

Research Article

Low-Complexity Hierarchical Mode Decision Algorithms Targeting VLSI Architecture Design for the H.264/AVC Video Encoder

Guilherme Corrêa,¹ Daniel Palomino,² Cláudio Diniz,¹
Sergio Bampi,¹ and Luciano Agostini²

¹Microelectronics Group (GME), Federal University of Rio Grande do Sul (UFRGS), Av. Bento Gonçalves, 9500, P.O. Box 15064, 91501-970 Porto Alegre, RS, Brazil

²Group of Architectures and Integrated Circuits (GACI), Federal University of Pelotas (UFPEL), Campus Universitário s/n, P.O. Box 354, 96010-900 Pelotas, RS, Brazil

Correspondence should be addressed to Guilherme Corrêa, guilherme.correa@co.it.pt

Received 18 December 2011; Accepted 1 March 2012

Academic Editor: Maurizio Martina

Copyright © 2012 Guilherme Corrêa et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In H.264/AVC, the encoding process can occur according to one of the 13 intraframe coding modes or according to one of the 8 available interframes block sizes, besides the *SKIP* mode. In the Joint Model reference software, the choice of the best mode is performed through exhaustive executions of the entire encoding process, which significantly increases the encoder's computational complexity and sometimes even forbids its use in real-time applications. Considering this context, this work proposes a set of heuristic algorithms targeting hardware architectures that lead to earlier selection of one encoding mode. The amount of repetitions of the encoding process is reduced by 47 times, at the cost of a relatively small cost in compression performance. When compared to other works, the fast hierarchical mode decision results are expressively more satisfactory in terms of computational complexity reduction, quality, and bit rate. The low-complexity mode decision architecture proposed is thus a very good option for real-time coding of high-resolution videos. The solution is especially interesting for embedded and mobile applications with support to multimedia systems, since it yields good compression rates and image quality with a very high reduction in the encoder complexity.

1. Introduction

The latest technological advances in multimedia systems have brought us a wide range of multimedia-capable devices, such as mobile phones, personal digital assistants (PDAs), portable computers, and digital televisions. Recently, high-resolution video applications are much more common even in battery-operated portable devices in order to provide more and more visual quality for consumers. In addition, video streaming and digital television applications introduce the requirement of real-time processing. This scenario implies in a very high amount of data to be processed, stored, or transmitted.

Video coding standards, such as H.264/AVC [1], include a series of coding tools to achieve high compression rates

while preserving the video visual quality. The prediction steps of H.264/AVC (composed by intra-frame prediction and inter-frames prediction) are responsible for the main contribution of compression provided by this standard, which results, for example, in 50% fewer bits to represent a video when compared to MPEG-2 [2]. This result was achieved through the insertion of a large number of coding modes in the prediction steps, which must be evaluated and compared to each other (for this reason they are also called candidate modes) so that the best one is selected to encode each macroblock (MB).

Rate-distortion optimization (RDO) [3, 4] is a well-known technique used in video encoding to achieve the best coding efficiency by measuring and comparing the distortion and the bit rate of an encoded MB for all

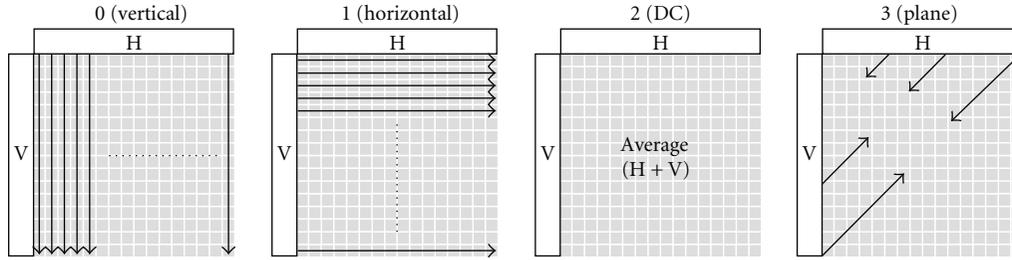


FIGURE 1: I16 MB prediction modes.

available prediction modes. However, the computation of a large number of prediction modes provided by H.264/AVC standard is extremely expensive in terms of computational complexity. For example, considering an HD 1080 p (1920×1080 pixels) video sequence with a group of pictures (GOP) composed by one intraframe followed by five interframes (IPPPPP), more than 8 million iterations composed by prediction, transform, quantization, inverse transform, inverse quantization, and entropy coding (called in this work as *encoding loop*) are needed for each video frame (148 iterations for each intramacroblock and 168 iterations for each intermacroblock). This large number of iterations leads to high energy consumption levels, which is critical in portable battery-operated multimedia devices. Besides, performance is also seriously affected, sometimes even forbidding the encoder to manipulate high-resolution video sequences in real time.

Recently, several works have proposed fast mode decision algorithms and VLSI designs for the H.264/AVC encoding process [5–12]. Even though all these works apply some heuristics at a certain point of the mode decision process, all of them use the RDO technique in some stage to select prediction modes and block sizes, limiting the reduction in terms of computational complexity and energy consumption of a fully RDO-based decision. The method presented in this paper is different. We propose a set of heuristics based on raw video data (which is available in the beginning of encoding process) to select the intra- and interframe coding modes *before* the encoding process itself, completely eliminating the use of RDO. Our technique was developed targeting a VLSI implementation into an H.264/AVC video encoder. The approach yields a computational complexity reduction of 47 times (in terms of encoding loop iterations) when compared to the original RDO-based solution. An average increase of 6.84% in bit rate and an average decrease of 0.25 dB in peak signal-to-noise ratio (PSNR) are noticed in worst-case decisions, but these drawbacks are still small when compared to the significant complexity reduction which satisfies real-time requirements of recent high-resolution video applications running into portable devices. The implemented VLSI design for intra- and interframe mode decisions allowed quantifying such gains provided by the heuristics.

The rest of this paper is organized as follows. Section 2 shows an overview of H.264/AVC coding modes and RDO-based mode decision. Section 3 presents some previous, related works found in the literature. Section 4 exposes the proposed method for fast mode decision, and Section 5 pre-

sents the VLSI design of each mode decision module. Section 6 presents rate-distortion performance results, a computational complexity analysis for the proposed method and the architectural synthesis results. Section 7 presents comparisons with related works, and Section 8 concludes this work indicating some future research directions.

2. Mode Decision in H.264/AVC

Intra-frame prediction explores the spatial redundancy in the current frame based on spatial neighbor samples, whilst inter-frame prediction, formed by motion estimation (ME) and motion compensation (MC), explores the temporal redundancy between frames.

Two intra-frame prediction types are defined for luminance (Y) samples: intra frame 16×16 (I16 MB) and intra frame 4×4 (I4 MB). For I16 MB, one mode among four prediction modes (shown in Figure 1) is chosen. Each mode generates a predicted MB using the 32 neighbor samples at the top and left boundaries of the original MB. For I4 MB, the MB is broken into sixteen 4×4 blocks that are individually predicted through nine prediction modes using up to 13 neighbor samples at the boundary of each 4×4 block (shown in Figure 2). For chrominance samples, the prediction is always applied over 8×8 blocks and uses the same I16MB modes, although it uses less neighbor samples and another order to identify the modes. The same prediction is applied for both blue and red chrominance channels [13].

In inter-frames prediction, the ME module searches in the reference frames the block which is the most similar to the current block. When the best match occurs, a motion vector (MV) is generated. One of the new features of H.264/AVC is the variable block-size motion estimation (VBSME), where one MB can be divided according to four partition types: $P16 \times 16$, $P16 \times 8$, $P8 \times 16$, and $P8 \times 8$, as shown in Figure 3. Each $P8 \times 8$ partition (called sub-macroblock) could be further subpartitioned according to four subpartition types: $SMB8 \times 8$, $SMB8 \times 4$, $SMB4 \times 8$, or $SMB4 \times 4$, as shown in Figure 4. Each partition or subpartition is predicted separately and has its own MV to point to a different block of the reference frame [13].

Besides these partition and subpartition sizes, the interframes prediction can decide to use the *SKIP* mode, which is a special block which does not send any information to the decoder. This case happens when the rate-distortion cost of sending MB information is higher than the cost of not sending any information regarding this MB [13].

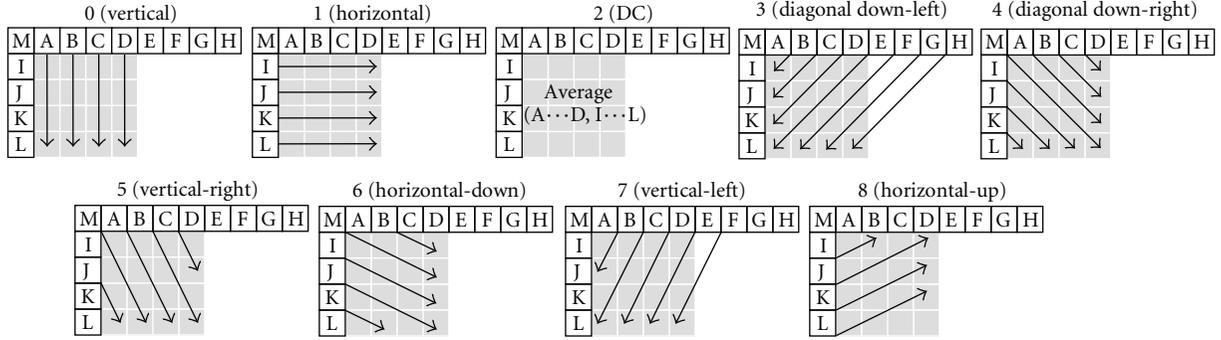


FIGURE 2: 14 MB prediction modes.

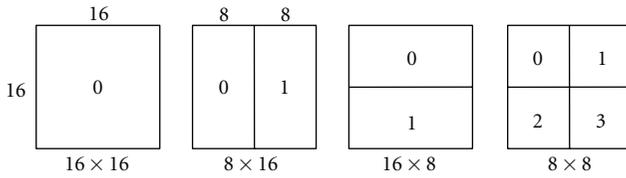


FIGURE 3: Partition types of a macroblock.

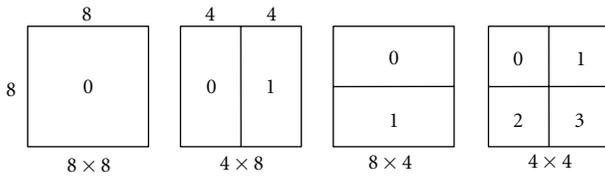


FIGURE 4: Subpartitions types of a P8 x 8 sub-macroblock.

