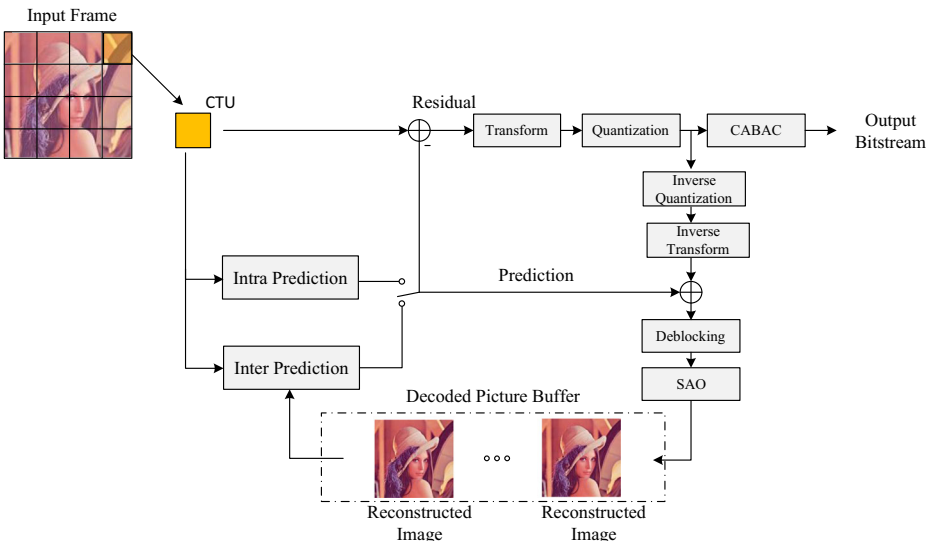
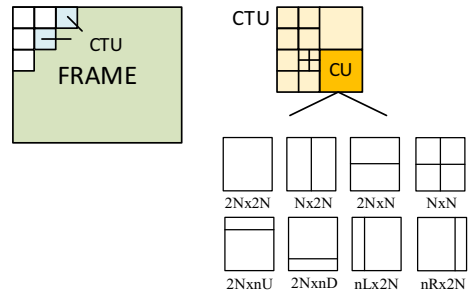


Fig. 1 HEVC block diagram

**Fig. 2** HEVC quad-tree coding block partitioning structure



The same concept can be applied to static blocks and/or portions of the image in which the motion characteristics are the same in order to reduce the workload of the HEVC encoding process [14, 26, 44–46, 56, 59]. Hereinafter, this kind of blocks shall be named temporally homogeneous regions.

In this paper, in order to reduce HEVC encoder complexity, several optimizations are proposed that are based on a previous image analysis aimed at identifying spatially and temporally homogeneous regions. The vast majority of the operations are performed by the embedded HW capabilities available on Intel GPUs for H264 encoding using the input frames in a pre-encoding stage, unlike other works [19, 26, 29, 36, 44–46, 56, 59] in which the image analysis is executed at the CTU level. Since the operations are executed on dedicated hardware, the logic units of the GPU are not used, so they are available for other tasks.

The proposed algorithms have been designed to reduce the execution time with an imperceptible quality loss. The proposed optimizations have been performed in a real-time encoder implementation using metrics based on human perception for the quality assessment, unlike in the state-of-the-art methods where peak signal-to-noise ratio (PSNR) is the only considered metric and a real-time environment is not employed.

The remainder of the paper is structured as follows. The related literature is outlined in Section 2, and the proposal is discussed in Section 3. The metrics employed to characterize the algorithm are outlined in Section 4. The results of the experiments and a detailed comparison with state-of-the-art methods are presented in Section 5. Our conclusions and ideas for future work are detailed in Section 6.

## 2 Related work

Because of the tremendous complexity [24, 31] inherent in the HEVC encoding process, the latest works focusing on video compression center their interest on reducing the main computationally demanding processes such as the quad-tree partitioning procedure and the intra/inter mode decision algorithms. For this purpose several works study the RD cost and prediction mode correlations between different depth levels and the spatial CU's neighborhood [28, 41, 48, 49]. This solution provides excellent results in single core executions but it could introduce unnecessary waits in a multithreading system because of the dependency between different CUs. Other approaches [28, 41, 69] are focused on reducing the encoder complexity by analysing the RD and Hadamard costs. However, the RD cost calculation is a computationally demanding process, as every time this is performed, it requires transforms, quantizations, and a pseudo-CABAC [31].

A fast splitting and pruning method is proposed in [5] using statistical parameters. The solutions based on statistical procedures obtain exceptional results for the sequences under study but they can cause big bitstream size increment when encoding sequences in which the characteristics of the video content vary continuously, such as scene cuts.

Several works can be found in the literature that use gradient algorithms to detect spatially homogeneous areas within an image [21, 25, 40, 53, 66, 68, 70]. For the same purpose, other works use the variance [21, 32, 52], standard deviation [37], mean absolute deviation [49] or the mean of absolute difference among pixels and the standard deviation of the variance operating at the same time [29]. All of these metrics are very computationally expensive operations.

Further to our comments above about the temporally homogeneous regions, different works are focused on detecting this kind of areas to stop the quad-tree partitioning process when the condition is met in a CU. This is possible because the motion parameters in temporally homogeneous regions are the same, so there is no need to divide them into sub-blocks [14, 46, 56, 59] using an adaptive partition size algorithm to characterize the fine movement [46]. In order to conduct precise temporal homogeneity detection, a motion estimation search algorithm is necessary [26, 44–46, 56, 59]. Due to the high computational complexity of the motion estimation [2, 38], a graphics processing unit (GPU) working as co-processor can be an ideal solution, thus offloading the CPU and saving power.

The works presented in [44, 45] employ the mean deviation of the motion vectors obtained from temporally and spatially nearby macroblocks, but they ignore the motion within every CU. Most of works conduct temporal homogeneity detection in the CTU loop [26, 44–46, 56, 59]. In our proposal the temporally homogeneous regions are categorized using the input frames, and in this way dependencies among different processes are avoided and thus the motion estimation can be easily mapped onto a co-processor in a pre-encoding stage.

In our previous works, several optimized algorithms [7–11] were proposed to accelerate the encoding process by implementing a fast method to decide the CU size leveraging the spatially [7, 9–11] and temporally [8, 9] homogeneous regions. The detection of spatially homogeneous regions is based on the relationship between DC and AC coefficients [7, 9–11], and can easily coexist with other methods as the gradient-based algorithms [36]. On the other hand, the temporal homogeneity analysis [8, 9] is based on a hardware H264 motion estimation practiced on the input frames. The main idea consists of halting the quad-tree partitioning recursive process when the current block has been categorized as spatially or temporally homogeneous by the analysis performed on the input frames. If this happens, we interpret that there is no need to continue dividing because the sub-blocks will produce similar decisions. Finally, it must be noted that intra CUs are only analysed for spatial homogeneity, while inter CUs are studied for both spatial and temporal homogeneities.

In contrast with other approaches [28, 41, 48], as the spatial homogeneity categorization is directly executed on the input images, this approach is recommended for parallel implementations as a result of its independency with neighboring CU blocks. For instance, the work presented in [7] implemented such method on the  $\times 265$  open source software [60], which software is an H265/HEVC video encoder application library that allows HD real-time encoding. The implementation of this software [60] is fast and computationally efficient as it supports Wavefront Parallel Processing (WPP) [31] and frame parallelism by means of a multithreaded implementation. The work presented in [10] proposed a CU size decision

method based on spatial homogeneity analysis that was customized for lossless compression mode [31]. This type of compression is commonly employed in certain medical applications with the idea of avoiding the appearance of artifacts, which make it difficult to provide an accurate diagnosis.

The temporal homogeneity-based optimization is presented in [8]. In this paper, the classification of temporally homogeneous regions is implemented on an Intel GPU that processes the input images. We leveraged the dedicated hardware of that GPU in order to perform the motion estimation that allows identifying temporally homogeneous regions. Consequently, the rest of resources, i.e. logic units, which compose the GPU are free for being employed for other tasks. Nonetheless, the encoding is performed using the HEVC test model HM v16.2 [31]. This software is typically employed for evaluating new techniques within the HEVC standard [31], but it is a non-optimized software in the sense that it does not have support for exploiting instruction level parallelism (ILP) in very long instruction word (VLIW) architectures. HM does not support either multithreading or any kind of frame parallelism, so encoding with this software requires a huge amount of time, thus making it impossible to achieve real-time execution.

The temporal and spatial analyses were integrated in a unique system in [9], accomplishing excellent outcomes, as can be noticed in the comparison developed in [9] with state-of-art methods. However, in [9] the method was implemented on the HEVC test model HM v16.2 too. In [11] a new set of novel optimizations based on spatial homogeneity detection were applied to intra and inter prediction mode decision algorithms within the HM v16.2 software. Hence, all of the methods mentioned above, including most of our previous works [8–11], were implemented on the HEVC test model. Because of the promising results achieved in [9, 11] with this software, our proposal is based on implementing the whole flow within a HEVC encoder capable of supporting real time such as  $\times 265$  [60]. We decided to use a microprocessor-GPU environment to test our approach in real time conditions, but the suggested ideas could easily be deployed in other encoders supporting real time regardless of the target hardware (FPGA, DSP, etc).

In this new approach, we provide an extension to the work originally presented in [9, 11] by implementing the previously proposed CU size [9, 11], inter partition [11] and intra prediction [11] mode decision algorithms in a real-time HEVC encoder. Furthermore, the motion vector map retrieved from the motion calculation employed for the temporal homogeneity classification is used to conduct a new simplification which makes it possible to dynamically change the range of the motion search window. Moreover, spatial homogeneity classification is improved by means of a hardware H264 intra prediction mode decision which is also executed on embedded hardware on Intel GPUs using the input frames. Due to the relation between the intra prediction modes of both H264 and H265, the outcome of this process is employed for diminishing the complexity of the HEVC decision. In order to simplify the HEVC intra/inter mode decision, the costs derived from the H264 intra prediction mode decision and the H264 motion search are used in a novel early stop condition. Our scheme performs the operations on GPU logic units and embedded hardware and then the processor performing the encoding is offloaded.

On the other hand, the state-of-the-art methods perform the quality assessment by just using PSNR. However, the encoding should leverage human perception when trying to optimize the encoding process in order to achieve a better visual experience from the user's perspective [13]. In this paper, the quality assessment of the proposed optimizations has been performed using metrics based on the human visual system (HVS).

All in all, to the best of our knowledge there is no flow considering spatially and temporal homogeneous regions and realizing such analyses on the input frames to offload the processor by means of the GPU. Furthermore, a real-time implementation and HVS-based metrics are utilized to verify the performance of our flow. Along this lines, the main contributions of this paper are outlined below:

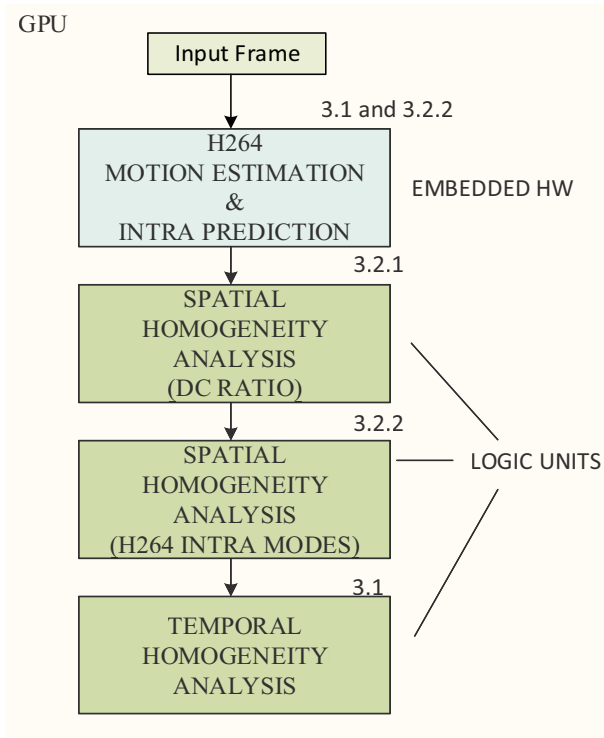
1. The detection of texturally homogeneous regions based on the input frames analysis, which is run on the logic units of the GPU.
2. The detection of temporally homogeneous regions based on the input frames analysis, employing the logic units and motion estimation hardware capabilities of the Intel GPU.
3. The application of fast methods for deciding the inter and intra CU size in texturally homogeneous regions.
4. The application of fast methods for deciding the inter CU size and partition mode in regions presenting temporal homogeneity.
5. The application of a fast method for deciding the intra mode in texturally homogeneous regions.
6. The detection of spatially homogeneous regions in which there is a predominant spatial direction, by means of the logic units and the H264 intra prediction hardware capabilities of the Intel GPU.
7. The application of a fast method for deciding the intra mode in spatially homogeneous regions by using information derived from the H264 intra prediction.
8. Adaptive search range based on temporal homogeneity detection.
9. A novel intra-inter decision based on the costs retrieved from the previous analyses.
10. Quality assessment and threshold determination based on HVS considerations.
11. Real-time encoding implementation.

