

# Fast mode decision on H.264/AVC baseline profile for real-time performance

Marcos Nieto · Luis Salgado · Julián Cabrera ·  
Narciso García

**Abstract** In this paper a new fast mode decision (FMD) algorithm is proposed for the recent H.264/AVC video coding standard, aiming to reduce its computational load without losing coding efficiency. This algorithm identifies redundancy and selects the minimum sub-set of modes for each macroblock (MB) required to provide high rate-distortion (RD) efficiency. It is based on a fast analysis of the histogram of the difference image between frames which classifies the areas of each frame as active or non-active by means of an adaptive thresholding technique. More coding effort is devoted to active areas with the selection of a large sub-set of Modes, as these areas are expected to be the most relevant in terms of RD cost. Results show reduction values around 35–65% of motion estimation (ME) time, preserving the RD cost for the Baseline Profile, by using P-Slices and without needing B-Slices. Moreover, the strategy works as an intelligent tool for real-time applications with constrained number of operations per frame: it wisely uses the given operational resources distributing them among those MBs that need it.

**Keywords** H.264/AVC · Real-time · Fast mode decision · Histogram-based segmentation · Adaptive thresholding

## 1 Introduction

The new international video coding standard H.264/AVC has been developed to improve video coding efficiency compared to previous standards such as MPEG-2 or recent MPEG-4. As the motion estimation (ME) phase is one of the key elements of a video encoder to achieve good rate-distortion (RD) performance, it has been greatly intensified in H.264/AVC by including some new efficient features like multiple reference frames and tree structured motion compensation. In previous standards, frames are divided into fixed size macroblocks (MB) as the basic ME unit. H.264/AVC allows the subdivision of MBs into smaller blocks, down to  $4 \times 4$  pixels, creating a full set of inter-prediction alternatives, usually called Modes, as shown in Fig. 1. The encoder selects the most appropriate choice between using large blocks with just few motion vectors, or smaller blocks and more motion vectors to transmit. A common approach to tackle this problem is based on RD optimization applying the Lagrangian multiplier technique computing the RD cost of all inter-prediction alternatives.

MBs corresponding to areas where there are few changes between frames usually get modes corresponding to large blocks, such as Mode 1 or SKIP-Mode, with just one motion vector, while in areas of detailed motion smaller partitions are more efficient

However, the complete analysis of all these Modes for each MB requires an enormous computational load that makes H.264/AVC much more complex than any other video codec. Therefore, much effort is currently being

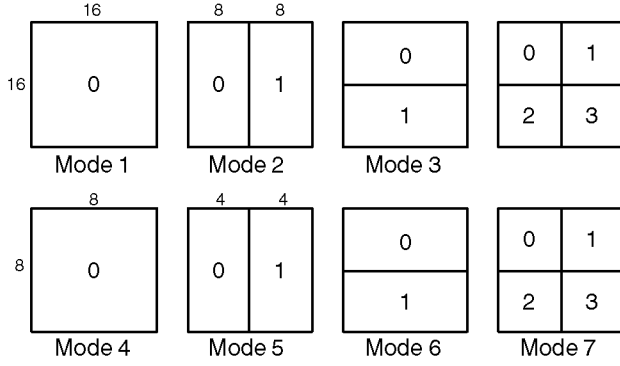


Fig. 1 H.264/AVC inter prediction modes

done by researchers to reduce this complexity, mainly by designing and developing fast mode decision (FMD) techniques. Most of these works study the RD cost of one initial mode and according to the result, continue or not the analysis of the rest of modes. On the contrary, there are other works, like [10], that efficiently performs Inter/Intra coding mode selection, which can be used as a basis to improve results by including into it a FMD algorithm.

The new FMD strategy proposed in this work automatically selects, previously to the ME phase, the best sub-set of Modes to be computed for each MB according to the distribution of activity within it.

It is specifically designed to be implemented into real-time encoders with almost negligible lost of coding efficiency. The FMD strategy relies on the assumption that the analysis of all modes is highly redundant in low activity areas of the image because many modes are likely to achieve very similar low RD costs [11]. The FMD algorithm carries out the identification of these areas previously to the ME process (which means that the algorithm does not compute transformation and quantization, neither VLC or inverse transformation and quantization [12]), as shown in Fig. 2. Based on this identification, the minimum sub-set of modes that each MB requires to offer good RD performance is selected so that only those modes are applied to each MB at the ME phase.

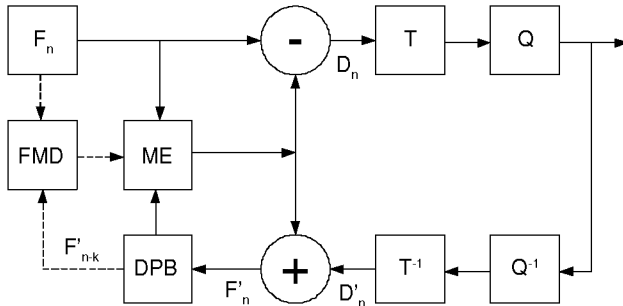


Fig. 2 Simplified scheme of encoder H.264/AVC with the proposed FMD module

The content of the paper is organized as follows: Sect. 2 depicts the whole FMD proposal; Sect. 3 describes the activity analysis process, including difference analysis, thresholding phase and multi-reference computation; Sect. 4 presents the algorithm to select the sub-set of modes to be computed; Sect. 5 shows the results obtained related to computational load reduction and RD cost and Sect. 6 presents the obtained conclusions.