As previously mentioned, the rate-distortion optimization (RDO) technique is suggested by the H.264/AVC standard to select, among all the candidate modes, which is the most efficient one in terms of compression and image quality. Equation (1) shows the Lagrangian rate-distortion cost formula, in which D denotes the distortion (measured in PSNR), R denotes the bit rate, λ denotes the Lagrangian multiplier, and J is the final rate-distortion cost

$$J = D + \lambda \cdot R. \quad (1)$$

Equation (1) is applied to each prediction mode for all MB inside a frame. Figure 5 shows the sequential execution flow of the H.264/AVC encoder in the JM reference software [14]. The first part, represented by intra-/inter-prediction, performs the intra- or interframe prediction considering one possible encoding mode. In the case of intra-frame prediction, this stage determines the predicted samples for one of all I16 MB and I4 MB modes presented in Figure 1 and Figure 2 and also for the chrominance blocks. In the case of inter-frame prediction, the most similar block is searched in the reference frames considering one of all possible partition and sub-partition sizes (P16 x 16, P16 x 8, P8 x 16, P8 x 8, SMB8 x 8, SMB8 x 4, SMB4 x 8, SMB4 x 4). After the intra-/interprediction, residual generation, transforms (T),

quantization (Q), and entropy coding are applied, the bit rate for the prediction mode is calculated. The distortion is evaluated after the inverse transform (IT), the inverse quantization (IQ), and the addition of the residue to the reconstructed prediction samples for each mode. All grey blocks in Figure 5 are executed once for each possible intra- and intermodes, whilst the mode decision block (in white) receives all prediction modes' bit rates and distortions (dashed lines) generated in each iteration of the encoding loop, calculates the rate-distortion cost according to (1), and finally selects the mode with the lowest cost to be used in the encoding process.

It is possible to perceive that the RDO technique is extremely complex, making its use in encoders for real-time systems and high-resolution videos difficult, especially when portable devices are considered (since energy consumption is directly proportional to computational complexity). The use of fast algorithms and hardware architectures is thus of essential importance to reduce the mode decision complexity, increasing the encoder performance with negligible PSNR drop and small bit rate increase.

3. Previous Works

Recently, several works have proposed fast mode decision algorithms for the H.264/AVC encoding process. All of them aim at the reduction of the decision process complexity by decreasing the number of possible encoding modes considered in the RDO process [5–12].

In [5, 7, 9, 11] some heuristics are proposed to discard intraprediction modes for RDO calculation based on the transformed coefficients. The method proposed in [9] uses the low-frequency transform coefficients to detect image homogeneity, selects the best intrablock size, and then performs the selection of intramodes based on the RDO technique. In [11], the authors propose a SATD-based intra-frame decision in which the modes with high sum of absolute transformed differences (SATDs) values are not considered in the RDO process. In [5] the transformed coefficients of each block are analyzed in order to determine which modes would be evaluated by the RDO process. The work claims that each transform block tends to assume a formation pattern according to the encoding mode applied in the intra-prediction process. In [7], the SATD value is also used

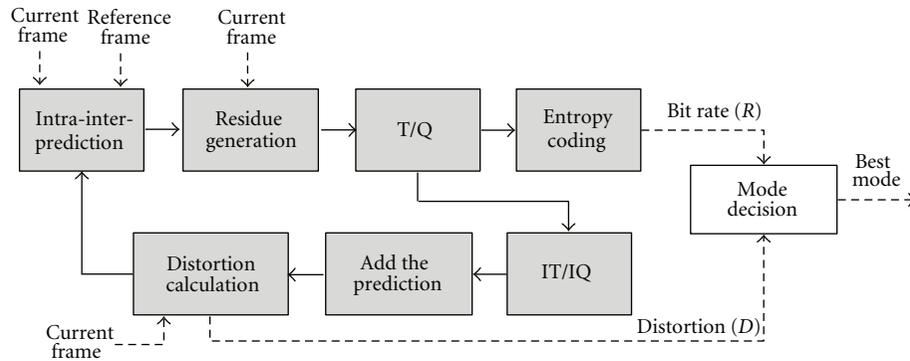


FIGURE 5: Diagram of the RDO-based encoding process.

to discard the less probable modes, as in [11]. However, the algorithm applies a bit rate estimation technique based on the number of coefficients quantized to zero after the quantization step. In [9], the low-frequency components of the transformed blocks are analyzed in order to identify the homogeneity of the original image, as proposed in this work. However, [9] is limited to decide only the best block size, so that the best 4×4 and the best 16×16 intraframe modes are still chosen through RDO iterations. As in [7, 11], the work [9] also eliminates probable modes by the analysis of SATD values.

The mode decision proposed in [6] chooses the best intra-frame mode based on the analysis of the directional gradient of each block. The two modes with the lowest gradient levels are selected and the RDO technique is applied to both of them and the DC mode. In [10], an intra-frame fast mode decision is performed based on the analysis of the motion field continuity. The Sobel operator is used in that work in order to detect borders and predict the spatial continuity of the objects.

Other works proposed heuristics for inter-frame mode decision. In [12], a set of heuristics for intra-frame and inter-frames mode decisions are proposed. The method is composed by stationarity detection through sum of absolute differences (SAD) calculation and border detection using the Sobel operator. When a specified condition is achieved, the RDO process is finalized and the encoding process occurs according to the best RDcost found so far. If no condition is achieved, all possible modes are still checked. The Sobel operator is also used in [10] to select larger or smaller partition blocks for inter-frames prediction. In [8], a fast mode decision algorithm based on the occurrence frequency of each prediction mode in P frames is proposed. The algorithm evaluates the performance of the most frequent modes before the less frequent modes. The method also applies a conditional evaluation of intra-frame modes, which happens only if the spatial correlation level is higher than the temporal correlation level.

Architectural designs based on heuristics have been also proposed in some works aiming at decreasing the encoding time for one MB [15–17]. In [15], a hardware design which applies a method similar to that in [6, 10] is proposed for the intramode decision. A filtering technique is used

to perform the edge detection so that the number of modes is reduced from nine to four in I4 MB blocks. For I16 MB and chrominance block decisions, only the detected mode and the DC mode are selected to be evaluated by the RDO technique. In [16], a low-complexity intra mode decision algorithm based on a cost function composed by distortion calculation and bit rate estimation is implemented in a hardware architecture. In [17], a modified three-step algorithm performs the intra mode decision decreasing the number of I4 MB candidate modes (from nine to seven) to be evaluated by the RDO technique. Regarding the inter mode decision, to the best of the authors knowledge there are no published works yet which propose a hardware architecture for the inter mode decision independently from the implementation of a motion estimation module.

Even though all these works propose the use of some heuristics, they apply the RDO method in some stage to select prediction modes and block sizes and do not allow the achievement of significant reductions in terms of computational complexity compared with the fully RDO-based decision.

4. Fast Hierarchical Mode Decision

In this work, a fast hierarchical mode decision for the H.264/AVC standard targeting high-resolution video encoding is proposed. The goal of the method is to perform a heuristic decision which uses only the original frame and the reference frames information instead of performing the complete RDO process. The heuristic solutions and their respective architectures were all developed for low-complexity video encoding systems and are thus based on the baseline profile of H.264/AVC. They can be, however, easily incorporated into other profiles.

The hierarchical mode decision was divided into two steps: intra-prediction decision and inter-prediction decision. To reduce the computational complexity, we have considered that P slices support only P blocks (I blocks are not supported, as explained in the next subsections), so that the two proposed decision modules are independent from one another. This way, the first module is used only for I slices (generated only by intra prediction) whilst the second one is used only for P slices (generated only by inter prediction).

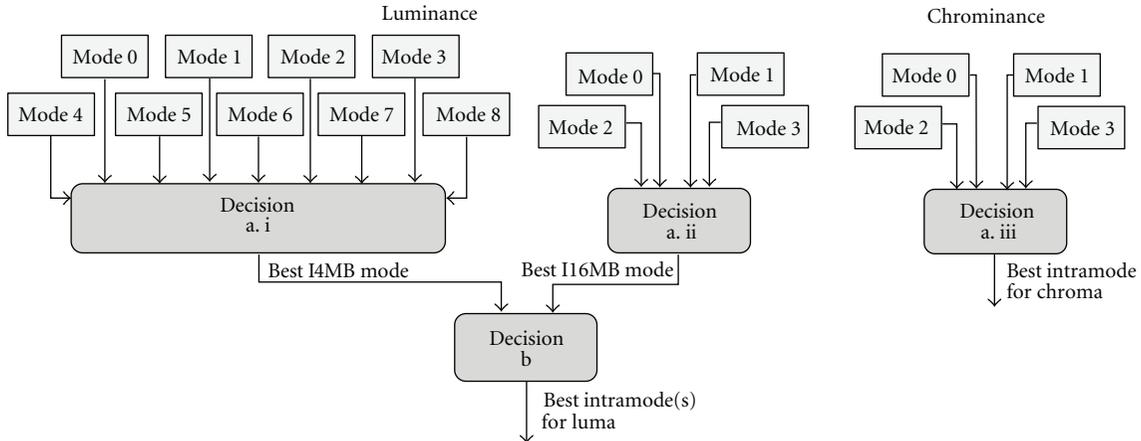


FIGURE 6: Hierarchical intra mode decision flow.

Even though B (bipredictive) slices are not supported in our work, the techniques proposed for P slices could be easily extended for B slices.

4.1. Fast Intramode Decision. The intra-frame mode decision was divided in a hierarchical way aiming at computational complexity decrease and modularization of the whole process. The steps which comprise intra-frame decision are as follows:

- (a) decision for equal block size:
 - (i) decision of the best I4 MB mode for each 4×4 luminance block,
 - (ii) decision of the best I16 MB mode for the luminance channel of the macroblock,
 - (iii) decision of the best 8×8 mode for the chrominance channels of the macroblock,
- (b) block-size decision for the luminance MB (I4 MB or I16 MB).

Each decision for equal block size (step (a)) is independent from one another and can be performed in parallel. The luminance block-size decision (step (b)) is performed after all decisions in step (a) are complete. Figure 6 presents the flow of decisions for the intra-frame prediction. As stressed by the orientation of the arrows in the flow, the decision process starts at the bottom of the hierarchy trees, which means that the best luminance I4 MB, I16 MB and chrominance 8×8 modes are decided before the final decision of the best luminance block size.

4.1.1. Intra Mode Decision for Equal Block Sizes. A distortion-based method was used to define which are the best modes for 4×4 luma block (I4 MB), the best mode for 16×16 luma block (I16 MB) and the best mode for 8×8 chroma block. These three steps were presented as *Decision a.i*, *Decision a.ii*, and *Decision a.iii* in the flow presented in Figure 6. The method consists simply in comparing the block prediction

TABLE 1: Comparison among SAD, SSD, and SATD.

SATD versus SAD			SSD versus SAD		
PSNR (dB)	Bit rate (%)	Number of Sums (%)	PSNR (dB)	Bit rate (%)	Number of Sums (%)
+0.008	-0.63	+361.29	+0.079	-1.13	+348.39

error of all candidate modes and choosing the mode with the lowest value as the best one.

The block prediction error may be computed through the use of several distortion metrics. The use of sum of absolute differences (SADs), sum of absolute transformed differences (SATD) and sum of squared differences (SSDs) was evaluated [13]. Table 1 presents a comparison between the uses of the three distortion metrics in the mode decision module. All results are presented as relative increase or decrease in comparison to the use of SAD.

Even though SATD and SSD present slightly better rate-distortion performance results (around 1% bit rate decrease), the computational complexity of these metrics is extremely larger than the computational complexity of SAD (an increase around 350% for both metrics). As the main goal of this work is to design a faster and low-complexity intra mode decision heuristic, the SAD metric was used in the method proposed in this work. Besides that, a hardware architecture design for SAD calculators is much simpler than a hardware architecture for SATD (which includes a 4×4 Hadamard transform) and for SSD (which includes a multiplier and a square root).

4.1.2. Intramode Decision for Different Block Sizes. As previously explained, the second step of the intra mode decision process, shown as *Decision b* in Figure 6, consists in deciding the best macroblock size (I4 MB or I16 MB). In the proposed method, the choice between the use of 16 4×4 predicted blocks (I4 MB) or just one 16×16 predicted block (I16 MB) is made using the SAD results calculated in the first step of the mode decision.