Although the flow is constructed on ideas published in our prior works, the points 6–11 are completely novel. The points 1–5 leverage the algorithms described in [9, 11], but they have been adapted to be run onto a GPU and thus enable a real-time encoding.

### 3 Proposed HEVC flow

In this section, an HEVC-compliant flow for the encoder is presented. The main objective is to reduce the computational complexity implicit in the HEVC encoder. This flow is based on a prior image analysis that determines that some regions that can be easily encoded. Lighter HEVC decision methods will be employed on these areas. Hence, overall compression execution time will be reduced. The proposed algorithms are implemented at the expense of negligible bitstream increases and without noticeable quality losses. Figure 3 shows the processes of the entire proposal which are executed on the GPU side. These processes are in charge of the image analysis that provides the necessary data to conduct the proposed simplifications within the CTU encoding loop. For every input frame, an H264 motion estimation and an H264 intra prediction mode decision are applied to analyse the frame using the GPU's dedicated embedded hardware.

As can be seen in Fig. 3, once the execution of the H264 motion estimation and intra prediction mode decision have concluded, the temporal and spatial homogeneity analyses of the images are performed. It should be noted that both processes are applied to the input image, so



**Fig. 3** Processes executed on GPU side for the image analysis

they may be implemented independently or just integrated inside the CU encoding loop. Thus, the algorithms that are responsible for spatially and temporally analysing the input images will be implemented on a pre-processing device to offload the processor, which implements the encoding loop. An independent loop working on entire frames reduces the number of waiting stalls, thus enabling efficient communication between the co-processor and the host. In the proposal, there is a one-frame delay between the pre-processing and the encoder, as the input frames are evaluated previously to the encoder API call. Hence, the aforementioned problems when communicating among the computer and the co-processor are eluded.

The flow diagram of the proposal that is run on the processor is depicted in Fig. 4. It should be noted that the findings of this work are designated with numeric values. These numbers belong to the subsections in which the proposed techniques are specified in detail. The novel methods proposed in this new approach with respect to our previous works are surrounded by an orange dashed line.

As Fig. 4 presents, if a CU is categorized as spatially homogeneous after consulting the classification performed by the GPU, fast intra and inter prediction decision methods are performed to diminish the complexity of the HEVC encoder. Besides, the subdivision happening during the CU size decision will be halted and consequently the following CTU will be examined.

The CU can be classified as spatially homogeneous for two reasons. The first one is based on detecting areas with homogenous texture and the second one is focused on the detection of image areas in which there is a predominant spatial direction.

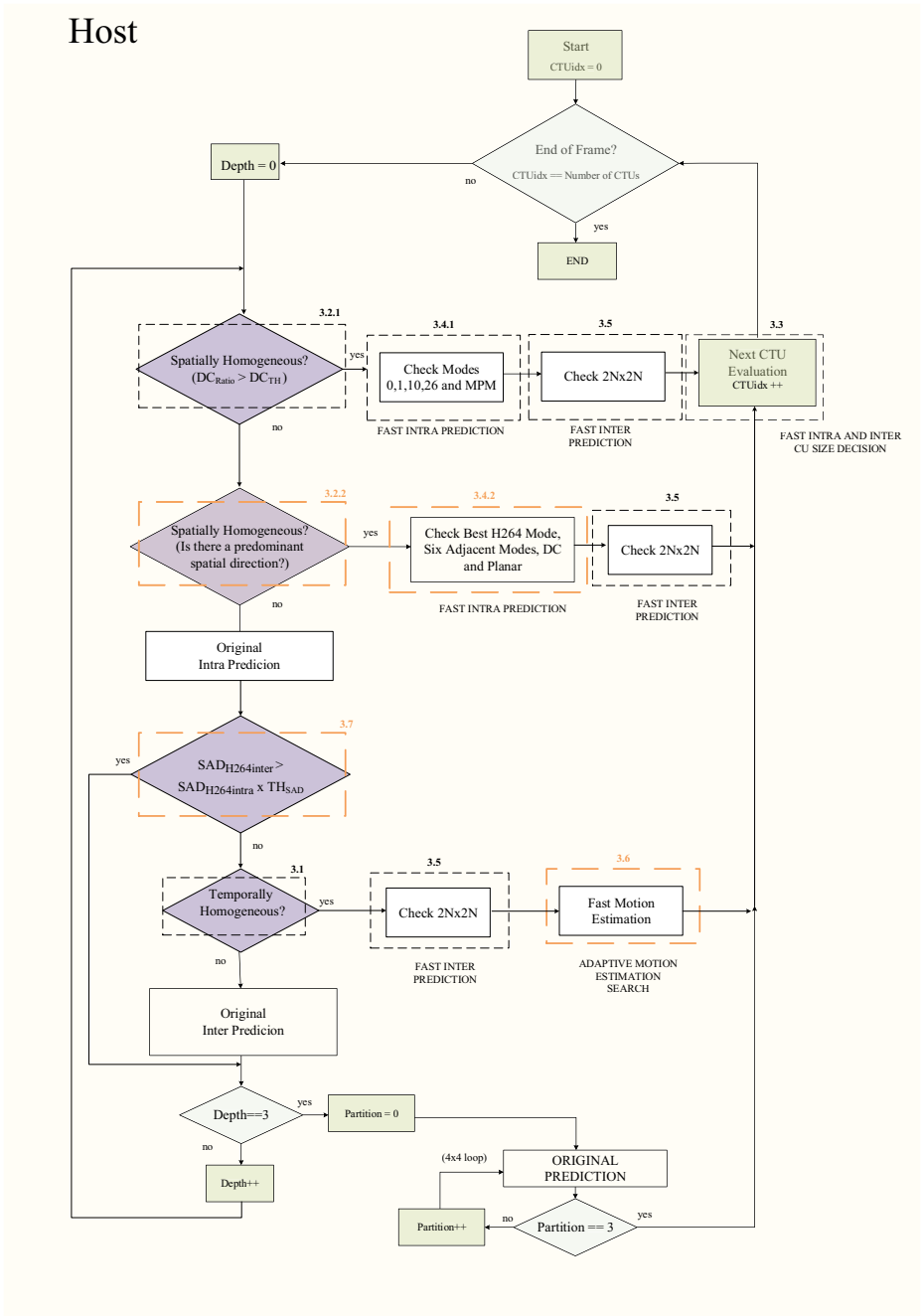


Fig. 4 Execution flow of the entire proposal

If  $SAD_{H264inter} > SAD_{H264intra} \times TH_{SAD}$ , inter prediction is not evaluated in our proposal by skipping the motion search and other computationally expensive processes. The temporal homogeneity is analysed when the previous condition is not met, that is, the CU is not



classified as spatially homogeneous and an inter slice is being processed. In case that the CU belongs to a static region or the blocks within the CU possess close motion characteristics, the algorithm will categorize the CU as temporally homogeneous. Thus, the CU will not be split into minor blocks. Moreover, if a CU is categorized as temporally homogeneous, there are two important consequences: firstly, the number of inter partition modes to be evaluated is diminished, and secondly, the motion estimation search window size is dynamically changed to the type of motion which the video sequence possesses.

On the contrary, if the CU is not considered as temporally or spatially homogeneous, then the original prediction methods are applied.

Meanwhile the depth value is 3, the ongoing CU size is  $8 \times 8$ , thus the  $4 \times 4$  sub-division is permitted. The  $4 \times 4$  loop depicted in Fig. 4 is exclusively applied to  $4 \times 4$  prediction units. If the  $4 \times 4$  loop finishes, a new CTU can be analysed and also the CTU counter is increased. When all the CTUs within a frame have been processed, the whole flow shown in Fig. 4 ends and the next frame can be encoded.

### 3.1 Motion homogeneity analysis

In our previous works [8, 9], an  $8 \times 8$  motion estimation algorithm is applied to the input frames employing *Intel's advanced motion estimation extension for OpenCL* [16, 23] for temporally homogeneous region detection. The H264 motion estimation accelerator on Intel GPUs [16, 17] will be responsible for executing this process. In this way, the functional units within the GPU are free to be employed for other tasks. In order to conduct an accurate motion estimation the hardware engine is configured to perform a motion estimation at  $8 \times 8$  level (four motion vector pairs per macroblock) by using quarter-pixel motion vector precision. The Hadamard Transform is applied to obtain the cost of the residuals. All the motion estimation operations are performed in an exhaustive manner within a  $[\pm 16, \pm 12]$  radius.

The outcome of this task releases an  $8 \times 8$  motion vector map. Nevertheless, there are CU sizes that are larger than  $8 \times 8$ , namely:  $16 \times 16$ ,  $32 \times 32$  and  $64 \times 64$ . For these larger sizes, the  $8 \times 8$  motion vector map is obtained for all the  $8 \times 8$  blocks within the CU, and then the mean absolute deviation of the  $8 \times 8$  motion vector maps is computed. In order to calculate this metric, first the motion vectors medians inside the  $8 \times 8$  motion vector maps are found and then the absolute deviations of these medians are calculated. Finally, the median of the absolute deviations is computed. The mean absolute deviation for temporal homogeneity evaluation in the x axis is described by Eq. 1. This metric is robust in terms of statistical dispersion measuring.

$$H_x = \frac{1}{M} \sum_{k,l}^M \text{abs} \left( mv_x(k,l) - \frac{1}{M} \sum_{k,l}^M mv_x(k,l) \right) \quad (1)$$

In this equation,  $M$  represents the amount of blocks with size  $8 \times 8$  that compose the current CU. The value indicated by  $mv_x(k,l)$  represents the horizontal component of the motion vector associated to the  $8 \times 8$  block identified by the coordinates  $(k, l)$  within a CU. The same process will be performed regarding the y-component with the intention of calculating the temporal homogeneity metric for the vertical motion direction ( $H_y$ ). Then, when  $H_x$  and  $H_y$  are below a threshold  $TH$ , then we infer that there is temporal homogeneity within the CU. Thus, it is not required to divide the CU into smaller blocks to obtain the motion estimation. It must be noted that this threshold  $TH$  has been empirically found, as shown in the experiments section.

Figure 5 illustrates an instance of motion vector map, which has been obtained after using the aforementioned motion evaluation to the second frame of the *tractor* sequence. As can be seen in this analysis, several areas have very similar motion vectors so they are temporally homogeneous candidates. An example of a temporally homogeneous block is highlighted in blue in Fig. 5. Analogously, a non-temporally homogeneous block is surrounded by a yellow box.

### 3.2 Spatial analysis

#### 3.2.1 Texture homogeneity classification based on DC ratio

The first approach for the detection of a spatially homogenous region, aka. *Smooth*, is based on the relationship between DC and AC coefficients, which has been used in our previous works [7–11] to detect areas with homogeneous texture. These sets of coefficients are obtained after applying a discrete cosine transform (DCT) within the video encoder. The main drawback of this calculation in the frequency domain is its high computational cost. Nonetheless, frequency and time domains can be used to represent the same signal according to the Parseval's formula. Thus, by employing the pixel domain it is possible to efficiently obtain the DC ratio ( $DC_R$ ) [9, 11]. In this way, if a block possesses a DC ratio over a threshold, then the block is considered to be homogeneous. The threshold determination is explained in detail in [11].