## 2 FMD overview

The proposed FMD strategy is composed by three main processing phases, which are depicted in Fig. 3: (1) the first module analyzes the activity of the input frame,  $F_n$ , by computing the differences between  $F_n$  and its reference frames at a  $4 \times 4$  pixels block level. The output is the image  $B_n^K$ , which is a mask that identifies the high activity regions of  $F_n$ ; (2) once the activity of the input frame is well described, each MB of the frame is independently analyzed in order to select the minimum sub-set of modes,  $S_n$ , that ensures good RD performance: the idea is to devote more coding effort, i.e., more modes, to those MBs that contain more activity; and (3) given a real-time constraint that limits the number of operations to be done at the ME stage of the encoder, and understanding this value as a resource, the algorithm wisely distributes this resource among those MBs that requires more coding effort to achieve good RD performance.

Particularly, the phase of activity analysis is subdivided into three different sub-modules, which are shown in Fig. 4. Firstly, the activity image,  $A_n$  is created using only the differences between  $F_n$  and  $F_{n-1}$  at a  $4 \times 4$  pixels block level. Derived from  $A_n$ , the energy distribution function  $CH_n$  is obtained and used to classify the frame under study according to its activity level, which is described with the parameter  $\alpha$ .

Each activity level has an associated automatic thresholding technique that binarizes  $A_n$  which becomes  $B_n$ . At this point,  $B_n$  identifies, with a binary value (active or non-active), the activity of each  $4 \times 4$  pixels block of  $F_n$ .

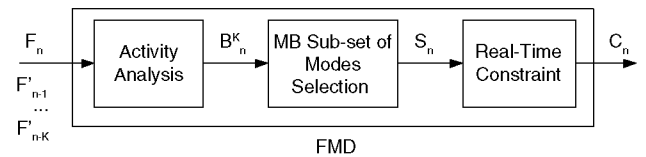
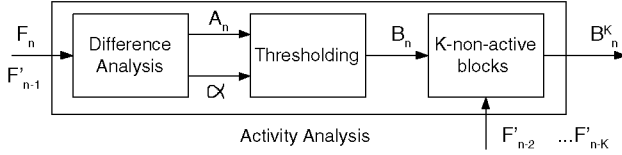


Fig. 3 Block diagram showing the processing phases of the proposed FMD strategy;  $B_n^K$  is the mask which identifies active areas of the frame  $F_n$  taking into account the K previous reference frames;  $S_n$  represents the sub-sets of modes for each MB, while  $C_n$  are the sub-sets of modes considering the real-time constraint



**Fig. 4** Detail of the activity analysis phase;  $A_n$  is the difference image for the frame  $F_n$ ,  $\alpha$  is the parameter that describes the distribution of activity within  $A_n$ , while  $B_n$  is the binary activity image after the thresholding phase. The multi-reference is used to compose the  $B_n^K$  activity image where taking into account previous reference frames, from  $F_{n-2}'$  to  $F_{n-K}'$

The mode selection phase analyzes the 16 values of activity corresponding to each MB and determines which modes are more likely to achieve good RD performance, and which are redundant:

- The more non-active blocks contained in a  $16 \times 16$  pixels MB are, the less modes will be selected by the FMD and then computed at the ME process, implying high computational cost saving.
- The more active blocks are, the more uncertainty is about which modes will provide better RD performance, so that the FMD selects large sub-set of modes not to loose RD efficiency even when not achieving significative reduction of computational cost.

However, before the mode selection phase, the algorithm exploits the multi-reference feature of H.264/AVC to transform some active blocks into non-active with respect to reference frames other than  $F_{n-1}'$ .

Our experimental results show that the proposed FMD reduces dramatically the overall ME computational load so that real-time implementations are feasible while keeping the average RD cost almost equal to that obtained with the analysis of the complete set of modes.

### 3 Activity analysis

The objective of this module is to provide a mask,  $B_n^K$  which identifies the activity regions of each frame. For that purpose, the differences between  $F_n$  and  $F_{n-1}'$  are computed, generating the activity image,  $A_n$ . A single thresholding technique would lead to the binarization of this image, but it will not be adapted to the activity level of the frame. Therefore, in this work we propose to apply different thresholding techniques according to the distribution of activity in each frame.

The reason of using of the direct difference between frames as a feature to classify frames is twofold: (i) it is a easy and fast feature to extract, and (ii) it is enough representative of the activity of the sequence at each frame.

The classification of the frames is done by computing the histogram of  $A_n$  and modeling the distribution of

activity levels with an energy function,  $CH(i)$ . According to the value of the parameters of the model, the frame is classified as having *low*, *medium* or *high* activity level.