The proposed algorithm picks the total distortion of the $16 \times 4 \times 4$ blocks (I4 MB), which had all their best modes individually decided by *Decision a.i* (Figure 6) and compares it with the distortion of the unique 16×16 block (I16 MB), which had its best mode decided by *Decision a.ii*. In most cases, I4 MB would present the smallest distortion value, since a finer prediction granularity was used in it and more encoding modes are allowed for 4×4 blocks than for 16×16 blocks (Figures 1 and 2). This way, a simple SAD comparison cannot be used between them. On the other hand, when the differences between these two distortion values in RDO-based mode decision are analyzed, a good correlation between it and the best block size is perceived.

A set of simulations were performed with seven video sequences typically used by the video coding community (*Station 2, Sunflower, Tractor, Man in Car, River Bed, Rolling Tomatoes, Rush Hour*) using the *Joint Model (JM) H.264/AVC* reference software set in full-RDO-based decision and intra-only MB modes [18]. It was possible to notice that in most cases when I16 MB had been chosen, the distortion values of I4 MB and I16 MB were very close. This happens because in such cases most of the $16 \times 4 \times 4$ blocks inside an I4 MB presented the same prediction modes, and thus only one 16×16 block for the whole MB can be used (i.e., I16 MB is chosen), since it generates less encoding information. On the other hand, when an I4 MB is chosen, the distortion values of I4 MB and I16 MB were very different. This happens because most of the $16 \times 4 \times 4$ blocks inside a I4 MB presented different modes, and then choosing only one 16×16 mode would generate a lot of residual information.

To better evaluate the relation presented in the previous paragraph, (2) was defined. This equation shows the difference of distortion (DD) calculation, where SAD_{I4MB} represents the sum of all residual generated by the 16 best 4×4 modes and SAD_{I16MB} is the residual generated by the best 16×16 mode

$$DD = SAD_{I4MB} - SAD_{I16MB}. \quad (2)$$

Figure 7 shows a graph in which the occurrence frequency of each block size chosen by the RDO-based mode decision is plotted with the corresponding DD calculated as in (2). It is possible to perceive that in most cases when I16 MB is selected (grey line) the DD value is very small. Around 97% of the macroblocks encoded as I16 MB presented a DD smaller than 600. On the other hand, when I4 MB is chosen by the RDO-based mode decision, the DD value is generally very large. Around 84% of the macroblocks presented a DD value larger than 600. This way, a threshold value for the DD value can be used to choose the best block-size for intra prediction with a small error regarding the decision.

The best threshold value was experimentally chosen aiming at the best results in terms of PSNR and bit rate, as follows. Firstly, all video sequences used in the simulations were encoded using the RDO-based mode decision for different block sizes. Meanwhile, distortion values and chosen block sizes were saved for future analysis. Then, the

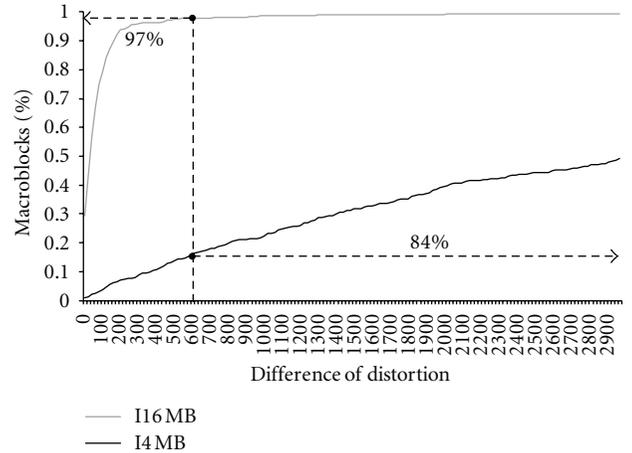


FIGURE 7: Occurrence of I4 MB and I16 MB and corresponding difference of distortion (DD) values.

DD values were calculated and compared with the chosen block size, as in Figure 7. Simulations were performed with a threshold ranging from 0 to 1000, adjusting it to improve bit rate and video quality. The threshold value set to 600 generated the best results in terms of bit rate and video quality for the evaluated sequences.

4.2. Fast Inter Mode Decision. As in the intra-frame mode decision process, the inter-frames decision was hierarchically divided, aiming at the simplification of the problem in simpler and independent steps. As previously mentioned, the first decision to reduce the inter prediction decision complexity was the elimination of I blocks support into P slices. This way, only P blocks generated through the inter-frames prediction are allowed in such slices. The impacts of this simplification were measured through tests using the reference software and the previously cited video sequences. A reduction of only 0.01 dB in video quality and an increase of only 0.39% in bit rate were noticed when P slices were restricted to inter predicted blocks.

Previous analyses of the occurrence frequency for each mode in P slices were performed in order to identify which mode should be evaluated first. The same eight video sequences related above were used in these analyses. The obtained results show that *SKIP* and P 16×16 macroblocks represent almost 60% of the whole set of used modes in 1080 p (1920×1080 pixels) video sequences, while the use of sub-macroblock partitions smaller than 8×8 represents about 6% of occurrence, as shown in Table 2. Other important result of Table 2 is that intra modes are rarely used (3.8%) and, as cited before, the elimination of these modes in P slices causes a very small reduction in rate-distortion performance. Similar tests were performed for 720 p (1280×720 pixels) video sequences, which presented a small increase in the occurrence of small block sizes (4×4 , 4×8 and 8×4). As this work is focused on high-resolution video sequences, only results for 1080p sequences are presented in Table 2.

It is possible to conclude from Table 2 that the partition/subpartition occurrence is higher when the block size

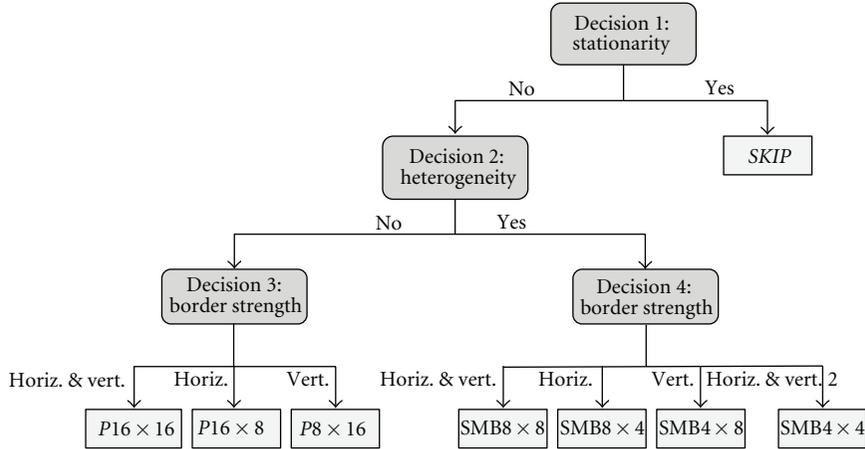


FIGURE 8: Hierarchical mode decision flow for inter-prediction decisions.

TABLE 2: Average occurrence and frequency for each mode in P slices.

Mode	Occurrence	Frequency
<i>SKIP</i>	83,645	29.1%
$P16 \times 16$	83,651	29.1%
$P16 \times 8$	30,364	10.6%
$P8 \times 16$	33,187	11.5%
$SMB8 \times 8$	27,251	9.5%
$SMB8 \times 4$	7,700	2.7%
$SMB4 \times 8$	8,472	2.9%
$SMB4 \times 4$	2,331	0.8%
<i>Intramodes</i>	10,895	3.8%

is larger especially for large video resolutions. These results confirm the high-definition digital video characteristics of containing a large amount of homogeneous and stationary areas. Based on this occurrence order, a sequence of steps was developed for fast inter mode decision. This sequence prioritizes the identification of the most frequent modes before the least frequent, thus decreasing the average number of performed decision steps.

Figure 8 shows the inter-frames hierarchical mode decision flow proposed in this work. The first performed inter mode decision step is at the top of the tree, as indicated by the arrows direction. In the first step, an analysis of the original and the reference macroblocks is performed in order to allow stationarity detection between two neighbor frames. The second step, based on heterogeneity detection, decides whether or not subpartitioning is necessary. The third and the fourth decisions are quite similar and are based on gradient calculation to detect object borders. The next subsections describe each step in details.

4.2.1. Stationarity-Based SKIP Mode Decision. Natural video sequences present the characteristic of being composed by constant motion and by temporally stationary regions. This characteristic is highly noticeable especially in background

images [12]. In the JM reference software, when the RDO-based decision is used, the *SKIP* mode is selected when the cost of not encoding residual information is smaller than the costs of all the other possible modes. Otherwise, a set of four conditions must be satisfied for a macroblock to be encoded with the *SKIP* mode. In [19], the authors affirm through experimental results that the satisfaction of one of them is already enough for a good indication of *SKIP* mode usage. Such condition dictates that the transform coefficients of the *SKIP* mode residue block are all quantized to zero.

Stationarity, also called stillness, is a simple and efficient way of identifying the similarity level of each frame and its neighbor frame in a video. In this work, the simple distortion calculation between a macroblock and its collocated macroblock is proposed to detect stationarity. If the SAD between the two macroblocks is smaller than a pre-defined threshold, the image region is considered temporally stationary, and this region is thus encoded as *SKIP*.

A series of tests were performed in order to verify the threshold SAD value between neighbor frames when *SKIP* and non-*SKIP* macroblocks were selected by the RDO-based decision. Average values obtained for both cases were used in tests which varied the SAD threshold for *SKIP* decision from 200 to 700. The best results in terms of rate-distortion performance were obtained when the value 500 was used. This means that the proposed *SKIP* decision consists in computing the SAD between co-located macroblocks in the reference and the current frame and then comparing this value with the threshold (500). If the SAD is smaller than it, the macroblock is considered stationary and it is encoded as *SKIP*. Otherwise, the macroblock is not considered stationary and it is encoded with another mode.

As the heuristic is a suboptimal solution of the RDO decision, it may select more or less *SKIP* macroblocks, depending on the video sequence. The average results have shown that, in general, more *SKIP* macroblocks were selected than in the RDO process. This caused an increase in the bit rate and also in the image quality when this heuristic was inserted in the reference software. An average increase of 0.04 dB was noticed in the PSNR when this heuristic

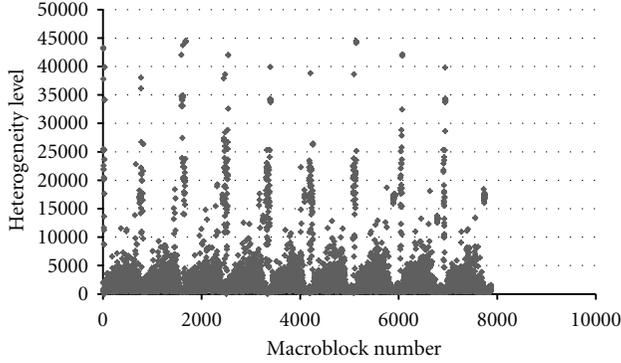


FIGURE 9: Heterogeneity level (H) for each macroblock encoded without sub-partitions in the sequence *Man in Car* (RDO-based decision).

was applied. On the other hand, the bit rate presented an increase of 1.75% when compared to the RDO-based decision.