The outcome of this stage of evaluation is a binary map of spatially homogeneous areas for every analysed block size, namely:  $64 \times 64$ ,  $32 \times 32$ ,  $16 \times 16$  and  $8 \times 8$ , where every position within these maps describes whether a block with an associated size is spatially homogeneous or not. Figure 6 shows the spatially homogeneous zones (dark green) for the *snow mountain* sequence. Finally, it is worth mentioning that in this work the spatial homogeneity categorization has been carried out by employing the logic units of a GPU.

#### 3.2.2 Novel spatial homogeneity classification based on H264 intra prediction

As has been mentioned in the prior sections, gradient-based methods are typically employed for identifying smooth regions [21, 25, 40, 53, 66, 68, 70], since this kind of algorithms permits categorizing regions within a picture possessing a predominant direction. Thus, in

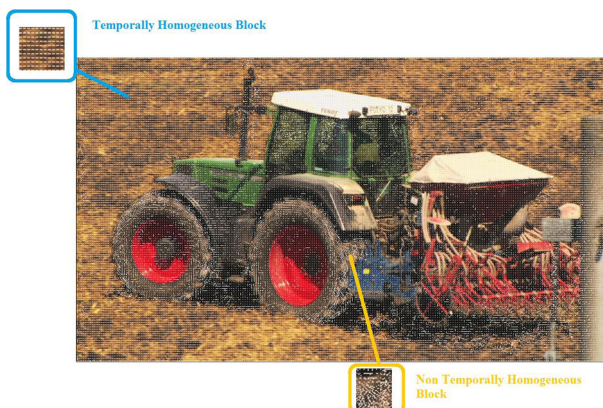
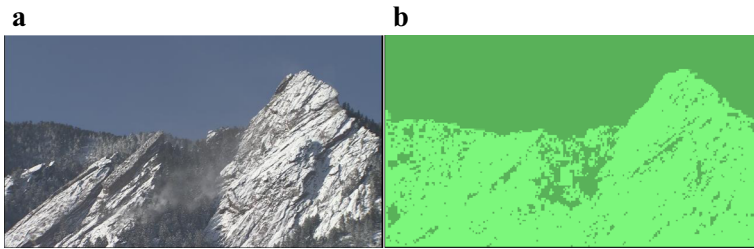


Fig. 5 Motion map retrieved from the hardware engine for the second frame of the *tractor* sequence



**Fig. 6** a Input frame, b Spatial homogeneity categorization

order to decrease the rate distortion (RD) cost, the blocks that show a spatially predominant direction should not be further divided [15, 30, 36].

In the same way, most of the works related to the optimization of the intra prediction mode decision process are based on gradient algorithms for both H264 [6] and HEVC encoders [20, 25, 36, 40, 67]. These works are based on the gradient angle and amplitude, which provide the spatial correlation of the image block to estimate the intra prediction modes, as typically the pixels along the direction of the local edge have the same value [36].

Therefore, we can deduce that it is possible to determine the predominant direction in an area of the image by means of the selected intra prediction modes. For this purpose, in the proposed approach, an H264 intra prediction algorithm is applied to every input frame using *Intel's advanced motion estimation extension for OpenCL* [16, 23]. As in the case of the motion estimation mentioned for temporal homogeneity analysis, all the H264 intra prediction calculations are performed by the embedded hardware available on Intel GPUs [16, 17]. The H264 intra prediction within the hardware engine is configured by default to perform the necessary computations to obtain the best-search prediction modes between adjacent macroblocks and associated residual values.

The output of this OpenCL extension is a set of the best H264 intra prediction modes for every block size. Three block sizes are allowed in H264 for intra macroblocks, namely:  $16 \times 16$ ,  $8 \times 8$  and  $4 \times 4$ . Thus, working at the macroblock level, one intra prediction mode is provided for the  $16 \times 16$  block size, four for  $8 \times 8$  and sixteen in the case of the  $4 \times 4$  block size. Furthermore, the cost associated with every block size is obtained from the hardware H264 intra prediction for every block size, so it is possible to infer what the best block size is, i.e. the block size with the lowest cost. It is important to note that H264 employs a unique block size for every macroblock, so all partitions within a macroblock are  $8 \times 8$  or  $4 \times 4$  if any of these sizes are selected, unlike HEVC, where combinations of different block sizes are allowed within a CTU.

In our proposal the cost of every block size is compared to obtain the best block size in the macroblock using the logic units of the GPU after the finalization of the hardware H264 intra prediction process. For the best block size, if all the partitions within the macroblock have the same intra prediction mode then the macroblock is classified as spatially homogeneous since there is a predominant direction within the macroblock.

In the HEVC encoding loop, a CU is categorized as spatially homogeneous if all the macroblocks (blocks of  $16 \times 16$  pixels) within the corresponding CU are categorized as spatially homogeneous. In the case of  $CU_{size} = 64 \times 64$ , it is necessary to query eight  $16 \times 16$  blocks, four when  $CU_{size} = 32 \times 32$ , and only one for  $CU_{size} = 16 \times 16$ . If  $CU_{size} = 8 \times 8$ , a CU is categorized as spatially homogeneous if the best H264 block size is  $8 \times 8$  and all the  $4 \times 4$  blocks within the  $8 \times 8$  block have the same intra prediction mode.

Figure 7 shows an example of two CUs categorized as spatially homogeneous for  $CU_{size} = 32 \times 32$ . The CU at the top left corner was classified as spatially homogeneous because the best H264 block size was  $16 \times 16$  for the all the macroblocks within the  $32 \times 32$  CU and also for all of them the best intra prediction mode was the vertical mode. In the case of the CU at the bottom right corner, the three  $16 \times 16$  macroblocks showed the DC intra prediction mode as the best mode, whereas the best block size for the fourth macroblock was  $8 \times 8$ . As can be seen in Fig. 7, for this macroblock all the  $4 \times 4$  blocks within the  $8 \times 8$  blocks use DC as the best intra prediction mode. Since the DC intra prediction mode was the choice for the three  $16 \times 16$  macroblocks and the  $8 \times 8$  macroblock, the corresponding  $32 \times 32$  CU was thus classified as a spatially homogeneous  $32 \times 32$  block.

### 3.3 Fast CU size decision

If a block within a CU is categorized as spatially or temporally homogeneous at a given depth, the subdivision process must be stopped because it does not make sense to continue, as the subsequent blocks after partitioning would possess very similar features. In this way, it is possible to accelerate the encoding process.

Finally, it should be noted that in inter CUs only the temporal homogeneity is studied, whereas the spatial homogeneity is evaluated for both inter and intra CUs.

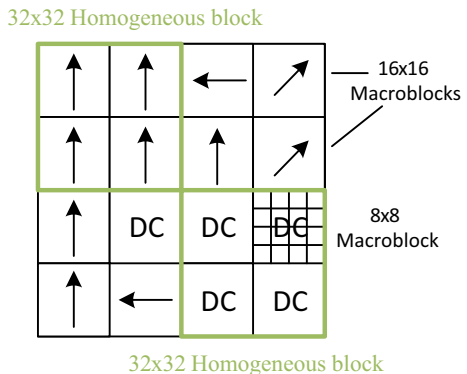
### 3.4 Intra mode decision

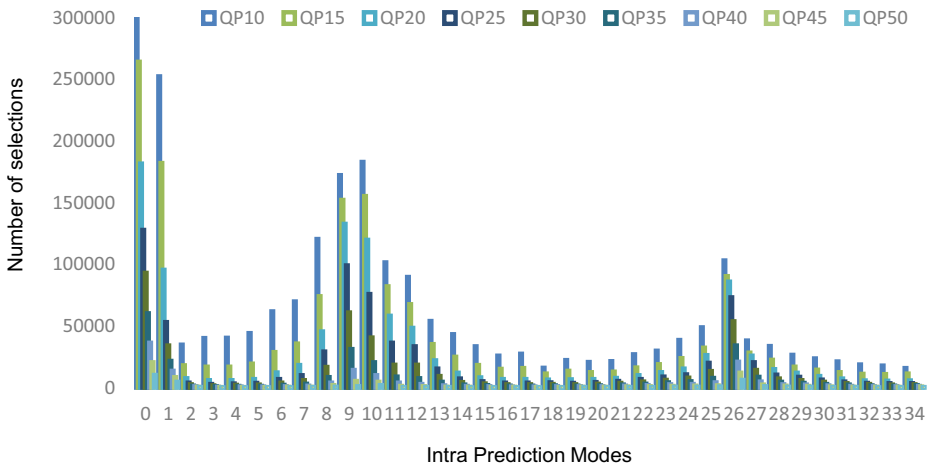
#### 3.4.1 Intra mode decision for smooth blocks

Apart from the CU size decision, it is crucial to find a rapid mechanism in order to select the intra prediction mode. For that purpose, it is important to diminish the number of intra prediction modes evaluated in smooth regions while maintaining a high quality. This simplification is based on one of our previous works [11]. Taking as stimuli typical sequences, an analysis of the most frequent modes for smooth regions was performed.

Figure 8 shows the histogram of the intra prediction modes at many QP values when using smooth CUs (*pedestrian* sequence). After studying this chart, it is possible to conclude that the most frequent modes are: mode 0 (planar), mode 1 (DC), mode 10 (horizontal) and mode 26 (vertical) regardless the QP value.

**Fig. 7** An illustration of the H264 intra prediction modes obtained for a CU ( $64 \times 64$  pixels)





**Fig. 8** Histogram of Intra prediction modes (pedestrian sequence) for different QP values

Apart from the aforementioned modes, the first most probable mode (MPM) [50] has been incorporated with the list of feasible modes, just in case it has not been previously added. As the analysis performed in [48] exposes, MPM has a mean probability close to 34% of becoming the best mode. Hence, a maximum amount of 5 modes (DC, planar, horizontal, vertical and MPM) are analysed per block.

Furthermore, since the regions labeled as smooth possess low complexity, the RDO cost function is just negligible increased. Accordingly, only SATD-based costs are used.

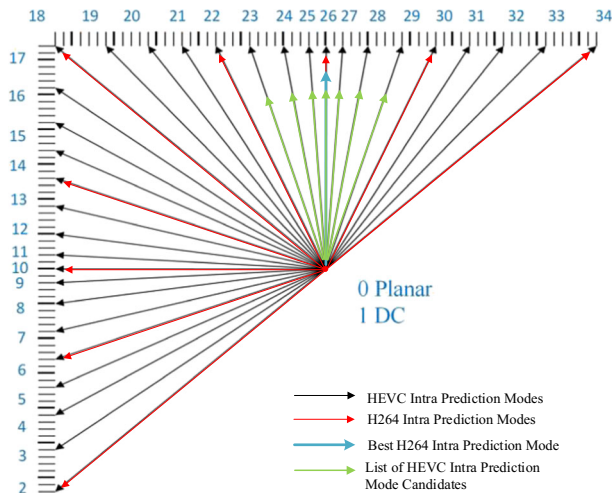
### 3.4.2 Novel intra mode decision based on H264 intra prediction mode decision

In the previous section, an optimized intra prediction mode decision is applied to smooth regions. In this section, a new improvement is suggested for determining the intra mode in spatially homogeneous regions where a spatially predominant direction exists.

As mentioned in Section 3.2.1, the H264 intra prediction obtained from *Intel's advanced motion estimation extension for OpenCL* provides the best intra prediction modes for every H264 block size. HEVC intra prediction is quite similar to H264 since the samples are predicted from the reconstructed neighboring pixels. The intra prediction mode categories remain identical: DC, planar, vertical, horizontal and directional. However, HEVC specifies 33 directional modes in contrast to the 8 directional modes specified by H.264. Figure 9 depicts how the H264 directional modes are a subset of the HEVC directional modes.