Once the type of activity is identified, an appropriate thresholding technique is used providing a binarization of  $A_n$  that ensures that only relevant activity areas are set as active, while those areas with less significant activity are set as non-active.

Finally, the analysis of the multi-reference will change this binarization  $B_n$  into  $B_n^K$  which is not binary as it contains different types of blocks: active, non-active and the k-non-actives.

#### 3.1 Difference analysis

The difference image,  $A_n$  is computed using the current original frame,  $F_n$ , and the immediately previous encoded and reconstructed frame,  $F_{n-1}'$ . The pixels values of the difference image are obtained as the MAE between these frames, computed on  $4 \times 4$  pixel blocks:

$$MAE = \frac{1}{16} \sum_{i=0}^3 \sum_{j=0}^3 |B_{ij}^n - B_{ij}^{n-1}| \quad (1)$$

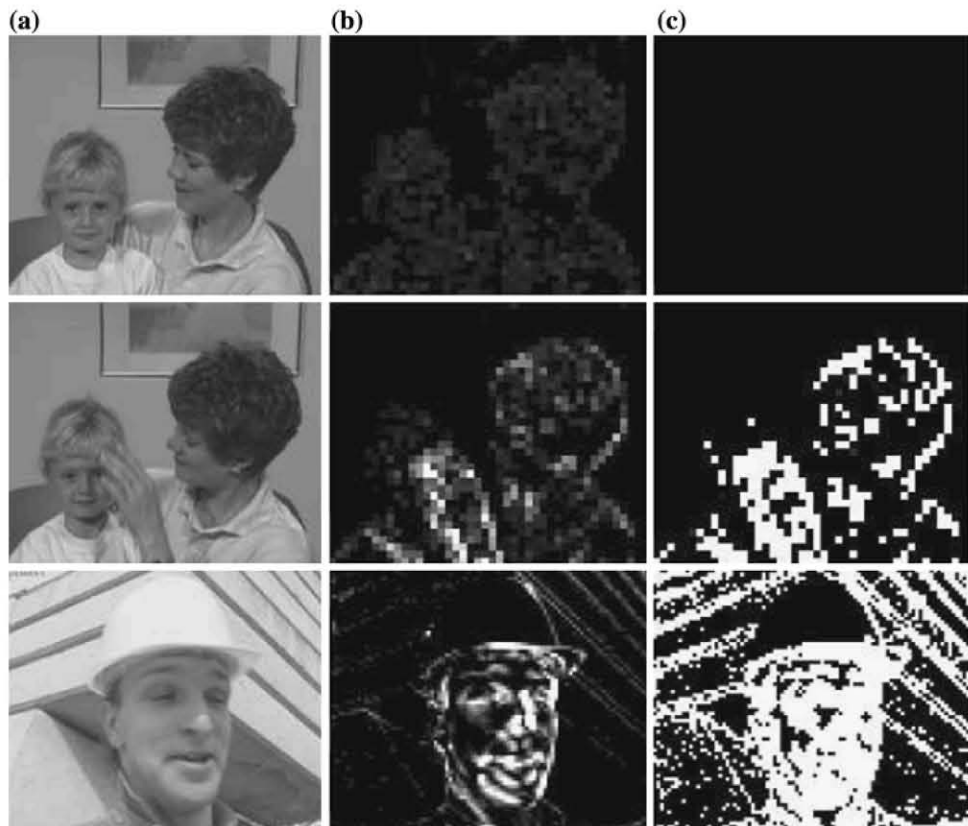
where  $B^n$  and  $B^{n-1}$  represent  $4 \times 4$  blocks belonging to  $F_n$  and  $F_{n-1}'$ , respectively. Therefore, the activity image resolution is  $4 \times 4$  times lower than that of the sequence pictures. As  $4 \times 4$  pixels blocks are the lower sub-MB partition allowed in H.264/AVC, the computation of the MAE on  $4 \times 4$  blocks is applied to significantly simplify the selection of the sub-set of modes process (introduced in Sect. 4). As a result, each  $16 \times 16$  pixels MB of the frame to be encoded is characterized by 16 activity values (one for each  $4 \times 4$  block of the MB).

Figure 5b shows examples of  $A_n$  corresponding to different situations (note that their resolution has been increased for better visualization, but their real resolution is  $4 \times 4$  times lower than that of the original frames): first row (Fig. 5b) shows the activity image between two almost static consecutive frames of *Mother & Daughter* sequence: the result is a picture with low activity values homogeneously distributed along the whole picture. Second and third row (Fig. 5b) show, for *Mother & Daughter* and *Foreman* sequences, respectively, the activity image between two consecutive frames with significant changes: in these situations, the main part of the activity is concentrated on small areas of the image.