4.2.2. Heterogeneity-Based SubPartition Detection. Natural video sequences are also composed by a large amount of homogeneous regions. According to [20], these regions generally present similar directional motion, while heterogeneous regions present disordered movements. This happens because homogeneous areas tend to belong to stationary regions of an image or tend to belong to regions of the same object of an image, which is composed by parts that move together and continuously in the same direction. This way, homogeneous regions are, in general, encoded using large block sizes, such as 16×16 , 16×8 , and 8×16 , while heterogeneous regions are encoded using smaller block sizes.

We have shown in a previous work that the heterogeneity of a macroblock can be detected through the analysis of some coefficients generated after the application of a forward transform, such as the Hadamard and the DCT [21, 22]. The heterogeneity H of a macroblock is thus calculated from a set of low-frequency coefficients from the transformed block. In [21], we propose to use only the first line and the first column of the transformed coefficients in the heterogeneity calculation, as shown in (3), where Y is the transformed macroblock and u and v are the macroblock lines and columns, respectively. The higher is the value of H , the higher is the heterogeneity of the macroblock

$$H = \sum_{u=1}^{15} |Y(u, 0)| + \sum_{v=1}^{15} |Y(0, v)|. \quad (3)$$

A set of experimental tests led to a threshold value which can be used to determine if subpartitions are used or not. If the heterogeneity of the macroblock is high (i.e., the H value is higher than the threshold), sub-partitions should be used. Otherwise, only large partitions should be used. Figures 9 and 10 show graphs in which each point represents a macroblock of the *Man in Car* video sequence. In Figure 9, each point is a macroblock encoded as $P16 \times 16$, $P16 \times 8$ or $P8 \times 16$ (i.e., without sub-partitions) and in

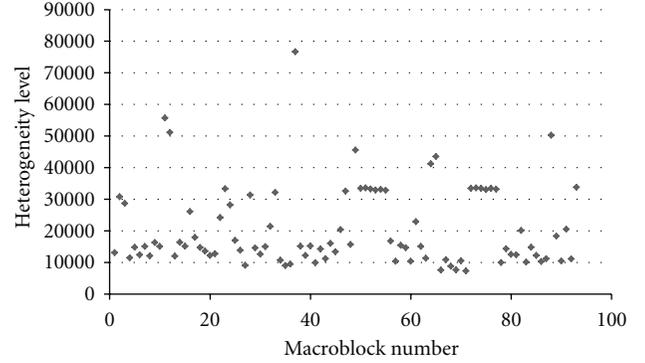


FIGURE 10: Heterogeneity level (H) for each macroblock encoded with sub-partitions in the sequence *Man in Car* (RDO-based decision).

Figure 10 each point is a macroblock encoded with sub-partitions. The decisions were all performed by the RDO process. It is possible to notice that most of the macroblocks encoded without sub-partitions present an H value smaller than 10,000, while most of the macroblocks encoded with sub-partitions present an H value larger than 10,000. As this behavior is presented in most of the analyzed videos, the value 10,000 was defined to be used in this heuristic as the threshold between the use of large partitions and the use of sub-partitions. Experimental results have shown that an increase of only 0.62% in the bit rate and a negligible image quality loss (around 0.01 dB) were perceived when this threshold was used in such decision step.

4.2.3. Border Detection for Partition Decision. The second decision step presented in the last sub-section decides whether the third or the fourth decision step takes place. The third step decides which partition size is used, and it is based on the difference between pixels that compose the lines and the columns of a macroblock, which is called the gradient. As previously explained, in natural video sequences all the parts of an object tend to move together in the same direction and they are probably grouped inside the same partition. This way, the detection of object edges around possible partition edges allows the definition of which is the best partition size for the macroblock. In the proposed decision method, different objects are considered to have different pixel values, whilst different regions of the same object present a higher probability of being composed by similar pixel values. Based on this, different objects can be distinguished with simple comparisons between neighboring pixels. Even though this is a generalization which incurs in some decision error, bit rate and image quality are not so strongly affected, as shown below.

The proposed edge detection heuristic is based on simple subtractions between neighboring pixels located around possible partition borders. In Figure 11, vertical and horizontal partition borders are shown as bold lines and the pixels used in the border calculation are shown in gray. Eight pixels are used from each line (or column) in the gradient calculation. Equation (4) Defines the summation performed

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0					p_3	p_2	p_1	p_0	q_0	q_1	q_2	q_3				
1																
2																
3																
4	p_3															
5	p_2															
6	p_1															
7	p_0															
8	q_0															
9	q_1															
10	q_2															
11	q_3															
12																
13																
14																
15																

FIGURE 11: Pixels used in the gradient calculation for partition size decision.

to determine the vertical border strength (VB) and (5) defines the horizontal border strength calculation (HB), where O is the pixel matrix from the original macroblock, i is the macroblock line index, and j is the macroblock column index. As shown in the equations, pixels are compared two by two (i.e., p_3 is compared with q_3 , p_2 is compared with q_2 , and so on), which allows that both abrupt and gradual pixel changes are found

$$VB = \sum_{i=0}^{15} \sum_{j=4}^7 (O_{i,j} - O_{i,(15-j)}), \quad (4)$$

$$HB = \sum_{j=0}^{15} \sum_{i=4}^7 (O_{i,j} - O_{(15-i),j}). \quad (5)$$

After computing the horizontal border (HB) and the vertical border (VB) values, a set of comparisons is performed to determine the best macroblock partition. A set of tests has shown that a threshold value set in 80 for the difference between horizontal and vertical borders leads to the best results in terms of rate-distortion performance. This way, Algorithm 1 was used to decide the best partition size.

Experimental results have shown that a bit rate increase of 3.59% and a PSNR increase of 0.02 dB were noticed when this method was used instead of the RDO-based decision. As previously explained, this is a sub-optimal solution of the RDO method and therefore it eventually performs different decisions, which leads to a different configuration of partitions (and thus a larger or smaller bit rate and PSNR difference) depending on the video sequence.

4.2.4. Border Detection for Subpartition Decision. The fourth decision step is similar to the third one, though it analyzes more than two borders. Figure 12 presents the structure of a macroblock divided in four smaller structures, the 8×8 sub-macroblocks. For each sub-macroblock, the vertical and horizontal border calculations are performed and its results are compared in order to decide the sub-partition type. Differently from the border calculation for partitions, sub-partition decision calculations are performed over four samples instead of eight. Besides, in order to allow identifying 4×4 partitions, each border is calculated in two parts: the top (or left) subborder and the bottom (or right) sub-border.

Equations (6)–(11) present the calculations for vertical and horizontal sub-partition borders of the first sub-macroblock. The vertical and the horizontal sub-partition borders are calculated as in (6) and (7), where VPB is the vertical sub-partition border, HPB is the horizontal sub-partition border, and VSB1, VSB2, HSB1, and HSB2 are the halves of each border (called sub-borders). Equations (8), (9), (10), and Equation (11) present the calculation of each vertical and horizontal sub-border, respectively, where O is the pixel matrix from the original macroblock, i is the macroblock line index, j is the macroblock column index, VSB1 and VSB2 are the first and second halves of the vertical sub-border, respectively, and HSB1 and HSB2 are the first and second halves of the horizontal sub-border, respectively

$$VPB = VSB1 + VSB2, \quad (6)$$

$$HPB = HSB1 + HSB2, \quad (7)$$

$$VSB1 = \sum_{i=0}^3 \sum_{j=2}^3 (O_{i,j} - O_{i,(7-j)}), \quad (8)$$

$$VSB2 = \sum_{i=4}^7 \sum_{j=2}^3 (O_{i,j} - O_{i,(7-j)}), \quad (9)$$

$$HSB1 = \sum_{j=0}^3 \sum_{i=2}^3 (O_{i,j} - O_{(7-i),j}), \quad (10)$$

$$HSB2 = \sum_{j=4}^7 \sum_{i=2}^3 (O_{i,j} - O_{(7-i),j}). \quad (11)$$

Border values are compared to previously defined thresholds, according to Algorithm 2. As in all previously shown heuristics, the thresholds were obtained through exhaustive experimental tests which took into consideration the rate-distortion performance. The proposed sub-partition decision method led to an average bit rate increase of only 0.34% and a PSNR increase of 0.02 dB when the thresholds 40 and 20 were used for sub-partition border and sub-partition sub-border, respectively.

```

If |HB-VB| ≤ 80 Then
  best_INTER_mode = P16×16
Else If HB-VB > 80 Then
  best_INTER_mode = P16×8
  // (macroblock is divided into two 16×8 partitions)
Else If VB-HB > 80 Then
  best_INTER_mode = P8×16
  // (macroblock is divided into two 8×16 partitions)

```

ALGORITHM 1: Algorithm used to define the partition size.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0			p_1	p_0	q_0	q_1					p_1	p_0	q_0	q_1		
1																
2	p_1								p_1							
3	p_0								p_0							
4	q_0								q_0							
5	q_1								q_1							
6																
7																
8			p_1	p_0	q_0	q_1					p_1	p_0	q_0	q_1		
9																
10	p_1								p_1							
11	p_0								p_0							
12	q_0								q_0							
13	q_1								q_1							
14																
15																

FIGURE 12: Pixels used in the gradient calculation for sub-partition size decision.

```

If |HPB-VPB| ≤ 40 Then
  best_INTER_mode = SMB8×8
  // (sub-macroblock is not divided into sub-partitions)
Else If HPB-VPB > 40 Then
  If VSB1 > 20 OR VSB2 > 20 Then
    best_INTER_mode = SMB4×4
    // (sub-MB is divided into four 4×4 sub-partitions)
  Else
    best_INTER_mode = SMB8×4
    // (sub-MB is divided into two 8×4 sub-partitions)
  Else If VPB - HPB > 40 Then
    If HSB1 > 20 OR HSB2 > 20 Then
      best_INTER_mode = SMB4×4
      // (sub-MB is divided into four 4×4 sub-partitions)
    Else
      best_INTER_mode = SMB4×8
      // (sub-MB is divided into two 4×8 blocks)

```

ALGORITHM 2: Algorithm used to define the sub-partition size.

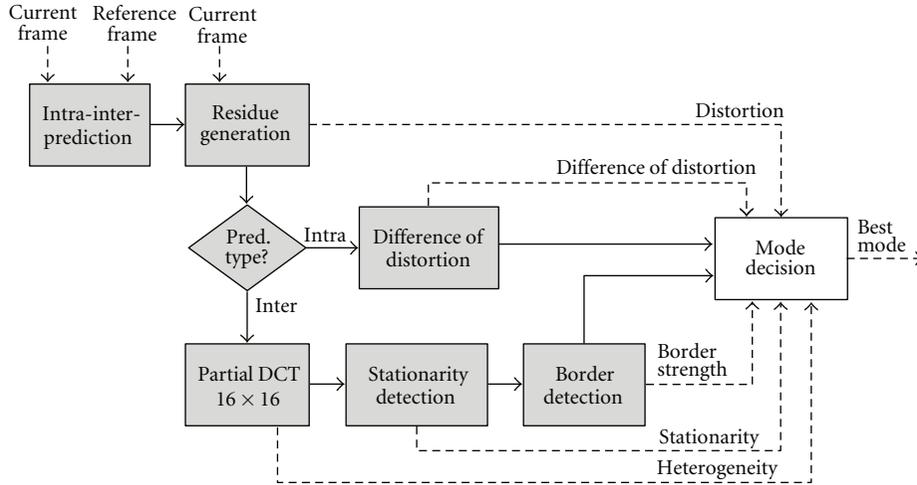


FIGURE 13: Block diagram of the proposed fast hierarchical mode decision for H.264/AVC.