In the proposed novel intra mode decision, if a CU is spatially homogeneous, then the HEVC intra prediction mode decision is simplified by reducing from 33 to 7 the number of evaluated directional prediction modes. This set of studied modes is then composed of the first mode selected by the H264 intra prediction mode method, and the six closest HEVC intra prediction modes, which are not available for the H264 standard. Moreover, the non-directional modes (planar and DC) are always taken into account, making 9 modes overall.

Figure 9 shows the HEVC intra prediction modes which are included in the candidate list for the proposed HEVC intra prediction mode decision algorithm when the best H264 intra prediction mode is the vertical mode. In Fig. 9, the best H264 intra prediction mode is



**Fig. 9** Relationship between H264 and HEVC intra prediction modes

highlighted using a blue arrow. The surrounding six HEVC intra prediction modes, which are highlighted in green color, are also added to the candidate list. Finally, the DC and planar modes are included in the candidate list too.

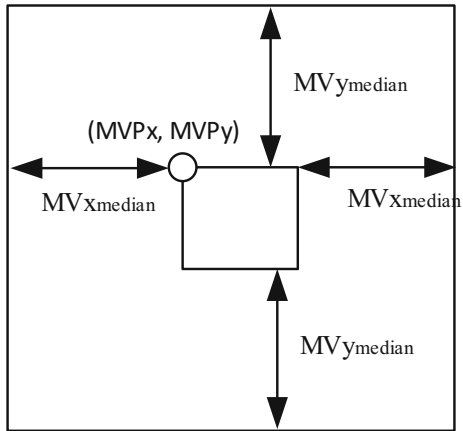
### 3.5 Inter partition mode decision

The optimized inter prediction method is not only used for stopping the quad-tree partitioning, but also used to diminish the complexity. This is done by decreasing the number of allowed inter partition modes, which are shown in Fig. 2. In order to establish this number, we use the algorithm proposed in our previous work [11]. In that work, after conducting several statistical experiments the aforementioned number is reduced from eight to one ( $2N \times 2N$  partition mode) for spatially homogeneous regions. After running several experiments based on the HM16.2 reference software, we concluded that the probability of selecting the  $2N \times 2N$  mode is very high. Once this assumption was checked after testing different sequences with several QPs, in this proposal only this mode has been considered. Finally, it is important to note that the  $2N \times 2N$  mode also includes the  $2N \times 2N$  SKIP mode.

### 3.6 Novel adaptive motion search window

When a CU is categorized as temporally homogeneous then the CU is a static region or it is an area which possesses a similar movement. Equation 1 provides an approximation of the kind of motion which the CU possesses, since all the partitions within the CU have similar motion vectors when a CU is classified as temporally homogeneous. Consequently, it is possible to determine whether or not the input video can be classified as a low or high motion sequence using the median motion vector derived from the mean absolute deviation equation. In our approach, the H264 motion vector median ( $MV_{median}$ ) obtained after analysing the input image is used to establish the motion search window when a CU is categorized as temporally homogeneous. Therefore, the motion search range will be

**Fig. 10** Adaptive motion estimation search window



$MV_{xmedian} \times MV_{ymedian}$ , centered at the motion vector predictor coordinates  $(MVP_x, MVP_y)$  as can be seen in Fig. 10.

The motion vector predictors ( $MVP_x$  and  $MVP_y$ ) are derived from the normative process specified in [31] by referring to the motion parameters of neighboring PUs. The motion estimation search process is the same as the one explained in [31] for the HM.16.2 software, but a simplified RDO process is implemented. Basically, the input image is used instead of the reconstructed image for the RDO cost calculation, which considerably reduces the motion search complexity.

### 3.7 Novel early stop condition

The H264 motion estimation and intra prediction mode decision performed by the embedded hardware provide the costs for the best H264 modes. The HEVC standard includes improved prediction methods that obtain a better intra/inter mode decision, but the H264 results can be used to implement an early stop condition to reduce the complexity of the HEVC intra/inter mode decision algorithms. If the cost of an H264 intra macroblock is very high compared to the H264 inter macroblock, it could be a good indication to skip inter prediction in the HEVC because it is likely to finally decide that HEVC intra prediction is the best prediction. Then, if  $SAD_{H264inter} > SAD_{H264intra} \times TH_{SAD}$ , inter prediction is not evaluated in our proposal by skipping the motion search and other computationally expensive processes. Since HEVC uses  $64 \times 64$  and  $32 \times 32$  coding units, which are not allowed in H264, the SAD costs for these CU sizes are obtained by adding the SAD costs of the  $16 \times 16$  blocks within the current CU.

As the case of the threshold TH described in Section 3.1, the  $TH_{SAD}$  value has been empirically found and validated in the corresponding experimental section.

## 4 Metrics

Several metrics have been proposed to estimate the performance of our flow. The execution time of the algorithm will allow us to determine what improvement is provided by the proposal in terms of speed. The quality assessment has been conducted by taking into account the following metrics:

- Peak signal-to-noise ratio (PSNR). This is the most widespread metric for quality comparison. It is the ratio between the maximum power of a signal and the power of noise. It does not take into the account the HVS.
- PSNR-HVSM [39]. This metric uses DCT coefficients and the JPEG quantization table to integrate HVS characteristics into the PSNR. Moreover, a masking model is utilized to emulate the HVS behavior.
- Structural Similarity Index (SSIM) [55]. It is based on the assumption that HVS has evolved to extract structural information. The frames to be compared are decomposed into luminance, contrast and structure. Then, a paired comparison is conducted for every component and a similarity measure is obtained by combining the results from the three component comparisons.
- Multi Scale SSIM (MS-SSIM) [57]. It is based on applying the SSIM metric to several downsampled images. The final result is the weighted average of those SSIM values.
- Three-Component Weighted Structural Similarity Index (3-Component SSIM Index, 3-SSIM) [27]. The image is divided into three types of regions: edges, textures and smooth regions. The weighted average of the SSIM metric for those regions is the final result.
- Spatial-Temporal SSIM (ST-SSIM) [33]. This metric applies SSIM to motion-oriented weighted windows to take into account temporal distortions.
- Visual Information Fidelity (VIF/VIFP) [42]. VIF is a generalization of the Information Fidelity Criterion (IFC) [43]. In IFC, the quality assessment is modeled as a transmission channel. The mutual information between the input and the output of this channel quantifies the information that HVS can perceive from the test image. VIF uses the same IFC structure, rating the quality perceived by humans. It compares mutual information between the output images of a transmission channel with a version of the same images after adding distortion to the transmission channel. VIFP is a version of VIF working in the pixel domain to reduce the computational complexity.
- Video Quality Metric (VQM) [61]. This is a modified DCT-based video quality metric based on Watson’s proposal [58].

It must be noted that all the metrics consider HVS but PSNR. Finally, as our purpose is to compare with respect to an HEVC encoder without any modification, the differences with respect to the baseline case have been computed in percentual terms. Equations 2, 3 and 4 describe these calculations. Equation 3 is applied for every quality metric, while eqs. 2 and 4 are just employed for evaluating performance. When using these equations, a negative magnitude then implies a decrease in any of these metrics. Furthermore, the Bjontegaard Delta rate (BD-rate) [54] metric has also been used to compare the quality of the proposed algorithms with respect to state-of-the-art methods.

$$\Delta\text{Time} (\%) = \frac{\text{Proposal Time} - \text{Reference Time}}{\text{Reference Time}} * 100 \quad (2)$$

$$\Delta\text{Quality}(\%) = \frac{\text{Proposal Value} - \text{Reference Value}}{\text{Reference Value}} * 100 \quad (3)$$



$$\Delta\text{Bitstream (\%)} = \frac{\text{Proposal (bytes)} - \text{Reference (bytes)}}{\text{Reference (bytes)}} * 100 \quad (4)$$

## 5 Experiments

### 5.1 Framework

Several sequences from [xiph.org](http://xiph.org) [65] and JCT-VT [4] have been used to test our proposal. As has been mentioned above, the  $\times 265$  software is used to test our approach in real-time conditions. The rate control algorithm was enabled to dynamically adjust the quantization parameter to achieve a target bitrate. The rate control is usually enabled in real-world applications since this is the only way of ensuring a constant data rate in a video transmission. Moreover, as all the methods mentioned in the literature section employ the HM software and the common test conditions to obtain their results, we have also implemented our proposal in HM to conduct a fair comparison.

The execution time is averaged after being measured three times for every sequence in order to avoid the impact of possible external interferences when encoding [9, 11]. An Intel(R) Core i7–6820@2.8GHz microprocessor has been used to perform the encoding of all the sequences, while the embedded Intel HD Graphics 530 GPU has been utilized to perform the temporal and spatial homogeneity analysis in combination with OpenCL v1.1 [22].

The quality metrics have been calculated using the *MSU Video Quality Measurement Tool* [34] and *VQMT: Video Quality Measurement Tool* [35]. The former has been used to obtain PSNR, SSIM, MS-SSIM, 3SSIM, ST-SSIM and VQM, whereas the VQMT tool provides VIFP and PSNR-HVSM.

### 5.2 Homogeneity analysis

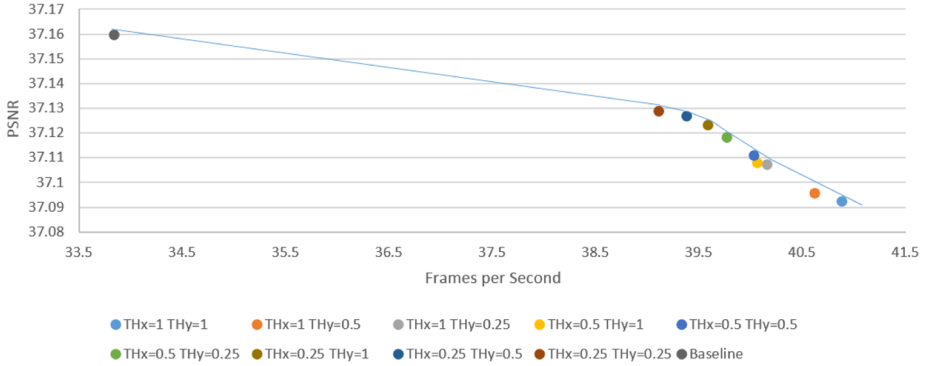
An analysis on temporal and spatial homogeneity has been carried out in this first experiment. In order to perform the spatial homogeneity process using the logic units within the GPU, we need 610  $\mu\text{s}$  on average for 720p sequences and 1.32 ms for 1080p. On the other hand, using the dedicated GPU hardware and some additional GPU logic units to analyse the output of the H264-dedicated hardware, the motion estimation process and the intra prediction mode decision take longer, as 8.1 ms and 17.4 ms are required for 720p and 1080p sequences, respectively. Therefore, for 1080p25 sequences, which allocate 40 ms to encode every frame, it is possible to execute two motion estimations at most.

#### 5.2.1 Temporal homogeneity threshold adjustment

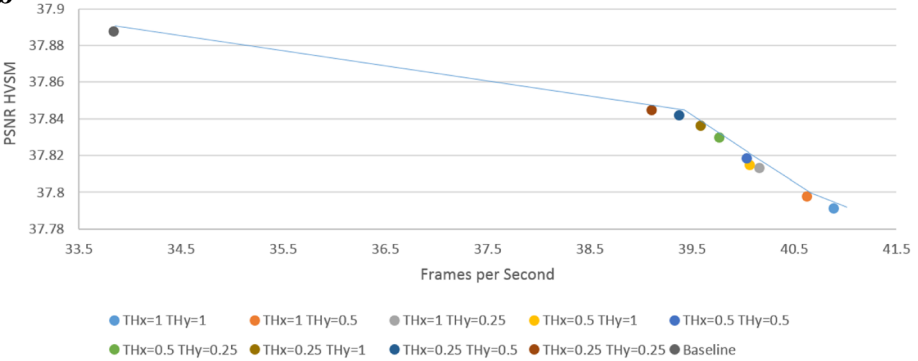
As mentioned above, the temporal homogeneity classification is based on our previous works [8, 9], but in this paper the thresholds determination leverages HVS, which provides a further improvement.