##### 3.1.1 Histogram analysis

As shown in the previous section there is a coarse classification of frames according to their activity level: low,

**Fig. 5** First row *Mother & Daughter* sequence (quasi-static frames), second row *Mother & Daughter* sequence (non-static frames); and third row *Foreman* (high activity frames): Column **a**  $F_n$ ; **b**  $A_n$  (with scaled levels for visualization) between original frame and immediately previous reconstructed frame, and **c**  $B_n^K$



medium and high activity. In all cases, the histogram is unimodal, and it can be considered unimodal for almost all type of sequences: even for high activity images, a significant part of the picture shows small changes at a  $4 \times 4$  pixel block level, and the more significant activity is distributed among many different levels of the histogram (i.e., different activity levels). Some examples are shown in Fig. 6.

The mode of the histogram is usually located on a MAE value close to zero, but not zero, as it is very strange to have zero-activity, due to decoding artifacts, noise, or acquisition properties of the camera.

### 3.1.2 Activity classification

Activity images  $A_n$  are classified automatically into these three categories. The classification is focused on the distribution of the activity values, which is described with the following function: the *normalized cumulative histogram of quadratic values*,  $CH(i)$ , whose expression is:

$$CH(i) = \frac{\sum_{k=0}^i k^2 \cdot P(k)}{\sum_{k=0}^I k^2 \cdot P(k)} \quad (2)$$

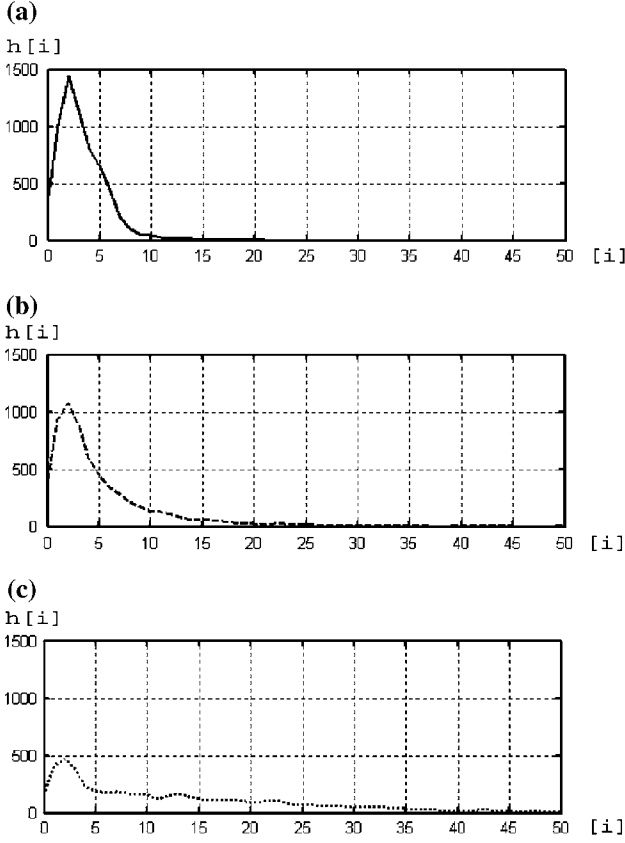
where  $P(k)$  represents the histogram probabilities and  $I$  is typically 255 for one-byte gray images. Some examples of  $CH(i)$  are shown in Fig. 7a–c in solid lines (only the zoom

to the values from 0 to 100 is shown in abscises, as there is not activity values reaching 255; while in ordinates the values are percentages). As it can be observed, these are monotonous growing functions where the last non-zero value of the activity histogram,  $i_{\text{last}}$ , is the minimum  $i$  value where  $CH(i_{\text{last}}) = 1$ . For  $i \geq i_{\text{last}}$   $CH(i) = 1$ .

The difference of activity level is depicted at the shape of the  $CH(i)$  functions: for low activity frames,  $CH(i)$  grows from 0 to 1 in just a few values, i.e., the whole activity energy is concentrated in the first MAE values (the values around the mode of the histogram), which means that there is almost no activity in the frame.

On the other hand, medium and high activity frames correspond to  $CH(i)$  functions which are much wider, which means that although their corresponding histograms are unimodal, the significant activity levels are distributed in more values than the ones located near the principal mode.

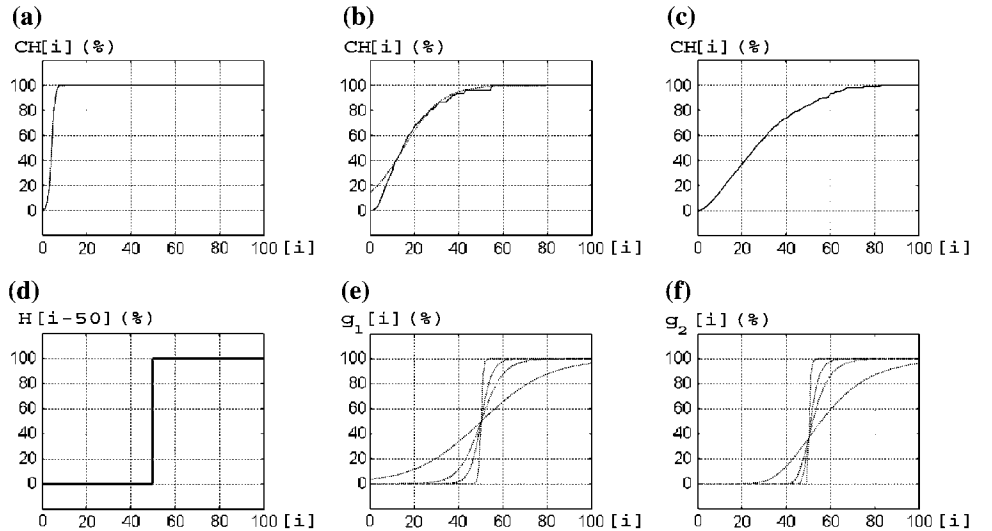
Therefore, the main difference is between low activity and high activity frames, whereas medium activity is a transition state between them. So, the classification is performed based on these  $CH(i)$  functions, which are modeled and then numerically parameterized. The value of the parameter of the best model for each  $CH(i)$  is used to classify it according to a set of fixed thresholds empirically obtained through the analysis of several standard test sequences.