4.3. Final Results of the Proposed Heuristics. The fast decision heuristics were evaluated together and compared with the RDO-based mode decision. This comparison considered the computational cost reduction and the PSNR and compression rates losses. The main results are presented in the next sub-sections.

4.3.1. Computational Complexity Results. As presented in Figure 5, the mode decision in the RDO-based encoding process is finished only after the execution of all possible intra- and inter-prediction candidate modes (i.e., after successive iterations of the encoding loop). Figure 13 presents a block diagram of our method in which all operations to perform mode decision are shown. The loop from Figure 5 is completely eliminated, significantly simplifying the decision process. Instead of performing the whole encoding process for all possible candidate modes, our method performs the mode decision right after the prediction residual generation and the difference of distortion calculation (when intra prediction is used) or after the prediction residual generation, the partial 16×16 DCT calculation, the stationarity detection, and the border detection (when inter prediction is used).

Considering only luminance mode decision, an analysis of the complexity reduction is done in the next paragraphs. When the RDO technique is applied for intra-frame mode decision, four 16×16 and nine 4×4 intra modes must be compared. This means that the decision of the best mode for a macroblock in an I slice occurs only after the evaluation of all the four prediction modes for the macroblock and all the nine prediction modes for 4×4 blocks (i.e., for each one of the 16 blocks which compose the macroblock). In terms of computational complexity, it means that the loop presented in Figure 5 is repeated four times for 16×16 intra-frame prediction and 144 times for 4×4 intra-frame prediction, totaling 148 iterations in the encoding flow.

The inter-frames mode decision in P slices is performed after the evaluation of the five partition modes (*SKIP*, $P16 \times 16$, $P16 \times 8$, $P8 \times 16$ and $P8 \times 8$) for each macroblock and the

four sub-partition modes for each one of the four 8×8 sub-macroblocks which compose a macroblock. In other words, 5 iterations of the loop presented in Figure 5 are executed for the macroblock modes and 16 iterations are executed for the sub-macroblock modes, totaling 21 iterations. Besides, as the H.264/AVC standard defines that P slices can also be composed by intra-frame macroblocks, all the 148 iterations for intra-frame modes are also performed, totaling 168 encoding repetitions per macroblock.

Considering the proposed fast hierarchical mode decision and its execution flow presented in Figure 13, the best mode decision of an I frame is performed right after the prediction residual generation and the difference of distortion calculation. It means that if the difference of distortion detects that the best macroblock partitioning is I16 MB, the encoding flow is executed only once considering the chosen 16×16 mode. Otherwise, the loop is executed 16 times, once for each 4×4 block in I4 MB. For inter-frames macroblocks in P slices, the best mode decision happens after the heterogeneity evaluation, the stationarity analysis, and the border detection calculations. The remaining encoding steps which are now outside the encoding loop (transform, quantization, inverse transform, inverse quantization, entropy coding, filtering) are then executed only once, considering the selected partition and the selected sub-partitions, if necessary.

Based on this complexity analysis and taking into consideration, a group of pictures (GOP) composed by one I frame and five P frames (IPPPPP) in an HD 1080p resolution video (8160 macroblocks per frame), the RDO-based encoding process would perform 8,062,080 iterations in the loop presented in Figure 5. On the other hand, the fast hierarchical mode decision performs 171,360 executions of the encoding flow to complete the encoding process considering the same GOP. This means a reduction of 47 times in the number of encoder iterations, which is a very high gain and justifies the small bit rate increase generated by these techniques, which is discussed in the next sub-section.

TABLE 3: Bit rate and PSNR for fast intra-frame mode decision and RDO-based mode decision.

Video Sequence	RDO decision		Fast intra frame mode decision		RDO \times fast intra frame mode decision	
	PSNR (dB)	Number of Bits (Kbits)	PSNR (dB)	Number of Bits (Kbits)	Δ PSNR (dB)	Δ bits (%)
<i>Station 2</i>	39.64	34,022	39.33	35,729	-0.30	+5.02
<i>Sunflower</i>	42.77	33,030	42.41	35,391	-0.36	+7.15
<i>Tractor</i>	39.40	58,208	39.04	60,939	-0.37	+4.69
<i>Man In car</i>	42.61	8,818	42.52	9,018	-0.09	+2.27
<i>River Bed</i>	38.65	56,735	38.28	58,962	-0.37	+3.93
<i>Rolling Tomatoes</i>	40.48	12,170	40.39	12,537	-0.09	+3.02
<i>Rush Hour</i>	41.65	20,013	41.48	21,306	-0.17	+6.46
Average	40.74	31,857	40.49	33,412	-0.25	+4.88

TABLE 4: Bit rate and PSNR for fast inter-frames mode decision and RDO-based mode decision.

Video sequence	RDO decision		Fast inter frame mode Decision		RDO \times Fast Inter frame mode Decision	
	PSNR (dB)	Number of Bits (Kbits)	PSNR (dB)	Number of Bits (Kbits)	Δ PSNR (dB)	Δ bits (%)
<i>Station 2</i>	39.64	4407	39.60	4630	-0.04	+5.06
<i>Sunflower</i>	42.58	7105	42.59	7636	+0.01	+7.47
<i>Tractor</i>	38.92	11344	38.87	11897	-0.05	+4.87
<i>Man in car</i>	42.43	2138	42.40	2171	-0.03	+1.54
<i>River bed</i>	40.36	3427	40.32	3740	-0.04	+9.13
<i>Rolling tomatoes</i>	41.47	3711	41.40	4066	-0.07	+9.57
<i>Rush hour</i>	40.44	6933	40.38	7596	-0.06	+9.56
Average	40.83	5581	40.79	5962	-0.04	+6.84

4.3.2. Rate-Distortion Performance Results. Evaluations were performed considering all the heuristics proposed in this work, which were integrated to the H.264/AVC reference software. All evaluations were performed for HD1080p (1920×1080 pixels) video sequences with groups of pictures (GOP) composed by one I frame followed by three P frames (IPPP), encoded with QPs 22, 27, 32, and 37. Tables 3 and 4 present average results for the proposed intra-frame mode decision and inter-frames mode decision, respectively. The first columns of each table present the results obtained through the application of the fully RDO-based decision, the central columns present the results obtained through the proposed method, and the final columns present a comparison between the two approaches in terms of image quality losses (Δ PSNR, in dB) and bit rate increase (Δ bits).

An average increase of 4.88% in bit rate and an average PSNR loss of 0.25 dB were noticed when the proposed intra-frame mode decision is used, in comparison to the RDO-based mode decision. When the proposed inter-frames mode decision is used, the average bit rate increase is around 6.84% and the average PSNR drop is under 0.04 dB. Even though such rate-distortion performance losses are not desirable, they are not significant when the computational complexity reductions provided by the fast decision method are taken into consideration.

5. Hardware Design for the Proposed Mode Decision Heuristics

Hardware architectures were designed considering the heuristics presented in the previous section. As previously stated,

the main goal of this work is to design hardware architectures for an H.264/AVC complete mode decision module which is able to process high-resolution video (such as HD1080p) in real time (30 frames per second).

5.1. Intra-Frame Mode Decision Architecture. The architecture for the intra-frame mode decision module was designed according to the hierarchical algorithm presented in Section 4.1. The architecture was designed considering the same hierarchical flow used in the algorithm itself. Two sub-modules are thus used in the architecture: (a) decision for equal block size and (b) decision for different block sizes. As previously explained, the decision for equal block sizes was designed using the SAD metric. The architecture designed to calculate the SAD distortion metric is presented in Figure 14, where O represents the original block and P represents the predicted block.

The SAD architecture performs the calculation of eight samples per clock cycle (two lines of a 4×4 block). Registers were used to decrease the critical path of the SAD calculator by generating a two-stage pipeline. The output register is responsible to accumulate partial SAD calculations, since only eight samples are processed per clock cycle. This way, a new complete SAD value is available at the output register after each two clock cycles. As the SAD calculation was used in all algorithms proposed in this work, the architecture presented in Figure 14 is the basic module in all presented architectures that use SAD. Figure 15 shows the block diagram of the whole intra-frame mode decision architecture.

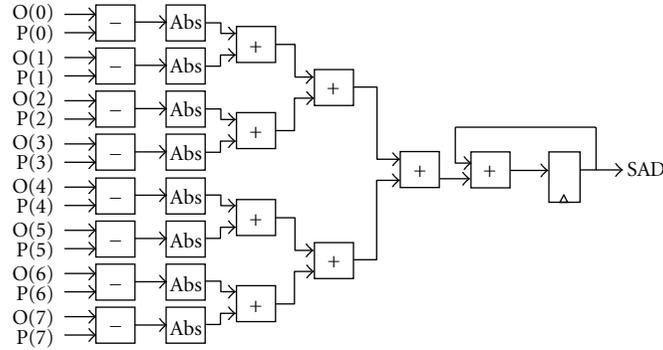


FIGURE 14: RTL diagram of the SAD calculator.

The architecture presented in Figure 15 is composed by 17 SAD calculators (nine for I4 MB mode decision, four for I16 MB mode decision, and four for chrominance block mode decision), three comparators which perform the decision between modes with same block sizes and the difference of distortion decision.

Distortion calculation for I4 MB mode decision is performed for the nine candidate modes ($SAD\ 0$ to $SAD\ 8$ in *I4 MB Decision* module in Figure 15). To allow the further decision between different block sizes (4×4 or 8×8), all the nine computed SAD values of one block are compared and the smallest one is summed up in the *I4 MB Accumulator* module. After the 16 smallest 4×4 SAD values belonging to one macroblock are computed, they are accumulated and stored before the second intra-frame decision step starts. In order to decide the best I16 MB mode, the four SAD values ($SAD\ 0$ to $SAD\ 3$ in the *I16 MB Decision* module) are compared and the smallest one is chosen by the comparator. No accumulator is necessary in this case, since only one 16×16 block composes the I16 MB. The chrominance mode decision selects the smallest SAD among the four chrominance prediction modes. The final decision of which is the best mode is performed in just one clock cycle and no other operation is necessary.

The *difference of distortion* module calculates the difference between the best accumulated I4 MB results and the best I16 MB result. The difference is then compared with the threshold value defined in Section 4.1 in order to perform the final intra-frame mode decision.

The intra-frame mode decision operation schedule is shown in Figure 16, where the number of clock cycles spent by each module is presented. Nonlabeled spaces correspond to periods in which the modules are not operational or their output results are not used.

As mentioned before, the SAD calculator was implemented with a two-stage pipeline and with eight input samples (two lines of a 4×4 block). This way, the architecture takes three clock cycles to deliver the first valid SAD considering a complete 4×4 block. When the pipeline is filled, a valid SAD result is available at the output at every two clock cycles. The architecture takes, therefore, 34 clock cycles to evaluate all nine I4 MB candidate modes and one more cycle to accumulate the last SAD result. Considering I16 MB mode decision, the SAD values are accumulated in the SAD

calculator itself (the output register in Figure 14). This way, the architecture takes 33 clock cycles (one less than the I4 MB mode decision) to accumulate the SAD for the four modes and one more cycle to compare them. The chrominance decision is similar to the I16 MB decision. However, as the block is composed only of 8×8 samples, the module takes only 10 clock cycles to perform the decision. Finally, when the distortion values for the best I4 MB modes and the best I16 MB mode are ready, the difference of distortion module evaluates these results in one clock cycle to decide which block size must be used. Then, the complete intra-frame decision architecture, in the worst case, uses 36 clock cycles to define which are the best prediction modes and the best block size.