In order to study the efficiency of the approach, the Pareto curves [12] have been constructed to compare quality and speed (in fps). In this test, the aforementioned temporal homogeneity thresholds ( $TH_x$  and  $TH_y$ ) have taken different values. The objective of this study

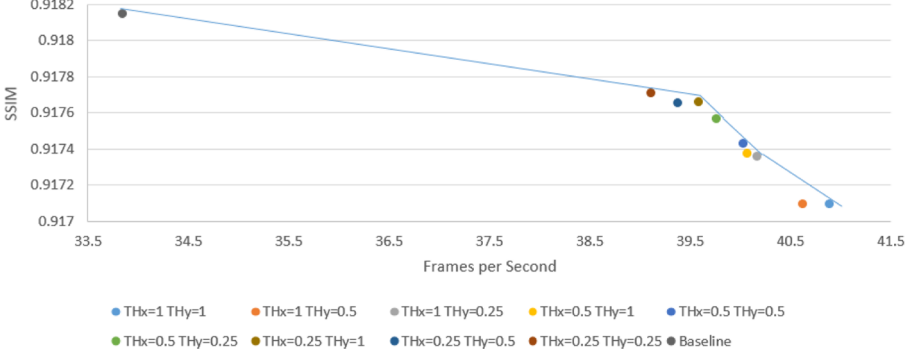
**a**



**b**



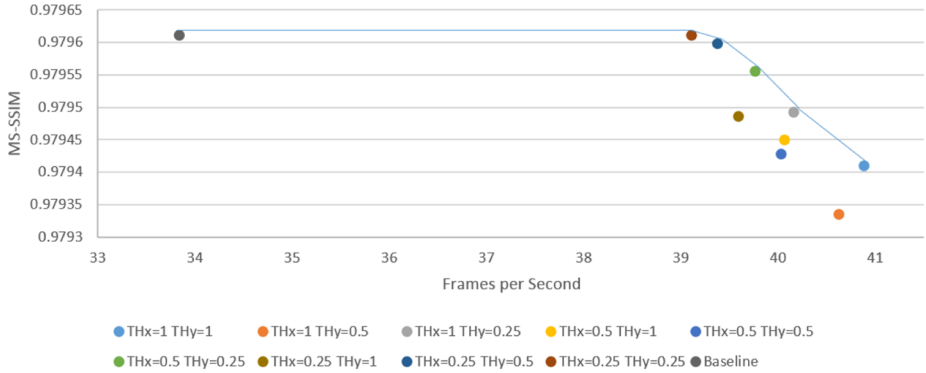
**c**



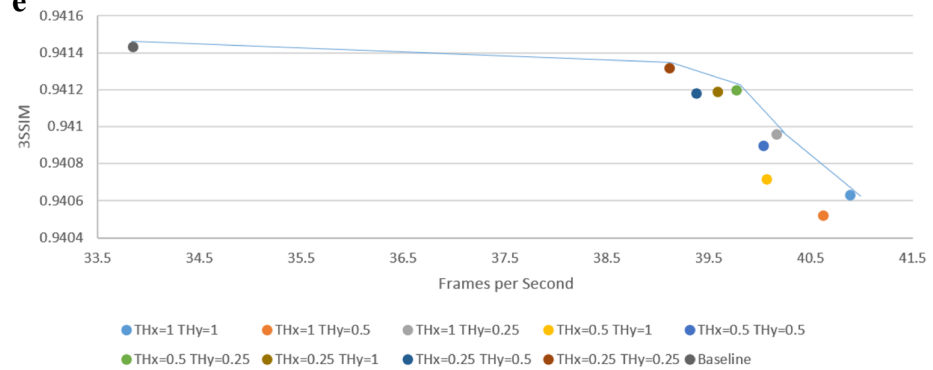
**Fig. 11** **a** Results obtained for the PSNR metric at 10Mbps using different thresholds. **b** Results obtained for the PSNR-HVSM metric at 10Mbps using different thresholds. **c** Results obtained for the SSIM metric at 10Mbps using different thresholds. **d** Results obtained for the MS-SSIM metric at 10Mbps using different thresholds. **e** Results obtained for the 3SSIM metric at 10Mbps using different thresholds. **f** Results obtained for the ST-SSIM metric at 10Mbps using different thresholds. **g** Results obtained for the VIFP metric at 10Mbps using different thresholds. **h** Results obtained for the VQM metric at 10Mbps using different thresholds

is to find the configuration that provides the best execution time reduction/quality loss trade-off. For this purpose, the values obtained from the proposed video quality metrics and the associated encoding speed have been analysed using four bitrates: 5Mbps, 10Mbps, 15Mbps

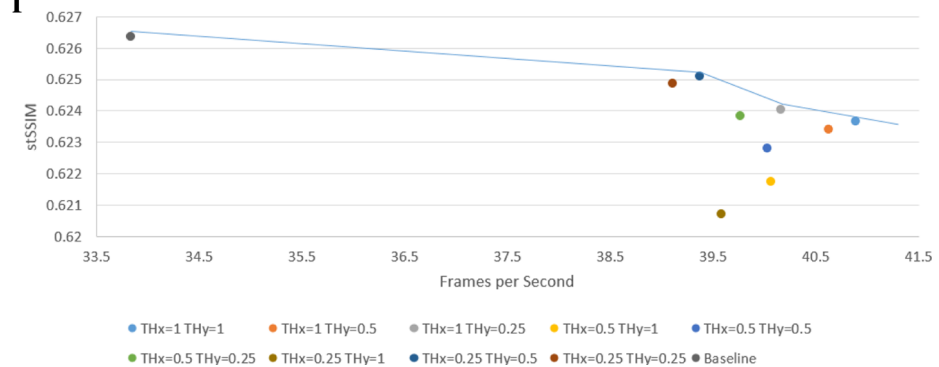
**d**



**e**



**f**



**Fig. 11** (continued)

and 20Mbps. Figure 11 shows the values obtained for 10Mbps when using different thresholds. It must be noted that this experiment is focused on the threshold adjustment for 10Mbps, but a similar procedure has been followed to obtain the final thresholds for 5Mbps, 15Mbps and 20Mbps. In Fig. 11, the best execution time reduction is obtained for the configuration that provides the highest number of fps, whereas the lowest quality loss is obtained for the configuration with the highest quality values, with the exception of the VQM metric, for

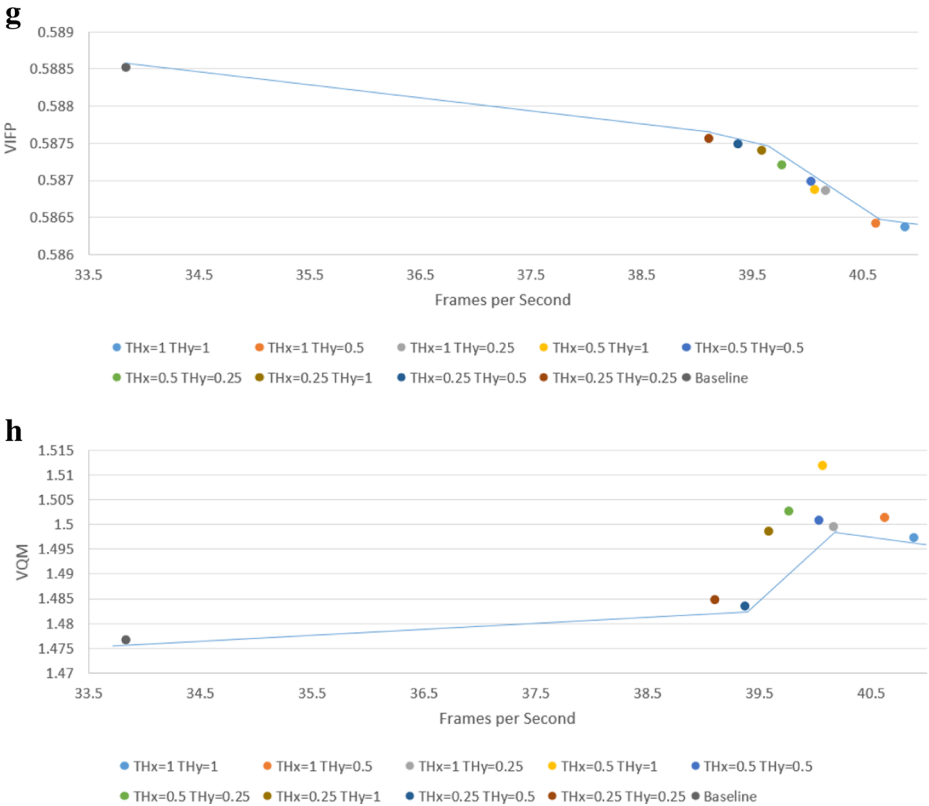


Fig. 11 (continued)

which a lower value implies better quality. The results that are shown in the charts are the average values obtained for these twelve HD sequences [65]: *aspen*, *blue\_sky*, *crowd\_run*, *ducks\_take\_off*, *in\_to\_tree*, *park\_joy*, *pedestrian*, *riverbed*, *rush\_hour*, *snow\_mountain*, *sun\_flower* and *tractor*.

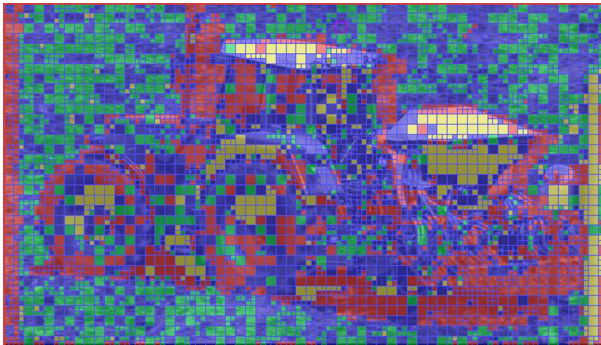
The baseline configuration is the original version of the  $\times 265$  software [60]. The rest of the configurations are obtained by varying the values of  $TH_x$  and  $TH_y$ . For instance, a  $TH_x$  value equal to 0.25 represents a quarter-pel horizontal movement, whereas 0.5 and 1 represent a half-pel and full-pel movement, respectively, and analogously for  $TH_y$ , which is the threshold for vertical motion vectors.

As can be observed, the baseline configuration provides the best quality, but at the expense of a poor performance. On the other hand, the configurations delivering the best trade-off are those belonging to the Pareto front. Table 1 summarizes the configurations that belong to the Pareto front for each quality metric in the case of using 10Mbps. As can be seen in Table 1, it is worth mentioning that the configurations  $TH_x = 1/TH_y = 1$  and  $TH_x = 1/TH_y = 0.25$  belong to the Pareto Front for all the quality metrics.

Figure 11 shows that the  $TH_x = 1/TH_y = 1$  configuration provides a worse quality than  $TH_x = 1/TH_y = 0.25$  for several metrics. Therefore,  $TH_x = 1/TH_y = 0.25$  is the best option for this bitrate since this configuration provides an excellent trade-off. Figure 11 also shows how the individual Pareto analysis for some metrics such as PSNR does not allow the

**Table 1** Points that belong to the Pareto front in quality/FPS space for 10Mbps

10Mbps	THx = 1	THx = 1	THx = 1	THx = 0.5	THx = 0.5	THx = 0.5	THx = 0.25	THx = 0.25	THx = 0.25	THx = 0.25
	THy = 1	THy = 0.5	THy = 1	THy = 1	THy = 0.5	THy = 0.5	THy = 1	THy = 0.25	THy = 0.5	THy = 0.25
3SSIM	X		X			X		X		X
MSSIM	X		X			X		X	X	X
PSNR	X		X		X	X	X	X	X	X
PSNRHVSM	X	X	X	X	X	X	X	X	X	X
SSIM	X		X			X		X		X
S4SSIM	X		X			X		X	X	X
VIFP	X	X	X	X	X	X	X	X	X	X
VQM	X		X		X	X	X	X	X	X



**Fig. 12** Texture classification after applying the proposal

discarding any configuration, since almost all the points belong to the Pareto front, whereas others such as VQM or ST-SSIM make it easier to determine the dominant points.