**Fig. 6** Histograms,  $h[i]$ , of activity images (showing values from  $i = 0$  to 50). There are differences at the length of the tail and the number of values around the mode: **a** low activity frame from *Mother & Daughter* sequence; **b** medium activity frame from *Mother & Daughter* sequence; and **c** high activity frame from *Foreman* sequence

$CH(i)$  curves are parameterized with a model based on a limit approximation function of the Heaviside function

**Fig. 7**  $CH(i)$  distributions, in solid lines, and  $g(i)$  in dotted lines, for: **a** low activity frames; **b** medium activity frames, and **c** high activity frames; **d** Heaviside function; **e** examples of model function  $g_1(i)$ , with  $\alpha = 0.5, 2.5, 5, 15$ ; and **f** examples of model function  $g_2(i)$



$$H(i - \mu) = \lim_{\alpha \rightarrow 0} (g_j(i)) \quad (3)$$

where  $g_j$  is the function used to model  $CH(i)$ ,  $j$  is the type of function used,  $\mu$  is the location parameter of the Heaviside function, and  $\alpha$  is the parameter that will be further used to classify the activity of the frame. Then, by determining an appropriate model function  $g_j(i)$  for  $CH(i)$ , it is possible to identify the type of activity through the value of the  $\alpha$  parameter selected.

Among the different existing limit functions  $g_j(i)$ , those presented in (4) and (5) have been chosen as they provide very good approximations of  $CH(i)$  for all the tested sequences as well as for their simplicity. Figure 7d shows the Heaviside function corresponding to  $\mu = 50$ ; Fig. 7e, f show model functions  $g_1$  and  $g_2$ , respectively, for several values of  $\alpha$ . Low values of  $\alpha$  correspond to model functions very close to the step function, whereas high values of  $\alpha$  define curves with softer slopes.

$$g_1(i, \alpha) = \frac{1}{1 + e^{-\frac{i-\mu}{\alpha}}} \quad (4)$$

$$g_2(i, \alpha) = e^{-e^{-\frac{i-\mu}{\alpha}}} \quad (5)$$

For each frame, the best approximation to  $CH(i)$  will be defined choosing what  $g(i, \alpha)$  [either  $g_1(i, \alpha)$  or  $g_2(i, \alpha)$ ] and what value of  $\alpha$  minimizes the error area,  $\Delta Area(\alpha)$ , defined as:

$$\Delta Area(\alpha) = \sum_{i=0}^I |CH(i) - g(i)|. \quad (6)$$

For both  $g_1(i, \alpha)$  and  $g_2(i, \alpha)$  several values of  $\alpha$  are tested resulting in different values of  $\Delta Area(\alpha)$ . The tested values are those obtained with the following equations, derived from Eqs. (4) and (5), with  $i$  varying from 0 to the first  $i$  value where  $CH(i) = 1$ :

$$\alpha_1(i) = -\frac{i - \mu}{\ln\left(\frac{100}{CH(i)} - 1\right)} \quad (7)$$

$$\alpha_2(i) = -\frac{i - \mu}{\ln\left(-\ln\left(\frac{CH(i)}{100}\right)\right)} \quad (8)$$

where  $\alpha_1(i)$  and  $\alpha_2(i)$  are the tested  $\alpha$  values respectively for  $g_1$  and  $g_2$ .

To simplify calculations,  $\mu$  is selected as the value of  $i$  where  $CH(i) = 0.5$ . This way,  $\alpha(i)$  is tested for integer  $i$  values from the value of  $i$  that makes  $CH(i) = 0.1$  to  $CH(i) = 0.9$ . In average, for a high activity frame (where the distribution is wider),  $50 \times 2$  (two model functions)  $\alpha$  values are tested.

Although absolute minimization of the error area is not ensured, the proposed approximation has shown excellent results reducing dramatically the computational cost associated to the determination of the model function, thus not affecting the real-time performance.