5.2. Inter-Frames Mode Decision Architecture. The architecture for inter-frames mode decision was designed according to the hierarchical algorithm presented in Section 4.2. As the algorithm steps are not dependent on one another, all operations can be executed in parallel when a hardware design is considered. This way, in order to increase the architecture performance, the stationarity calculation for *SKIP* mode decision, the heterogeneity detection, the partition size decision and the sub-partition size decision are calculated concurrently in the proposed hardware architecture. It is important to notice that although all the decisions steps are always executed, some results may not be used since the inter-frames decision algorithm is hierarchical. For example, if the first decision step decides that the macroblock should be encoded as a *SKIP* macroblock, then all the other performed calculations are unnecessary and the results are discarded.

The inter-frames mode decision architecture is divided into the following modules: (1) stationarity-based *SKIP* mode decision, (2) partition/sub-partition detection, and (3) partition and sub-partition sizes decision.

5.2.1. Stationarity-Based SKIP Mode Decision Architecture. The *SKIP* mode decision is performed comparing the 16×16 macroblock in the current frame with the co-located macroblock in the reference frame. This comparison is performed using the SAD metric (calculated between current and co-located macroblock). This way, an instance of the architecture used for SAD calculations (see Figure 14) was also used to perform the *SKIP* mode decision. The SAD

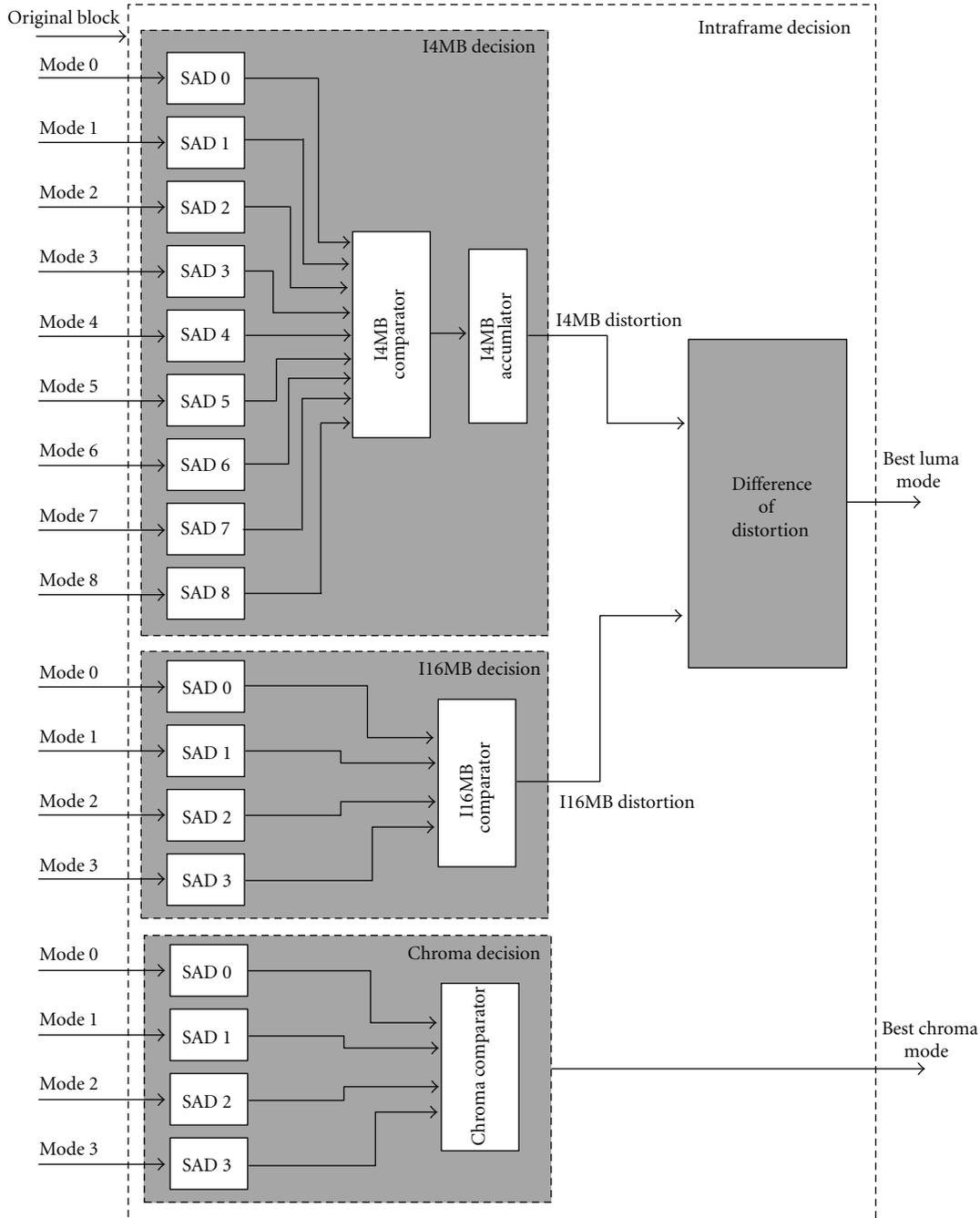


FIGURE 15: Block diagram of the intra-frame mode decision architecture.

calculation for a 16×16 macroblock consumes 32 clock cycles, since eight samples are processed in parallel (two lines of samples of a 4×4 block). One extra cycle is necessary to compare the SAD result with the threshold defined in Section 4.2.1 in order to define whether or not the macroblock should be encoded as *SKIP*. In the proposed architecture, this extra cycle is spent in the last step of the global inter-frames decision.

5.2.2. Heterogeneity-Based Subpartition Detection Architecture. As previously explained, the sub-partition detection

is performed through a heterogeneity metric, which is calculated using a DCT transform over the original 16×16 block (the current macroblock). As only some coefficients are needed to detect the heterogeneity level (top line and left column of coefficients), a partial 16×16 DCT architecture was designed in which unnecessary coefficients, not in the top and left column, are ignored. Besides the reduction in the number of coefficients, the DCT matrix was simplified to work with power-of-two operations. This is a very useful adaptation when hardware architectures are considered, since multiplications and divisions can be all replaced by

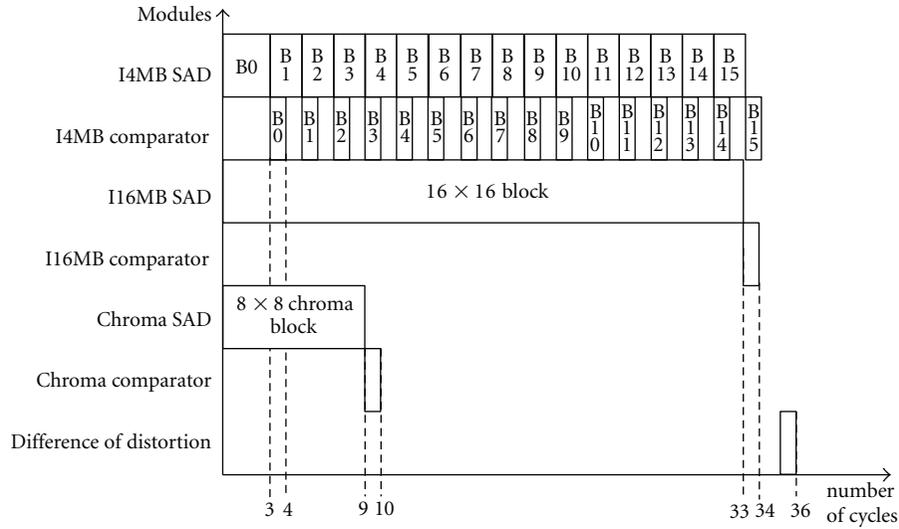


FIGURE 16: Schedule diagram of the intra-frame decision architecture.

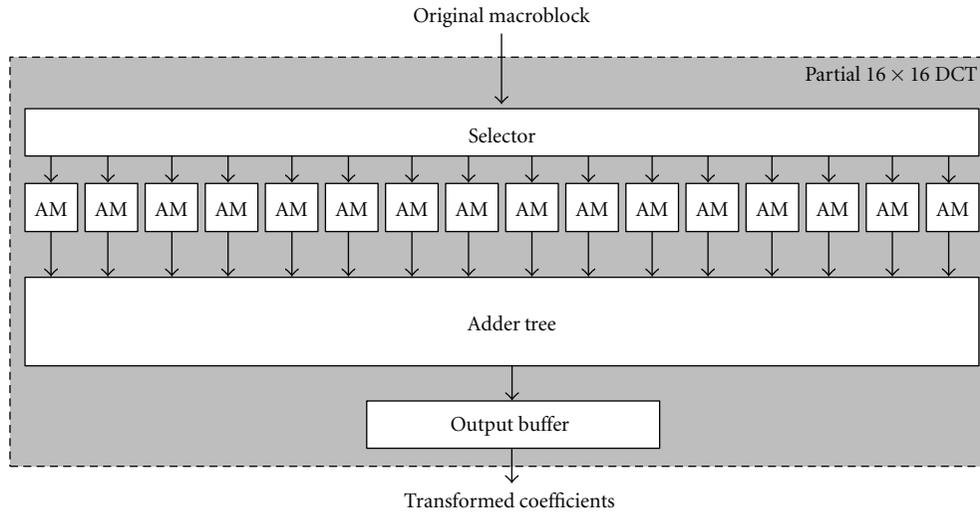


FIGURE 17: Block diagram of the partial 16×16 DCT.

simple shift operations (which introduce no hardware cost). Then, the DCT architecture implemented in this work is composed only by adders, subtractors, and shifters. Some coefficient calculations are very similar, so that it was also possible to reuse several operands and partial results. Figure 17 shows the block diagram of the architecture which generates all coefficients used by the heterogeneity metric.

The architecture was designed with a parallelism of 256 samples (i.e., the complete 16×16 original block must be available in the architecture input). However, as memory bandwidth is a common issue in such kind of systems, an input buffer was designed to group all samples before the architecture processing. In Figure 17, the *Selector* module is responsible for reordering the input samples and for delivering them to the *Adder Modules* (AM) input according to the calculation pattern. The AM modules perform all partial calculations which can be used in the calculation

of more than one coefficient. Each AM calculator was implemented with four pipeline stages in order to achieve the throughput requirements for real-time processing. The *adder tree* module performs the final calculations for each coefficient summing up the combined results from the AM modules and delivers the final coefficient results to the *output buffer*. Four clock cycles are needed to deliver one valid result from the *adder tree* to the *output buffer*. Finally, the buffer reorders the computed coefficients and delivers them to the final decision module.

5.2.3. Border Detection for Partition/Sub-Partition Decision Architecture. The architecture designed to calculate the border strength and to determine the partition and sub-partition sizes are similar to the SAD calculator used before. However, samples from the original block are used instead of the prediction residual. Figure 18 presents the architecture

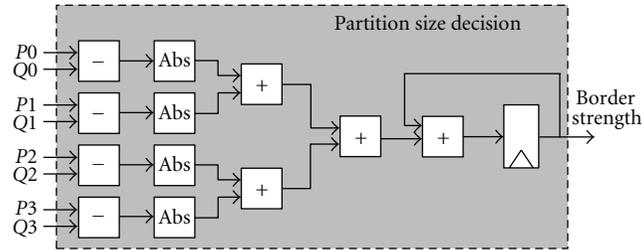


FIGURE 18: Architecture for partition border strength calculation.