### 5.3 Testing the whole flow

In this experiment, the whole flow shown in Fig. 4 is evaluated. In general, the areas presenting spatial and temporal homogeneity have selected large blocks as preferred modes. As an instance of this issue, Fig. 12 depicts the outcome of our CU size decision algorithm, which

**Table 2** Results of our previous works in a real time environment

Sequence	$\Delta$ Time (%)	$\Delta$ 3SSIM (%)	$\Delta$ ST-SSIM (%)	$\Delta$ MSSIM (%)	$\Delta$ VQM (%)	$\Delta$ PSNR (%)	$\Delta$ PSNR HVSM (%)	$\Delta$ VIFP (%)	$\Delta$ SSIM (%)
Aspen 1080p30	-18.67	0.10	0.04	0.04	-0.32	-0.03	-0.06	0.04	0.01
BlueSky 1080p25	-19.37	0.05	-0.16	-0.01	0.72	-0.15	-0.18	-0.24	-0.02
CrowdRun 1080p50	-11.16	-0.12	-0.34	-0.02	0.38	-0.14	-0.18	-0.28	-0.08
DucksTakeOff 1080p50	-7.95	-0.06	-0.20	-0.02	0.20	-0.03	-0.04	-0.03	-0.04
InToTree 1080p50	-25.88	-0.29	-1.36	-0.07	1.44	-0.28	-0.44	-1.04	-0.26
ParkJoy 1080p50	-24.21	-0.16	-1.01	-0.05	0.85	-0.26	-0.34	-0.51	-0.13
PedestrianArea 1080p25	-12.81	-0.01	-0.06	0.00	2.58	-0.03	-0.05	-0.07	-0.01
Riverbed 1080p25	-8.06	0.00	0.00	0.03	0.01	0.00	0.00	0.00	0.00
RushHour 1080p25	-9.20	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00
SnowMountain 1080p25	-34.84	-0.02	-0.37	0.00	0.77	-0.15	-0.04	-0.21	-0.04
Sunflower 1080p25	-10.30	0.00	-0.21	0.01	2.86	-0.01	-0.01	-0.01	0.00
Tractor 1080p25	-11.77	-0.03	-0.06	0.00	-0.01	-0.05	-0.08	-0.10	-0.01
Average	-16.19	-0.05	-0.31	-0.01	0.80	-0.09	-0.12	-0.20	-0.05

**Table 3** Results for the entire proposal

Sequence	$\Delta$ Time (%)	$\Delta$ 3SSIM (%)	$\Delta$ ST-SSIM (%)	$\Delta$ MSSIM (%)	$\Delta$ VQM (%)	$\Delta$ PSNR (%)	$\Delta$ PSNR HVSM (%)	$\Delta$ VIFP (%)	$\Delta$ SSIM (%)
Aspen 1080p30	-27.74	-0.11	-0.1	-0.14	0.12	-0.17	-0.26	-0.84	-0.09
BlueSky 1080p25	-27.51	-0.07	-0.21	-0.2	0.92	-0.37	-0.39	-1.3	-0.04
CrowdRun 1080p50	-19.22	-0.23	-0.64	-0.08	0.51	-0.21	-0.26	-1.98	-0.18
DucksTakeOff 1080p50	-12.12	-0.08	-0.31	-0.03	0.34	-0.24	-0.26	-0.04	-0.06
InToTree 1080p50	-32.18	-0.31	-1.86	-0.12	1.5	-0.61	-0.79	-1.84	-0.30
ParkJoy 1080p50	-34.11	-0.18	-1.32	-0.11	0.99	-0.39	-0.47	-0.71	-0.16
PedestrianArea 1080p25	-19.93	-0.1	-1.19	-0.1	2.78	-0.49	-0.51	-0.97	-0.09
Riverbed 1080p25	-19.46	-0.01	-0.11	0.00	0.76	-0.22	-0.24	-0.03	-0.01
RushHour 1080p25	-16.33	-0.21	-1.01	-0.06	0.12	-0.39	-0.40	-0.04	-0.2
SnowMountain 1080p25	-42.92	-0.17	-0.77	-0.09	1.23	-0.21	-0.13	-0.71	-0.17
Sunflower 1080p25	-17.84	-0.12	-1.1	-0.07	3.16	-0.11	-0.12	-0.53	-0.13
Tractor 1080p25	-15.99	-0.09	-0.86	0.00	0.51	-0.07	-0.12	-0.22	-0.07
Average	-23.78	-0.14	-0.79	-0.08	1.08	-0.29	-0.33	-0.77	-0.13

considers the analysis of spatially and temporally homogeneous regions, after being applied to the second frame of the *tractor* sequence.

Table 2 shows the results when combining all the optimizations proposed in our previous works [7–11] implemented in the  $\times 265$  real time encoder in terms of *execution time*.

With the intention of reducing the size of Table 2, an average of the results for different bitrate values is shown: 5Mbps, 10Mbps, 15Mbps and 20Mbps.

As can be seen in Table 2, the average quality losses are negligible. On the other hand, the execution time is diminished by an amount that ranges from 7.95% to 34.84%, depending on the number of blocks categorized as temporally or spatially homogeneous.

Table 3 shows the results for the entire proposal after combining the new set of novel optimizations presented in this paper with our previous works. As mentioned above, these optimizations consist of improved CU size decision algorithms, new intra and inter mode decisions methods, a new early stop condition for the selection between inter and intra prediction and a novel adaptive motion search window. As can be observed, the execution time reduction ranges from 16% to 42.9% (23.8% on average), while providing a negligible decrease in terms of quality. In order to complement this study, a subjective comparison has been performed using the Mean Opinion Score (MOS), ITU-R BT.500 [18] for the worst quality loss (2.78% for the *pedestrian* sequence, using the VQM metric), observing that this loss is almost imperceptible.

**Table 4** Performance comparison with state-of-the-art methods for the all intra configuration

Seq.	[70]	[19]	[49]	[41]	[28]	[48]	[69]	[29]	Proposal									
	BD-Rate $\Delta T$	BD-Rate $\Delta T$	BD-Rate $\Delta T$	BD-Rate $\Delta T$	BD-Rate $\Delta T$	BD-Rate $\Delta T$	BD-Rate $\Delta T$	BD-Rate $\Delta T$	BD-Rate $\Delta T$	BD-Rate $\Delta T$								
Class A	0.85	-54.8	0.55	-20.4	1.14	-50	0.39	-37.8	1.41	-35	2.28	-21.9	1.15	-59	0.80	-56.2	0.53	-53.71
Class B	0.69	-58.7	0.70	-20.5	1.26	-53.2	0.56	-40.6	0.79	-24.3	2.16	-28.6	0.94	-61	0.72	-59.4	1.05	-52.82
Class C	0.80	-47.2	0.74	-19.5	0.60	-34.2	0.84	-37.3	1.85	-33.6	1.50	-17.8	1.08	-56.5	1.18	-50.7	0.88	-46.39
Class D	1.37	-46.9	0.94	-19.5	0.25	-34.5	0.84	-35.5	1.86	-31.1	1.19	-14.5	1.00	-54.7	1.13	-49.7	0.55	-44.13
Class E	1.05	-65.2	X	X	2.43	-62.7	0.65	-39	X	X	X	X	1.53	-64	1.03	-65.2	1.16	-61.74
Average	0.95	-54.6	0.73	-20	1.14	-46.9	0.66	-38	1.48	-31	1.78	-20.7	1.14	-59	0.97	-56.2	0.83	-51.76



**Table 5** Performance comparison with state-of-the-art methods for the low delay P configuration

Method	Metric	Class A	Class B	Class C	Class D	Average
[64]	BD-Rate	2.04	2.05	2.15	1.56	1.95
	$\Delta T$	-44.59	-47.04	-36.53	-32.14	-40.07
[47]	BD-Rate	1.56	1.18	2.68	0.77	1.55
	$\Delta T$	-42.73	-40.98	-34.90	-26.65	-36.31
[63]	BD-Rate	2.06	2.94	1.73	1.47	2.05
	$\Delta T$	-58.75	-62.90	-53.60	-52.85	-57.03
[1]	BD-Rate	1.56	1.34	1.24	1.07	1.30
	$\Delta T$	-37.25	-45.68	-34.35	-32.25	-37.38
[62]	BD-Rate	1.50	1.97	1.61	2.08	1.79
	$\Delta T$	-50.70	-55.13	-45.53	-42.40	-48.44
[29]	BD-Rate	X	1	0.8	1.3	1.03
	$\Delta T$	X	-20.9	-18	-17.2	-18.7
[69]	BD-Rate	X	0.7	1	0.8	0.83
	$\Delta T$	X	-15	-14	-13	-14
Proposal	BD-Rate	1.3	1.41	1.09	1.34	1.28
	$\Delta T$	-40.3	-47.52	-35.1	-37.6	-45.13

## 5.4 Performance comparison

In addition to the prior experiments, the proposed encoding flow shown in Fig. 4 is compared with the state-of-the-art approaches. In order to do this, both the BD-rate [3] and  $\Delta T$  (Time) have been measured after encoding with the HEVC test model HM16.2. The HM software is still necessary to conduct a fair comparison because all the methods referred in Section 2, including most of our previous works [8–11], were implemented on the HEVC test model. In order to encode the sequences, the common test conditions suggested in [4] have been employed. It must be noted that, following the recommendations by JCT-VT [4], on the one hand the rate control algorithm has been disabled and, on the other, just four QP values (22, 27, 32 and 37) have been studied. Nevertheless, in order to provide a compact result, Tables 4 to 8 just show the average of the four encodings possessing different QP values.

The HM16.2 reference software provides the configuration files for encoding. Intra slices are only applied in the *all intra* coding, so the temporal redundancy is not used. On the other hand, in the *low-delay P* coding type only the opening frame in a video sequence is encoded as an intra slice. The following frames are then encoded using P-slices, which only employ the preceding slices in display order as reference.

Table 4 to 7 show the BD-Rate and the  $\Delta T$  for every JCT-VT sequence class using the *all intra* and *low delay P* configurations, respectively. Tables 4 and 5 contain the data regarding

**Table 6** Performance comparison with our former works for the all intra configuration

Seq.	[9]		[11]		Proposal	
	BD-Rate	$\Delta T$	BD-Rate	$\Delta T$	BD-Rate	$\Delta T$
Class A	0.37	-29.3	0.40	-41.3	0.53	-53.71
Class B	0.72	-30.1	0.82	-42.3	1.05	-52.82
Class C	0.49	-20.5	0.53	-33.5	0.88	-46.39
Class D	0.3	-19.7	0.30	-32.5	0.55	-44.13
Class E	0.67	-41	0.87	-53.1	1.16	-61.74
Average	0.51	-28.12	0.58	-40.5	0.83	-51.76

**Table 7** Performance comparison with our previous works for the low delay P configuration

Seq.	[9]		[11]		Proposal	
	BD-Rate	$\Delta T$	BD-Rate	$\Delta T$	BD-Rate	$\Delta T$
Class A	1.23	-36.8	1.1	-26.79	1.3	-40.3
Class B	1.37	-46.62	1.14	-32.02	1.41	-47.52
Class C	1.07	-36.32	0.68	-21.68	1.09	-35.1
Class D	1.3	-37.18	0.58	-20.6	1.34	-37.6
Average	1.24	-39.23	0.87	-25.27	1.28	-45.13

state-of-the-art approaches, while Tables 6 and 7 show the improvement with respect to our prior proposals [9, 11]. As Table 4 and 6 indicate, our flow clearly outperforms the algorithms proposed in [48, 49] and [28] in terms of both BD-Rate and  $\Delta T$  for *all intra* configuration. The average execution time reduction is 51.8%. As observed in Tables 5 and 7, the proposal also outperforms the methods proposed in [47, 64] and [1] for the *low delay P* configuration, obtaining an average 45.1% execution time reduction. However, for remaining configurations it is difficult to know which proposal is the best, since every method delivers a different  $\Delta T$ /BD-Rate trade-off. Hence, the Pareto fronts [12] have been constructed in Figs. 13 and 14 to identify the best trade-offs for both *all intra* and *low delay P* configurations, respectively. As it shown, the results provided by our flow always belong to the Pareto front for both configurations. Therefore, we can conclude that our proposal is an efficient flow. Table 8 includes a performance comparison between our approach and the HM16.2 in terms of bitstream size and PSNR. As can be seen in the tables, the proposed algorithms have considerably reduced the encoding time at the expense of a negligible bitstream increase and without noticeable quality losses.