The alpha value selected is correlated with RD performance as it is shown in Fig. 8. This figure shows three examples of the sequence *Paris* (that does not belong to the set of training sequences), for the QPs values 24, 28 and 32. The RD cost shown is normalized and depicts the average RD cost for the full sequence (150 frames). These curves are decreasing only: higher  $\alpha_i$  values imply that more frames are considered as *High activity* frames (with  $\alpha_i \geq 10$ ), so that the algorithm automatically adapts the reduction of computational cost to avoid high RD deviations.

This selection provides a good trade-off between low RD cost and high reduction of computational cost, compared with other values of  $\alpha$ : increasing the  $\alpha_i$  above the  $\alpha$  (the one automatically selected by the algorithm) value does not ensure a significant RD cost improvement while the gain in computational cost is seriously affected. On the other side for low  $\alpha_i$  values, more reduction of computational cost is achieved, at the cost of high RD deviation.

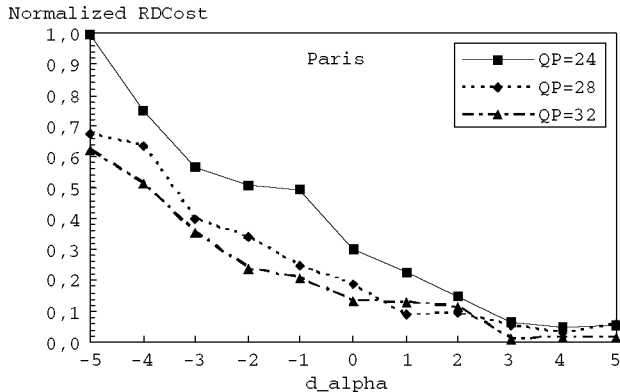


Fig. 8 Normalized average RD cost obtained with  $\alpha_i$  values different from the selected as best  $\alpha$  value by adding  $d\_alpha$  ( $\alpha_i = \alpha + d\_alpha$ )

Once the function and parameter offering the minimum  $\Delta Area(\alpha)$  are identified, we have found the best model of the  $CH(i)$  function. Figure 7a–c show three examples of  $CH(i)$  with each respective computed model  $g_j(i)$  in dotted line. For the three examples, the curve that fits better is  $g_1$ , but for other tested frames,  $g_2$  offers better results, specially for those distributions not symmetrical around  $CH(i) = 0.5$ . As it is shown, the accuracy of the models is very high (the difference between  $CH(i)$  and  $g_i$  is almost negligible), specially for medium and high values of  $i$ , except for the example shown in Fig. 7b. This is particularly significant as these values correspond to the most important part of the activity between frames, and therefore, need to be accurately estimated.

The classification of the frames is done accordingly to the analysis of the parameters obtained experimentally for several long standard test sequences (like *Tempete*, *Foreman*, *Coastguard*, *Mother & Daughter* and others), taken into account different codification parameters, such as high and low QP values, constant and variable bitrate, different levels and profiles, etc. As a result of the analysis, the values of  $\alpha$  that best classify images regarding their activity are those shown in Table 1.

### 3.2 Thresholding techniques

For each type of frame we propose different thresholding techniques that binarize  $A_n$  into  $B_n$  setting as active only those blocks containing relevant activity. Specifically, these thresholding techniques have been defined according to the expected threshold value that they should offer for each type of frame. The following criterion is used by analyzing the histogram of each type of frame:

- For low activity frames, almost all the energy of the activity values is concentrated in a few values around the mode (we will call this as the “main cluster”), before the curve falls. Therefore, the threshold should be approximately the value of the corner of the histogram. Active blocks should be only those higher than the distribution around the mode.
- For medium activity frames, activity is more distributed, and the main cluster is usually wider. The threshold should be selected lower than the corner of the histogram distribution.

Table 1 Histogram classification

Parameter $\alpha$	Classification
$0 \leq \alpha < 2.5$	Low activity
$2.5 \leq \alpha < 10$	Medium activity
$\alpha \geq 10$	High activity

- For high activity frames, activity is even more distributed: the main cluster is wider than in the previous cases, and the tail of the histogram contains a significant number of blocks with high activity. The threshold should be placed near the mode of the histogram, setting as non-active only the lower activity values of the distribution.

Threshold selection has been a well known problem for image processing for many years and multiple solutions have been proposed to binarize gray level images. However, most of the automatic thresholding techniques do not offer good results with unimodal distributions as they are designed to work on multimodal histograms. For our unimodal histograms, the best and simpler technique is a variation of the corner method proposed by P. Rosin. It estimates the position of the corner of the curve of the histogram. A straight line is placed joining the main peak of the unimodal histogram and the last non-zero value of the histogram. The most distant point of the curve from the straight line is set as the threshold value.