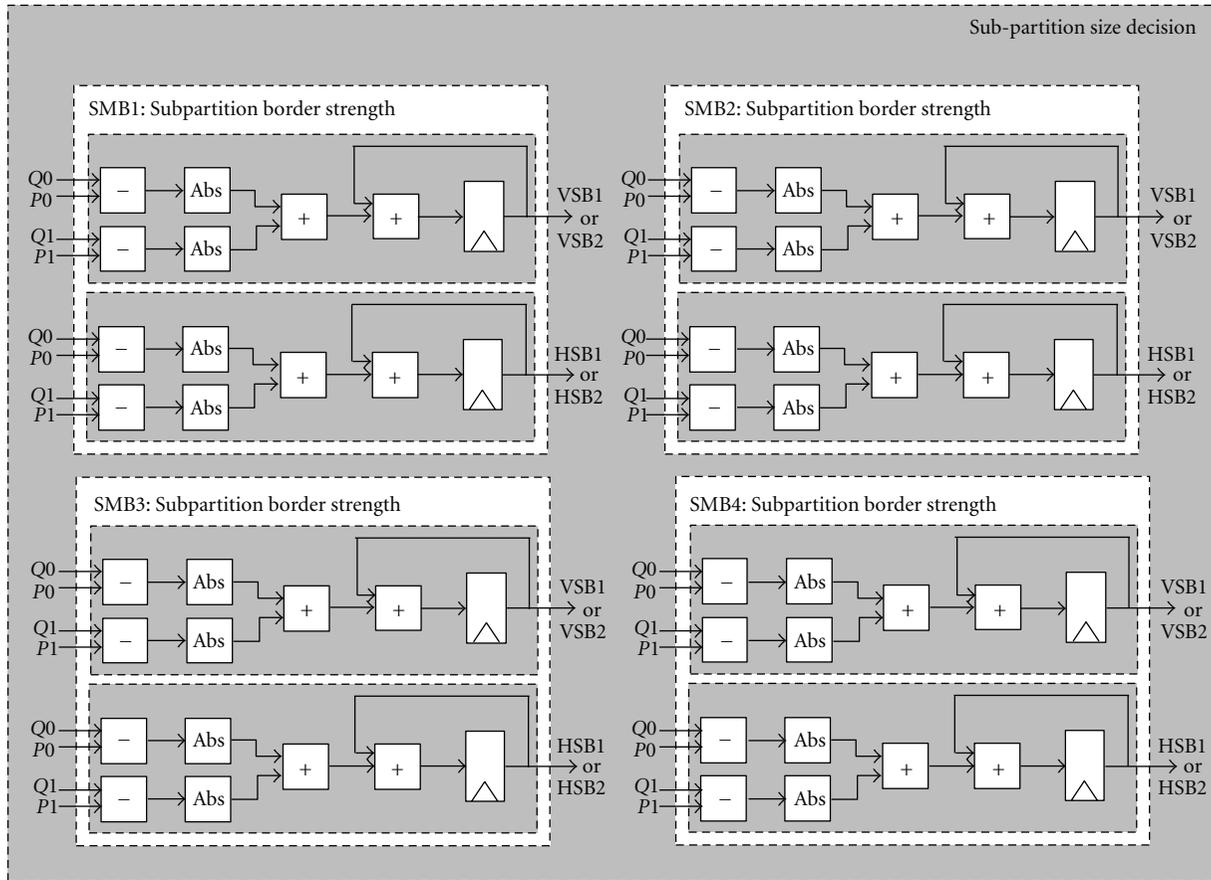


FIGURE 19: Architecture for sub-partition border strength calculation.

used to detect the partition border strength, where P_3, P_2, P_1 , and P_0 represent four samples on the left side of the possible partition border and Q_3, Q_2, Q_1 , and Q_0 represent samples on the right side of the possible partition border (for more details, see Figure 11). One clock cycle per 4×4 block line is required to compute the partition border strength, in a total of 16 clock cycles to perform the calculation of a complete border. After that, the border strength is used in the decision between 16×16 , 16×8 and 8×16 partition sizes, as explained in Section 4.2.3.

The architecture for sub-partition border strength calculation used to decide between 8×8 , 4×8 , 8×4 , and 4×4 sub-partitions is similar to that presented in Figure 18. However, fewer samples are used (only P_0, P_1, Q_0 , and Q_1). Figure 19

presents this architecture. Each submodule computes the subborder strengths corresponding to each sub-macroblock belonging to a macroblock (SMB1, SMB2, SMB3, and SMB4 in the figure). The inputs P_0, P_1, Q_0 , and Q_1 in each submodule represent the pixels p_0, p_1, p_2 , and p_3 belonging to each sub-macroblock, as previously presented in Figure 12. As explained in Section 4.2.4, the border strength calculation for sub-partition decision is performed in two steps. Four clock cycles are used to perform the calculation of each sub-border strength (defined as VSB1, VSB2, HSB1, and HSB2 in Section 4.2.4), totalizing 8 clock cycles to compute all sub-partition strengths. Two sub-border strengths per sub-macroblock are thus computed after each four clock cycles. After the first four clock cycles, VSB1 and HSB1 are the

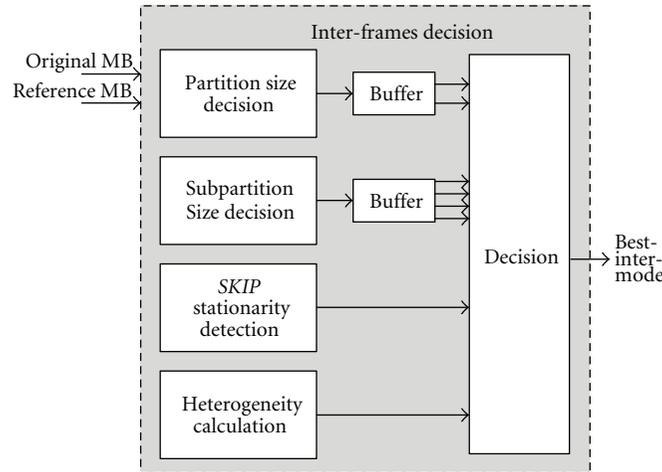


FIGURE 20: Block diagram of the interframes mode decision architecture.

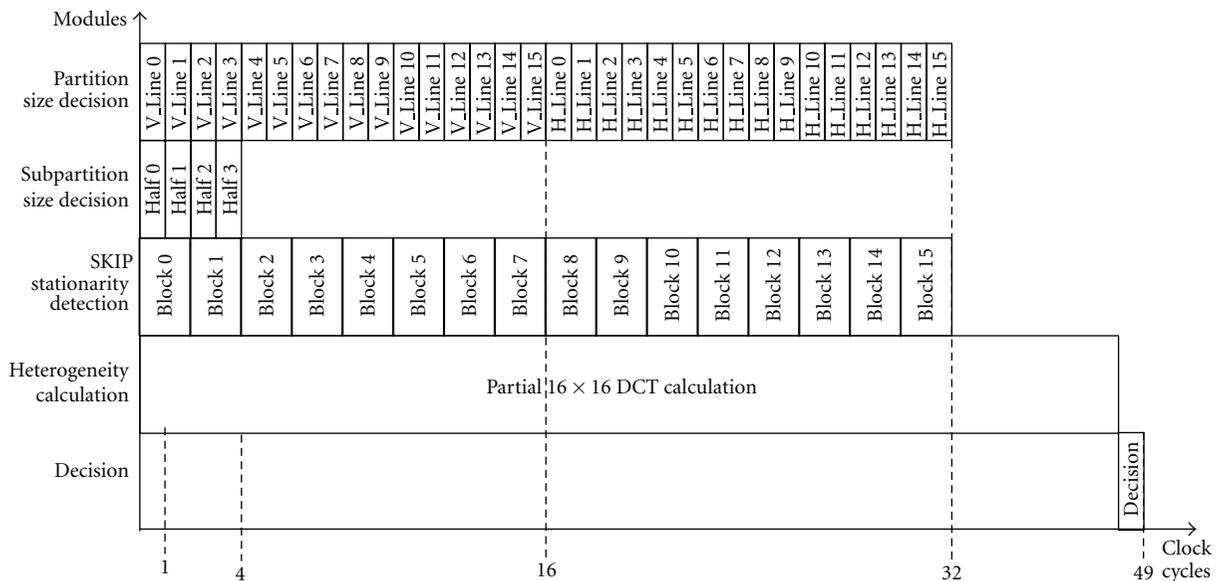


FIGURE 21: Schedule diagram of the inter-frames mode decision architecture.

available values at the output of each sub-module. At the end of the following four clock cycles, VSB2 and HSB2 are available at the output of the same sub-modules. Once all strengths are computed, they are finally compared to their respective threshold values and the final sub-partition size is chosen.

5.2.4. Complete Inter-Frames Decision Architecture. All architectures presented in the previous sections were integrated to generate the inter-frames decision module, as shown in Figure 20. The *partition size decision* module is responsible to calculate the horizontal and vertical border strengths, the *subpartition size decision* module performs sub-border strength calculations, the *stationarity detection* compares co-located blocks for *SKIP* mode decision, and the *heterogeneity calculation* performs the partial DCT calculation to decide whether or not sub-partitions are used. As previously

mentioned, all these operations are performed in parallel, since there is no data dependency between these operations. The results generated by these modules are then evaluated by the *Decision Module*, according to the hierarchical decision method proposed in Figure 8.

In order to provide a better understanding of the architecture working flow, Figure 21 shows the temporal diagram of the inter-frames mode decision. The partition size decision is able to process one block line per clock cycle (e.g., P0–P3 and Q0–Q3 of Figure 11). It means that 16 clock cycles are needed to process each complete border (vertical and horizontal), totalizing 32 clock cycles to generate both border strengths. The sub-partition size decision generates four sub-border strengths per clock cycle. This way, only four clock cycles are needed to generate the 16 border strengths for the whole macroblock. The *SKIP* stationarity detection module is able to perform the SAD calculation between the

TABLE 5: Synthesis results for the intraframe mode decision architecture.

Number of ALUTs	FPGA Altera Stratix II		TSMC 0.18 μm standard cells	
	Number of DLRs	Frequency (MHz)	Area (Number of gates)	Frequency (MHz)
3,267	2,312	98.43	28,518	129.10

TABLE 6: Synthesis results for the interframes mode decision architecture.

Number of ALUTs	FPGA Altera Stratix II		TSMC 0.18 μm Standard Cells	
	Number of DLRs	Max. frequency(MHz)	Area (number of gates)	Max. Frequency(MHz)
5,796	2,859	146.56	85,877	142.40

original and the reference macroblock in 32 clock cycles, since it takes 2 clock cycles to calculate the SAD for one 4×4 block. Finally, the heterogeneity calculation takes 48 clock cycles to generate all coefficients used in the sub-partition detection, and it is thus the critical path of the inter-frames mode decision architecture. The best mode is chosen in one final clock cycle, totalizing 49 cycles to complete the inter-frames mode decision process.

6. Architectural Synthesis Results

Both intra-frame and inter-frames mode decision architectures were synthesized targeting two different technologies in order to allow their use in different applications: (1) Altera Stratix II Field-Programmable Gate Array (FPGA) [23] (using Altera Quartus II) and (2) TSMC 0.18 μm standard cells library [24] (using Mentor Graphics Leonardo Spectrum [25]).