**Table 8** Performance comparison with HM16.2 test model in terms of  $\Delta$ PSNR and  $\Delta$ Bitstream size

Class	Sequences	All Intra		Low Delay P	
		$\Delta$ PSNR (dB)	$\Delta$ Bitstream (%)	$\Delta$ PSNR (dB)	$\Delta$ Bitstream (%)
A	Traffic	-0.01	0.37	-0.02	0.22
	PeopleOnStreet	-0.03	0.08	-0.03	0.69
B	Kimono	-0.004	0.68	-0.01	0.43
	ParkScene	-0.02	0.21	-0.03	0.56
	Cactus	-0.03	0.64	-0.02	0.62
	BasketballDrive	-0.04	0.55	-0.03	0.61
	BQTerrace	-0.07	-0.7	-0.02	0.50
C	BasketBallDrill	-0.04	0.69	-0.06	0.72
	BQMall	-0.03	0.21	-0.03	0.42
	PartyScene	-0.02	0.46	-0.01	0.05
	RaceHorses	-0.01	0.92	-0.01	0.71
D	BasketballPass	-0.04	0.18	-0.04	0.41
	BQSquare	-0.06	0.04	-0.02	0.51
	BlowingBubbles	-0.14	0.02	-0.007	1.07
	RaceHorses	-0.03	0.03	-0.04	1.59
E	FourPeople	-0.04	0.31	-0.02	0.32
	Johnny	-0.03	0.66	-0.03	0.46
	KristenAndSara	-0.06	0.43	-0.02	0.34
Average	-0.04	0.32	-0.03	0.57	

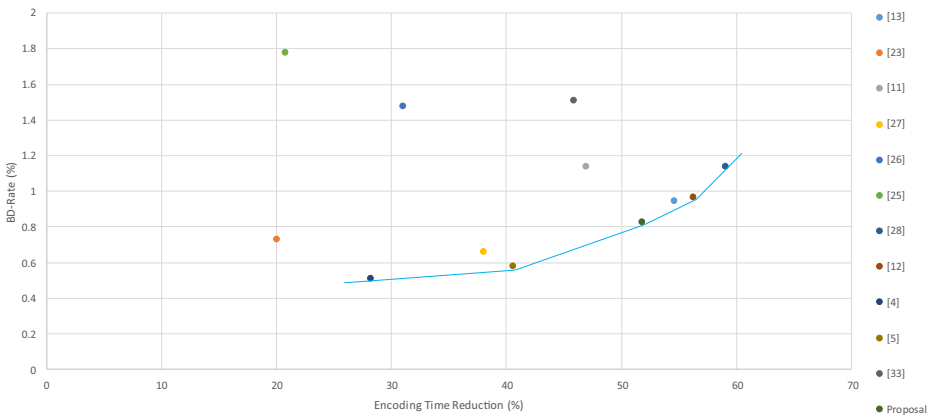


Fig. 13 Pareto analysis for the all intra configuration

## 6 Conclusion

In this paper, an HEVC-based encoding flow suited for real time is presented. The main objective of the proposal is to decrease the inherent HEVC encoding complexity leveraging both spatial and temporal homogeneity analyses, which are performed by a GPU directly on the input frames, thus offloading the processor that is running the encoding process. The motion calculation demanded for temporal homogeneity detection and the H264 intra prediction mode decision used for spatial homogeneity detection are performed on dedicated hardware belonging to an Intel GPU. In this way, the GPU logic units are not executing those processes and then become accessible for other tasks as the analysis of the retrieved data as well as the smooth analysis.

Thus, the outcome of this analysis is used to determine whether or not the proposed fast versions of the HEVC methods can be applied, resulting in a fast encoder with a negligible quality loss. These ideas have been implemented in a real-time software to verify that the proposed approach can be deployed in a multithreaded system. The optimum thresholds for temporal homogeneity detection and the adaptive motion search have been empirically obtained after studying a wide range of quality metrics based on HVS. Finally, the proposal

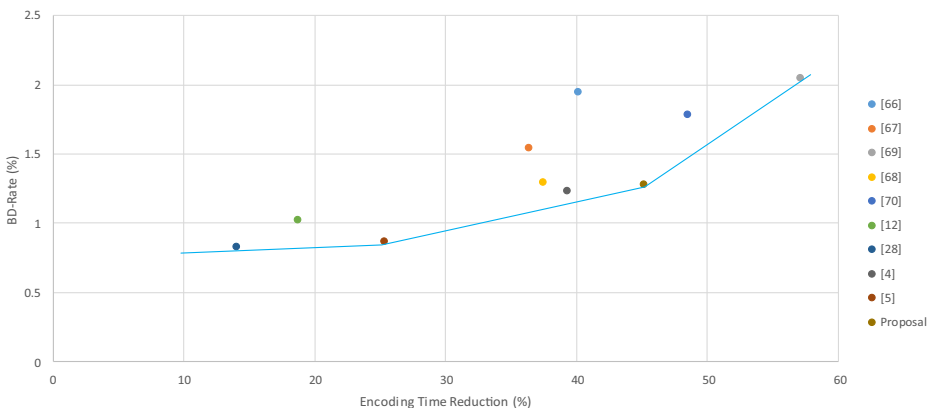


Fig. 14 Pareto analysis for the low-delay P configuration

has been compared with state-of-the-art methods, always delivering one of the best trade-offs in terms of BD-rate and execution time.

Future work will involve the evaluation of more fast algorithms for further complexity reductions based on the data retrieved from the H264 dedicated hardware belonging to Intel GPUs.

**Acknowledgments** This work has been partially supported by Spanish research projects TIN 2015-65277-R and TIN-2012-32180, as well as the UCM-Banco Santander Grant PR26-16/20B-1.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## References

- Ahn S, Lee B, Kim M (2015) A novel fast CU encoding scheme based on spatio-temporal encoding parameters for HEVC inter coding. *IEEE Trans Circuits Syst Video Technol* 25(3):422–435
- Alcocer E, Gutierrez R, Lopez-Granado O, Malumbres MP (2016) Design and implementation of an efficient hardware integer motion estimator for an HEVC video encoder. *Journal of Real-Time Image Processing*
- Bjontegard G (2001) Calculation of average PSNR differences between RD curves. ITU-T SC16/Q6 13th VCEG meeting, Austin
- Bossen F (2012) Common test conditions and software reference configurations. JCT-VC Document, JCTVC-K1100
- Cho S, Kim M (2013) Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding. *IEEE Trans Circuits Syst Video Technol* 23(9):1555–1564
- de Frutos-López M, Orellana-Quirós D, Pujol-Alcolado JC, de María FD (2010) An improved fast mode decision algorithm for intraprediction in H.264/AVC video coding. *Signal Process Image Commun* 25(10):709–716
- Fernandez DG, Del Barrio AA, Botella G, Garcia C (2016) 4K-based intra and inter prediction techniques for HEVC. *Proc. SPIE, Real-Time Image and Video Processing 2016*, 98970B
- Fernández DG, Del Barrio AA, Botella G, Garcia C (2016) Fast CU size decision based on temporal homogeneity detection. In: *Design of Circuits and Integrated Systems (DCIS), 2016 Conference on*, 1–6
- Fernández DG, Del Barrio AA, Botella G, Garcia C (2018) Fast and effective CU size decision based on spatial and temporal homogeneity detection. *Multimedia Tools and Applications* 77(5):5907–5927
- Fernández DG, Del Barrio AA, Botella G, Garcia C, Meyer-Baese U, Meyer-Baese A (2016) HEVC optimizations for medical environments. In: *Proc. SPIE 9871, Sensing and Analysis Technologies for Biomedical and Cognitive Applications 2016*, 98710B
- Fernández DG, Del Barrio AA, Botella G, Garcia C, Prieto M, Hermida R (2018) Complexity reduction in the HEVC/H265 standard based on smooth region classification. *Digital Signal Processing* 73:24–39
- D. G. Fernández, A. A. Del Barrio, Guillermo Botella, Uwe Meyer-Baese, Anke Meyer-Baese, Christos Grecos (2017) Information fusion based techniques for HEVC. *Proc. SPIE 10223, Real-Time Image and Video Processing 2017*, 102230M. 10.1117/12.2262604
- Fraunhofer Institute for Telecommunications (2018) Perceptually optimized video coding. Retrieved from: <https://www.hhi.fraunhofer.de/en/departments/vca/research-groups/image-video-coding/research-tops/perceptually-optimized-video-coding.html>. Accessed 17 Dec 2018
- Goswami K, Lee J-H, Jang K-S, Kim B-G, Kwon K-K (2014) Entropy difference-based early skip detection technique for high efficiency video coding. *Journal of Real-Time Image Processing*
- He G, Zhou D, Goto S (2013) Transform-based fast mode and depth decision algorithm for HEVC intra prediction. *IEEE 10th International Conference on ASIC (ASICON)*, pp. 1–4
- Intel Corporation (2016) Introduction to advance motion extension for OpenCL. Retrieved from: <https://software.intel.com/en-us/articles/intro-to-advanced-motion-estimation-extension-for-opencl>. Accessed 17 Dec 2018
- Intel Corporation White Paper (2012) Performance Interactions of OpenCL\* Code and Intel® Quick Sync Video on Intel® HD Graphics 4000