This method fits the established criterion for threshold selection in the two first types of frames with minor changes. For the high activity frames, another thresholding technique is proposed. The used thresholding techniques are described in the following paragraphs:

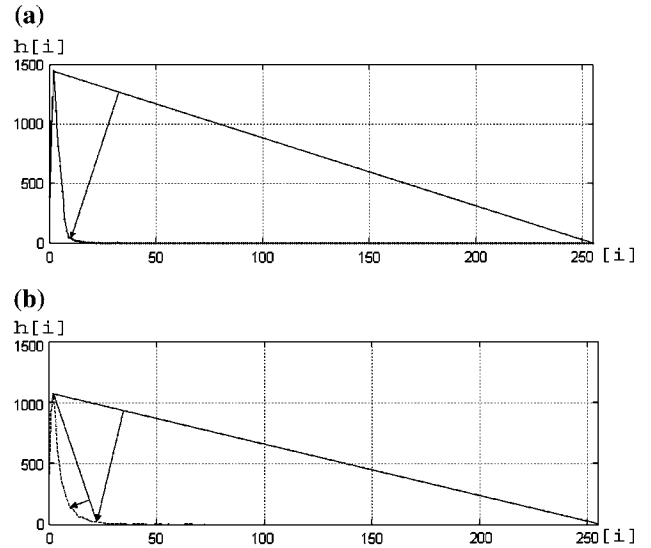
### 3.2.1 Low activity frames thresholding

In this case we employ the corner method, placing the straight line between the main peak and the last value, 255, instead of the last non-zero value of the histogram. Figure 9a shows an example of the corner method applied to a unimodal activity histogram: with this modification we ensure that most of the blocks will be set as non-active. Note that activity images in this case contain very low values and there are almost no relevant areas of activity.

Figure 5c shows the thresholded activity images for some example images. Black regions correspond to the activity image areas identified as non-active (activity values below the selected threshold), and white regions correspond to areas identified as active (activity values above the threshold). Particularly, Fig. 5c, first row, displays the thresholded activity image for *Mother & Daughter* in the case of a low activity frame; as expected,  $B_n$  is black, showing that all blocks have been classified as non-active.

### 3.2.2 Medium activity frames thresholding

The histograms of these frames contain more activity at the tail than the histograms of low activity frames. Direct use



**Fig. 9** **a** Refined corner method for histograms of low activity frames; **b** two iterations of the corner method for histograms belonging to medium activity frames

of the corner method would result in a classification of some blocks as non-active, blocks which should have been classified as active as their activity values are relevant but lay too close to the dominant histogram mode. To overcome this situation, a second iteration of the refined corner method is proposed to determine the threshold value: the corner found at the first iteration will work as final point for the second corner straight line, maintaining the main peak as initial point. Fig. 9b shows the scheme of two iterations of the corner method proposed. It is applied on a histogram corresponding to *Mother & Daughter* activity image presented in Fig. 5b second-row. In this example, the threshold was set to 23 in the first iteration, value which would discard relevant activity areas. However, the second iteration produces a lower value of the threshold, 10, which avoids discarding them. This can be observed in the *Mother & Daughter* thresholded activity image, presented in Fig. 5c, second row.

### 3.2.3 High activity frames thresholding

Finally, histograms corresponding to high activity frames contain more activity than the other types of histograms at the tail of the histogram. In this case two or more iterations of the corner method still show an excessively high threshold for these kind of active frames.

Therefore, we propose to combine the corner method with the use of another thresholding method that automatically selects a threshold based on the activity distribution. It is based on the classic Kittler and Illingworth method that



considers the analyzed histogram as the result of the mixture,  $f$ , of two Gaussian distributions:

$$f(i) = q_1 \cdot n_1(i) + q_2 \cdot n_2(i) \quad (9)$$

where  $q_1$  and  $q_2$  are the respective proportions, computed as:

$$q_1(t) = \sum_{i=0}^t P(i) \quad (10)$$

$$q_2(t) = \sum_{i=t+1}^l P(i) \quad (11)$$

where  $P(i)$  are the values of the histogram;  $n_1(i)$  and  $n_2(i)$  are normal distributions having  $(\mu_1, \sigma_1)$  and  $(\mu_2, \sigma_2)$  as the respective means and standard deviation, and  $t$  is the threshold that separates the distributions.

The final threshold is automatically determined as the  $t$  value that minimizes the Kullback distance between distributions as  $J$ , the directed convergence :

$$J = \sum_{i=0}^l P(i) \cdot \ln \left[ \frac{P(i)}{f(i)} \right]. \quad (12)$$

The process leads to the following expression that has to be minimized [15]:

$$F(t) = A - B(t) + C(t) \quad (13)$$

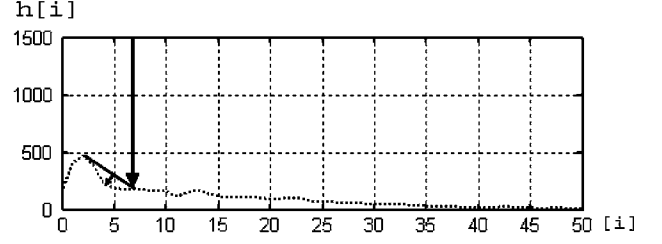
$$A = \frac{1 + \ln 2\pi}{2} \quad (14)$$

$$B(t) = q_1(t) \ln q_1(t) + q_2(t) \ln q_2(t) \quad (15)$$

$$C(t) = \frac{1}{2} (q_1(t) \ln \sigma_1^2(t) + q_2(t) \ln \sigma_2^2(t)). \quad (16)$$

Therefore, the value  $t$  that minimizes  $F(t)$  is chosen as the *Kullback* threshold.