The synthesis results for intra-frame mode decision architecture are presented in Table 5. This architecture used 3,267 altera look-up tables (ALUTs) and 2,312 dedicated logic registers (DLRs) from the specified FPGA, which represent only 4% of the overall device resources. When targeting FPGA prototyping, the architecture reached an operation frequency higher than 98 MHz. The synthesis results for standard cells used less than 29 Kgates and it is able to run at 129.1 MHz.

Considering the highest operation frequency reached by the intra-frame decision architecture and the number of clock cycles used to process a complete macroblock in an HD1080p video sequence (discussed in Section 5.1), the architecture processing rate capability was calculated in frames per second (fps). The intra-frame mode decision is capable of processing up to 468 fps (HD1080p videos), which is much higher than the minimum requirements for real-time video coding. This way, in order to achieve a minimum frame rate of 30 frames per second, the architecture operation frequency can be downscaled to 8.5 MHz, significantly decreasing its energy consumption.

The inter-frames mode decision architecture synthesis results are presented in Table 6. This architecture used 5,796 ALUTs and 2,859 DLRs from the specified FPGA, reaching an operation frequency of 146.6 MHz. The standard cells synthesis reached an operation frequency of 142.4 MHz, using less than 86 Kgates. Considering the standard cells

TABLE 7: Comparison between the proposed mode decision and other related works.

Work	ΔPSNR (dB)	$\Delta\text{bit rate}$ (%)	$\Delta\text{complexity}$ (times)	Decision type
Lee et al. [11]	-0.14	+0.69	-4.7	Intra
Fengqin et al. [5]	-0.02	+3.00	-5.9	Intra
Lee and jeon [8]	-0.09	+0.09	-4.0	Intra
Jeon et al. [6]	-0.15	+4.00	-6.0	Intra
Proposed intra	-0.25	+4.88	-13.0	Intra
Wu et al. [12]	+0.03	+0.60	-6.0	Inter
Liquan et al. [10]	-0.07	+0.27	0.0	Inter
Proposed inter	-0.04	+6.84	-33.0	Inter

results of frequency and the number of clock cycles necessary to process each macroblock in P slices, the architecture is capable of processing up to 369 fps (HD1080p videos), which is more than enough for real-time applications. To achieve the minimum rate of 30 frames per second, the architecture frequency can be downscaled to 11.9 MHz.

7. Comparisons with Related Works

This section presents a comparison between the proposed method and other related works found in the literature (presented in Section 3). Table 7 presents results in terms of PSNR drop, bit rate increase, and mode decision computational complexity decrease, which is measured in number of iterations of the *encoding loop* previously defined. Complexity reduction results ($\Delta\text{complexity}$) for all works considered the worst case decisions and did not consider the complete GOP, so that intra-frame and inter-frames mode decision can be separately compared to the other corresponding intra-only or inter-only decision methods.

From Table 7, it is possible to perceive that the computational complexity reduction provided by the method proposed in this work is much larger than the reduction noticed in other related works, which allowed complexity reductions varying from 0 to 6 times in comparison to the RDO-based mode decision. This work provides a complexity reduction of 33 times when inter-frames mode decision is considered and a reduction of 13 times in computational

TABLE 8: Comparison between the proposed intra mode decision architecture and other works.

Work	Technology	Hardware resources	Cycles per MB	Max. frequency	HD1080 p fps	Min. frequency (HD1080 p)
Wang [15]	UMC 0.18 μm	10.302 gates	416	66 MHz	31	62.21 MHz
Kao [16]	TSMC 0.13 μm	11.229 gates	672	75 MHz	—	—
Lin [17]	TSMC 0.13 μm	94.700 gates*	560	140 MHz	30	140 MHz
This work	FPGA	3.267 ALUTs 2.312 DLRs	36	98.43 MHz	335	8.26 MHz
	TSMC 0.18 μm	28.518 gates	36	129.1 MHz	439	8.26 MHz

* Hardware resources corresponding to the complete intra prediction module.

complexity when the proposed intra-mode decision is used. On the other hand, the expressive reduction in computational complexity is only possible at the cost of a higher loss in compression efficiency. Nevertheless, such losses are not significantly higher than the losses noticed in other related works which provided a much lower computational complexity reduction.

Table 8 presents a comparison between the proposed intra mode decision architecture and other works found in the literature [15–17]. To the best of the authors' knowledge, no inter mode decision architecture implemented independently from the motion estimation module was published in the literature so far.

The proposed intra mode decision architecture requires the smallest number of clock cycles to process an intra macroblock, which represents a reduction of more than 11 times in comparison with [17], and a reduction of more than 18 times when compared with [16]. The architecture also presents the highest throughput among the related works: an increase of more than 11 times and 14 times in the number of HD1080p frames encoded per second for FPGA and TSMC 0.18 versions, respectively, compared with [15]. All the results presented for the proposed intra mode decision architecture were obtained considering maximum frequency operation. The higher throughput achieved with this architecture allows reducing the operating frequency down to 8.26 MHz while still processing HD1080p videos at 30 fps. With such low operation frequency, very low power can be achieved with the architecture, which is an excellent alternative for battery-powered devices.

8. Conclusions

This work proposed a set of heuristics targeting fast mode decision for H.264/AVC and their respective VLSI architectures. The proposed method is based on a hierarchical organization of these mode decision heuristics, focusing on the acceleration of the best mode decision for computational and power-constrained devices. Luminance and chrominance intra-frame decisions were divided into two steps, one of them based on the distortion to choose the best mode for a fixed block size and the other one based on difference of such distortions to choose the best block size. Inter-frames mode decision was divided into a stationarity-based earlier SKIP mode decision, a heterogeneity-based sub-partition detection, and two-border strength-based partition and sub-partition sizes decision.

The mode decision computational complexity was decreased in 47 times through the application of the proposed method in comparison to the RDO-based mode decision. On the other hand, a relatively small increase in bit rate and a negligible decrease in PSNR were noticed. To the best of the authors' knowledge, no other previous work found in the literature is capable of reducing computational complexity in more than 6 times in comparison to the RDO technique.

The developed hardware architectures for both decision modules have shown that the proposed method is capable of processing high-resolution video sequences in real time. The minimum requirement of processing at least 30 frames per second is more than fulfilled, which means that the maximum operation frequency can be downscaled for further energy consumption reduction, which is extremely important for portable devices with limited battery resources.

The obtained results encourage the use of the proposed mode decision algorithms and their respective hardware architectures in H.264/AVC encoders applied to complex multimedia systems with support to high-resolution videos. The fast mode decision is especially useful in energy and computationally limited devices, such as smartphones, video cameras, portable TVs, and other multimedia embedded systems.

References

- [1] International Telecommunication Union (ITU), *ITU-T Recommendation H.264/AVC (05/03): Advanced Video Coding for Generic Audiovisual Services*, International Telecommunication Union, Geneva, Switzerland, 2003.
- [2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.
- [3] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for: Video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, 1998.
- [4] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 688–703, 2003.
- [5] W. Fengqin, F. Yangyu, L. Yajing, and L. Weihua, "Fast intra mode decision algorithm in H.264/AVC using characteristics of transformed coefficients," in *Proceedings of the 5th IEEE International Conference on Visual Information Engineering (VIE '08)*, pp. 245–249, July 2008.
- [6] Y. I. Jeon, C. H. Han, S. W. Lee, and H. S. Kang, "Fast intra mode decision algorithm using directional gradients for

- H.264,” in *Proceedings of the 2009 2nd International Congress on Image and Signal Processing (CISP '09)*, October 2009.
- [7] H. Kim and Y. Altunbasak, “Low-complexity macroblock mode selection for H.264/AVC encoders,” in *Proceedings of the International Conference on Image Processing (ICIP '04)*, pp. 765–768, Atlanta, Ga, USA, October 2004.
- [8] J. Lee and B. Jeon, “Fast mode decision for H.264,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '04)*, pp. 1131–1134, June 2004.
- [9] Y. M. Lee, J. D. Wu, and Y. Lin, “An improved SATD-based intra mode decision algorithm for H.264/AVC,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 1029–1032, April 2009.
- [10] S. Liqun, L. Zhi, Z. Zhaoyang, and S. Xuli, “Fast inter mode decision using spatial property of motion field,” *IEEE Transactions on Multimedia*, vol. 10, no. 6, Article ID 4657454, pp. 1208–1214, 2008.
- [11] Y. M. Lee, Y. T. Sun, and Y. Lin, “SATD-based intra mode decision for H.264/AVC video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 3, Article ID 5308396, pp. 463–469, 2010.
- [12] D. Wu, F. Pan, K. P. Lim et al., “Fast intermode decision in H.264/AVC video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 7, pp. 953–958, 2005.
- [13] I. Richardson, *H.264/AVC and MPEG-4 Video Compression-Video Coding for Next-Generation Multimedia*, John Wiley and Sons, Chichester, UK, 2003.
- [14] K. Suhring, “H.264/AVC Reference Software,” In: Fraunhofer Heinrich-Hertz-Institute, <http://iphome.hhi.de/suehring/tml/download/>.
- [15] J. C. Wang, J. F. Wang, J. F. Yang, and J. T. Chen, “A fast mode decision algorithm and its vlsi design for H.264/AVC intra-prediction,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 10, pp. 1414–1422, 2007.
- [16] Y. C. Kao, H. C. Kuo, Y. T. Lin et al., “A high-performance VLSI architecture for intra prediction and mode decision in H.264/AVC video encoding,” in *Proceedings of the 2006 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS '06)*, pp. 562–565, December 2006.
- [17] Y. K. Lin, C. W. Ku, D. W. Li, and T. S. Chang, “A 140-MHz 94 K gates HD1080p 30-frames/s intra-only profile H.264 encoder,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 3, Article ID 4783023, pp. 432–436, 2009.
- [18] M. Weipeng, Y. Shuyuan, G. Li, and P. Chaoke, “An efficient fast mode decision algorithm based on motion cost for h.264 inter prediction,” in *Proceedings of the 2nd International Symposium on Intelligent Information Technology Application Workshop (IITA 2008)*, pp. 550–553, December 2008.
- [19] A. C. W. Yu, G. R. Martin, and H. Park, “Fast inter-mode selection in the H.264/AVC standard using a hierarchical decision process,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 2, Article ID 4400036, pp. 186–195, 2008.
- [20] A. C. Yu, “Efficient block-size selection algorithm for inter-frame coding in H.264/MPEG-4 AVC,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. III169–III172, May 2004.
- [21] G. Corrêa, C. Diniz, S. Bampi, D. Palomino, R. Porto, and L. Agostini, “Homogeneity and distortion-based intra mode decision architecture for H.264/AVC,” in *Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems (ICECS '10)*, pp. 591–594, December 2010.
- [22] D. Palomino, G. Corrêa, Claudio Diniz, S. Bampi, L. Agostini, and A. Susin, “Algorithm and hardware design of a fast intra-frame mode decision module for H.264/AVC encoders,” in *Proceedings of the 24th Symposium on Integrated Circuits and Systems Design (SBCCI '11)*, August 2011.
- [23] Altera Corporation: FPGA, CPLD and ASIC, <http://www.altera.com/>.
- [24] Taiwan Semiconductor Manufacturing Company Limited, <http://www.tsmc.com/>.
- [25] The EDA Technology Leader-Mentor Graphics, <http://www.mentor.com/>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