18. ITU-R BT.500 (2012) Methodology for the subjective assessment of the quality of television pictures. International Telecommunication Union, Geneva
19. Jiang W, Ma H, Chen Y (2012) Gradient based fast mode decision algorithm for intra prediction in HEVC. In: 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), pp. 1836–1840
20. Jiang W, Ma H, Chen Y (2012) Gradient Based Fast Mode Decision Algorithm for Intra Prediction in HEVC. In: International Conference on Consumer Electronics, Communications and Networks (CECNet)
21. Khan MUK, Shafique M, Grellert M, Henkel J (2013) Hardware-software collaborative complexity reduction scheme for the emerging HEVC intra encoder. Design, Automation Test in Europe Conference Exhibition (DATE) 2013:125–128
22. Khronos OpenCL Working Group (2011) The OpenCL specification version 1.1. Revision 44
23. Khronos OpenCL Working Group (2016) Online documentation for cl\_intel\_advanced\_motion\_estimation. Retrieved from: [https://www.khronos.org/registry/cl/extensions/intel/cl\\_intel\\_advanced\\_motion\\_estimation.txt](https://www.khronos.org/registry/cl/extensions/intel/cl_intel_advanced_motion_estimation.txt). Accessed 17 Dec 2018
24. Koumaras H, Kourtis M, Martakos D (2012) Benchmarking the encoding efficiency of h.265/HEVC and h.264/AVC,” in Future Network Mobile Summit (FutureNetw)
25. Leal da Silva T, Agostini LV, da Silva Cruz LA (2015) Fast intra prediction algorithm based on texture analysis for 3D-HEVC encoders” in Journal of Real-Time Image Processing
26. Lee JH, Park CS, Kim BG, Jun DS, Jung SH, Choi JS (2013) Novel fast PU decision algorithm for the HEVC video standard”, IEEE International Conference on Image Processing (ICIP)
27. Li C, Bovik AC (2009) Three-component weighted structural similarity index. Proc. SPIE 7242–72420, Image Quality and System Performance VI
28. Lim K, Lee J, Kim S, Lee S (2015) Fast PU skip and split termination algorithm for HEVC intra. IEEE Transactions on Circuits Systems for Video Technology 25(8)
29. Liu X, Liu Y, Wang P, Lai C, Chao H (2016) An adaptive mode Decision algorithm based on video texture characteristics for HEVC intra prediction. IEEE Transactions on Circuits Systems for Video Technology, vol. PP: 99
30. Mallikarachchi T, Fernando A, Arachchi H (2014) Efficient coding unit size selection based on texture analysis for HEVC intra prediction. IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6
31. McCann K, Rosewame C, Bross B, Naccari M, Sharman K, Sullivan G (2014) High efficiency video coding (HEVC) encoder description 0v16 (HM16). JCT-VC High Efficiency Video Coding N14 703
32. Min B, Cheung R (2014) A fast cu size decision algorithm for HEVC intra encoder. IEEE Transactions on Circuits and Systems for Video Technology PP(99):1
33. Moorthy AK, Bovik AC (2010) Efficient motion weighted spatio-temporal video SSIM Index. Proc. SPIE 7527–75271, Human Vision and Electronic Imaging XV
34. MSU Graphics & Media Lab (Video Group) (2016) MSU video quality measurement tool. Retrieved from: [http://www.compression.ru/video/quality\\_measure/video\\_measurement\\_tool.html](http://www.compression.ru/video/quality_measure/video_measurement_tool.html). Accessed 17 Dec 2018
35. Multimedia Signal Processing Group (MMSPG) (2016) VQMT: video quality measurement tool. Retrieved from: <http://mmspg.epfl.ch/vqmt>. Accessed 17 Dec 2018
36. Na S, Lee W, Yoo K (2014) Edge-based fast mode decision algorithm for intra prediction in HEVC. IEEE International Conference on Consumer Electronics (ICCE), pp. 11–14
37. Öztekin A, ErÇelebi E (2015) An early split and skip algorithm for fast intra CU selection in HEVC. In: Journal of Real-Time Image Processing
38. Pastuszak G, Trochimiuk M (2015) Algorithm and architecture design of the motion estimation for the H.265/HEVC 4K-UHD encoder. Journal of Real-Time Image Processing
39. Ponomarenko N, Silvestri F, Egiazarian K, Carli M, Astola J, Lukin V (2007) On Between-Coefficient Contrast Masking of DCT Basis Functions. Third International Workshop on Video Processing and Quality Metrics (VPQM), Scottsdale
40. Ramezanpour M, Zargari F (2016) Fast HEVC I-frame coding based on strength of dominant direction of CUs. Journal of Real-Time Image Processing
41. Shang X, Wang G, Fan T, Li Y (2015) Fast CU size decision and PU mode decision algorithm in HEVC intra coding. IEEE International Conference on Image Processing (ICIP)
42. Sheikh HR, Bovik AC (2006) Image Information and Visual Quality. IEEE Trans Image Process 15(2):430–444
43. Sheikh HR, Bovik AC, de Veciana G (2005) An Information Fidelity Criterion for Image Quality Assessment Using Natural Scene Statistics. IEEE Trans Image Process 14(12):2117–2128
44. Shen L, Liu Z, Liu S, Zhang Z, An P (2009) Selective disparity estimation and variable size motion estimation based on motion homogeneity for multi-view coding. IEEE Transactions on Broadcasting 55(4)

45. Shen L, Liu Z, Yan T, Zhang Z, An P (2010) View-adaptive motion estimation and disparity estimation for low complexity multiview video coding. *IEEE Transactions on Circuits and Systems for Video Technology* 20(6)
46. Shen L, Liu Z, Zhang Z, Shi X (2008) Fast Inter Mode Decision Using Spatial Property of Motion Field. *IEEE Transactions on Multimedia* 10(6):1208–1214
47. Shen L, Liu Z, Zhang X, Zhao W, Zhang Z (2013) An effective CU size decision method for HEVC encoders. *IEEE Transactions on Multimedia* 15(2):465–470
48. Shen L, Zhang Z, An P (2013) Fast CU size decision and mode decision algorithm for HEVC intra coding. *IEEE Transactions on Consumer Electronics* 59(1)
49. Shen L, Zhang Z, Liu Z (2014) Effective CU size decision for HEVC intracoding. *IEEE Trans Image Process* 23(10):4232–4241
50. Sullivan G, Ohm J, Han W-J, Wiegand T (2012) Overview of the high efficiency video coding (hevc) standard. *IEEE Trans Circuits Syst Video Technol* 22(12):1649–1668
51. Sze V, Budagavi M, Sullivan GJ (2014) *High Efficiency Video Coding (HEVC)*, 1st ed. Springer International Publishing. Available: <https://link.springer.com/book/10.1007%2F978-3-319-06895-4>. Accessed 17 Dec 2018
52. Tian G, Goto S (2012) Content adaptive prediction unit size decision algorithm for HEVC intra coding.” in *Picture Coding Symposium (PCS)*, pp. 405–408
53. Ting Y-C, Chang T-S (2014) Gradient-based PU size selection for HEVC intra prediction. *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1929–1932
54. Vanne J, Viitanen M, Hämäläinen TD (2014) Efficient mode decision schemes for HEVC inter prediction. *IEEE Transactions on Circuits and Systems for Video Technology* 24(9):1579–1593
55. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans Image Process* 13(4):600–612
56. Wang H-M, Lin J-K, Yang J-F (2006) Fast inter mode decision based on hierarchical homogeneous detection and cost analysis for h.264/AVC coders. In: *IEEE International Conference on Multimedia and Expo*, pp. 709–712
57. Wang Z, Simoncelli EP, Bovik AC (2003) Multi-scale structural similarity for image quality assessment. In: *Proceedings of the 37th IEEE Asiloma Conference on Signal, Systems and Computers*, Pacific Grove
58. Watson AB (1998) Toward a perceptual video quality metric. *Human Vision, Visual Processing, and Digital Display VIII*(3299):139–147
59. Wu D, Pan F, Lim K, Wu S, Li Z, Lin X, Rahardja S, Ko C (2005) Fast intermode decision in h.264/AVC video coding. *IEEE Transactions on Circuits and Systems for Video Technology* 15(7):953–958
60. x265 project (2017) <http://x265.org/>. GNU GPL 2 license, source code available at: <https://bitbucket.org/multicoreware/x265/wiki/Home>. Accessed 17 Dec 2018
61. Xiao F (2000) DCT-based Video Quality Evaluation—Final Project for EE392J
62. Xiong J, Li H, Meng F, Wu Q, Ngan KN (2015) Fast HEVC inter CU decision based on latent SAD estimation. *IEEE Transactions on Multimedia* 17(12):2147–2159
63. Xiong J, Li H, Meng F, Zhu S, Wu Q, Zeng B (2014) MRF-based fast HEVC inter CU decision with the variance of absolute differences. *IEEE Trans Multimedia* 16(8):2141–2153
64. Xiong J, Li H, Wu Q, Meng F (2014) A fast HEVC inter CU selection method based on pyramid motion divergence. *IEEE Transactions on Multimedia* 16(2):559–564
65. xiph.org (2016) Derf’s test media collection. Retrieved from: <https://media.xiph.org/video/derf/>. Accessed 17 Dec 2018
66. Ye T, Zhang D, Dai F, Zhang Y (2013) Fast mode decision algorithm for intra prediction in HEVC. in *Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service*, pp. 300–304
67. Ye T, Zhang D, Dai F, Zhang Y (2013) Fast mode decision algorithm for intra prediction in HEVC. in the *Fifth International Conference on Internet Multimedia Computing and Service (ICIMCS’13)*
68. Zhang Y, Li Z, Li B (2012) Gradient-based fast decision for intra prediction in HEVC. In *Visual Communications and Image Processing (VCIP)*, pp. 1–6
69. Zhang H, Ma Z (2014) Fast Intra mode decision for high efficiency video coding (HEVC). *IEEE Transactions on Circuits Systems for Video Technology* 24(4):660–668
70. Zhang H, Ma Z (2016) Fast intra mode and CU size decision for HEVC. *IEEE Transactions on Circuits Systems for Video Technology* PP(99):1–7



**D. G. Fernández** received the B.Sc degree in Telecommunications Engineering from the Technical University of Madrid (Madrid, Spain) and the M.Sc degree in Computer Science Security from the International University of La Rioja (La Rioja, Spain), in 2003 and 2014, respectively. Since 2003 he has been a Video Group Engineer at PRODYS designing, implementing and optimizing video codecs for DSPs and FPGAs. He is currently a PhD student at Complutense University of Madrid (UCM) (Madrid, Spain). His current research interests include HEVC video coding, compression and embedded systems.



**Guillermo Botella** received the M.A. Sc. degree in Physics in 1998, the [M.A.Sc.](#) degree in Electronic Engineering in 2001 and the Ph.D. degree in 2007, all from the University of Granada, Spain. He was a research fellow funded by EU working Vision Research Laboratory at University College London, UK. After that he joined as Assistant Professor at the Department of Computer Architecture and Automation of Complutense University of Madrid, Spain. His current research interests include Digital Signal Processing for VLSI, FPGAs, GPUs, HPC and Vision Algorithms.



**Alberto A. Del Barrio** received the Ph.D. degree in Computer Science from the Complutense University of Madrid (UCM), Madrid, Spain, in 2011. Since 2017, he has been an Interim Associate Professor of computer science with the Department of Computer Architecture and System Engineering, UCM. His research interests include Design Automation, Arithmetic as well as Video Coding Optimizations.



**Carlos García** received his MS in Physics and PhD in Computer Science from the Universidad Complutense de Madrid (UCM) in 1999 and 2007, respectively. His research interests include parallel computing, computer architecture and code optimization for high performance computing. His current research also addresses heterogeneous systems and energy-aware trade-off.

**Manuel Prieto-Matías** received the Ph.D. degree from the Complutense University of Madrid (UCM) in 2000. He is now Associate professor in the Department of Computer Architecture at UCM and serves as vice-dean for External Relations and Research at the School of Computer Science and Engineering. His research interests include areas of parallel computing and computer architecture. Most of his activities have focused on leveraging parallel computing platforms and on complexity-effective micro-architecture design. His current research addresses emerging issues related to asymmetric processors, heterogeneous systems and energy-aware computing, with a special emphasis on the interaction between the system software and the underlying architecture. He has co-written numerous articles in journals and for international conferences in the field of parallel computing and computer architecture.



**Christos Grecos** (MIEEE'01-SMIEEE'06, SMSPIE'09) is a professor in visual communications standards and Head of School of Computing, University of the West of Scotland (UWS), UK. He leads the Centre of Audio-Visual Communications and Networks (AVCN) with UWS, and his main areas of expertise are image and video compression algorithms and standards. He has more than around 140 publications and has obtained significant funding for his research from the Engineering and Physical Sciences Research Councils (UK), Technology Strategy Board (UK), European Regional Development Fund and Scottish Funding Council. He is in the international and technical program committees of many prestigious conferences. Before coming to Scotland he has worked at the universities of Central Lancashire and Loughborough. He received his PhD degree in image/video coding algorithms from the University of Glamorgan, UK.

## Affiliations

**D. G. Fernández<sup>1</sup> · Guillermo Botella<sup>1</sup> · Alberto A. Del Barrio<sup>1</sup> · Carlos García<sup>1</sup> · Manuel Prieto-Matías<sup>1</sup> · Christos Grecos<sup>2</sup>**

D. G. Fernández  
davidgfe@ucm.es

Guillermo Botella  
gbotella@ucm.es

Carlos García  
garsanca@ucm.es

Manuel Prieto-Matías  
mpmatias@ucm.es

Christos Grecos  
christos.graikos@cwu.edu

<sup>1</sup> Department of Computer Architecture and Automation, Complutense University of Madrid, Ciudad Universitaria s/n, 28040 Madrid, Spain

<sup>2</sup> Department of Computer Science, Central Washington University, Washington, WA 98926-7520, USA