It has been proved to be the best general thresholding method , working well with unimodal and multimodal distributions. In the case of our long tail histograms, the Kullback threshold is usually located on the left side of the long tail of the curve, although still leaving as non-active some relevant activity values. Therefore, to reduce the error of setting as non-active those relevant activity values, the Kullback method is combined with the corner method as follows: the Kullback threshold is first determined, and it is used as the final point for the corner method to obtain a lower threshold for any situation. As it is shown in the example of Fig. 10, Kullback threshold is seven. Even though it could be considered as a low threshold value, in the case of high activity frames, its direct application would imply quite a significant number of activity values to be considered as non-active. After applying the corner method with seven as the final point, a threshold value of four is obtained. This threshold is better adapted to the



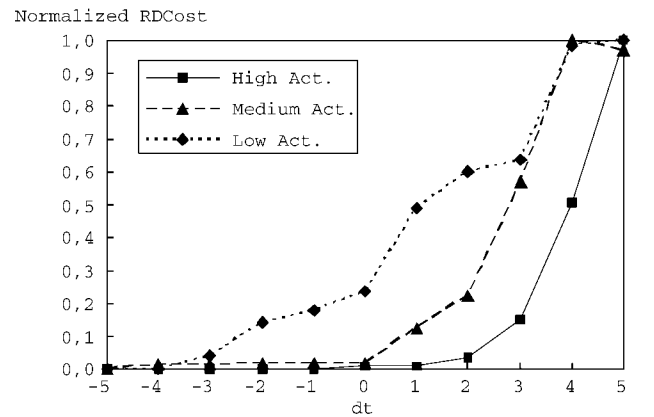
**Fig. 10** Kullback threshold used as final point for the corner method (for histograms of high activity frames)

histogram characteristics, keeping a large number of image areas as active, and providing excellent results as will be presented further in this paper.

The impact on the RD performance due to the selection of the threshold values is shown in Fig. 11. The normalized RD cost obtained for an example frame of the *Paris* sequence is depicted for different thresholds  $t_i$  different than the threshold  $t$  selected by the thresholding algorithms. It can be observed that higher threshold values ( $dt > 0$ ), although providing better reduction of computational cost, incur in high RD deviations (notice that for higher thresholds, more reduction of the computational cost is achieved). Analogously, lower thresholds values ( $dt < 0$ ), although providing slightly better RD performance, will not offer so significant reductions of computational cost.

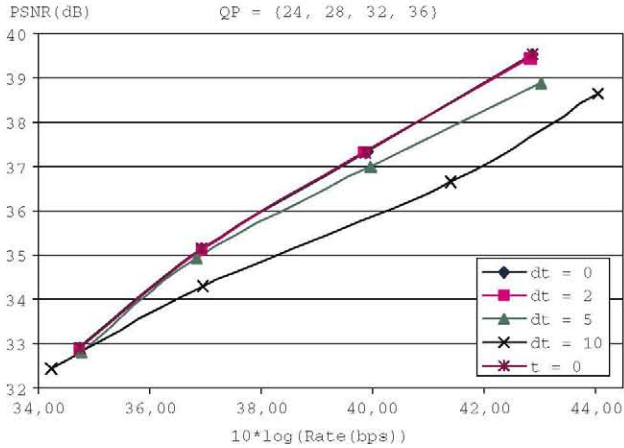
Additionally, Fig. 12 shows, for an example frame of the *Paris* sequence, the RD performance for different values of  $dt$ . In this case, a typical RD-curve is depicted, showing results of PSNR (dB) and Rate, in logarithmic units for QPs between 24 and 32.

As it can be observed, the curve corresponding to the proposed strategy (threshold “ $dt = 0$ ”) stays above all the curves corresponding to thresholds higher than “ $dt = 0$ ” ( $dt > 0$ ). On the other side, curves corresponding to lower



**Fig. 11** Normalized average RD cost obtained with  $t_i$  values different from the selected as best  $t$  value for the three different thresholding algorithms presented ( $t_i = t + dt$ )





**Fig. 12** Rate-distortion curve for an example frame from *Paris* sequence, showing RD performance for different  $dt$  values

thresholds ( $dt < 0$ , and being the limit  $t = 0$ ), do almost coincide with the RD curve provided by the proposed algorithm. Therefore, the proposed algorithm attempts to select the maximum threshold value which provides the closer RD curve to that corresponding to the full-search algorithm (without using any FMD strategy) while obtaining high reduction of the computational cost.

### 3.3 Multi-reference frames

Previous sections presented an adaptive thresholding set of techniques that perform the binarization of the activity image between frames. The result is a binarized activity image,  $B_n$ , whose pixels contain the activity information of each  $4 \times 4$  pixels block of the original frame  $F_n$ . As it has been described, each MB of  $F_n$  is then characterized by 16 activity values, each one corresponding to the activity value of each  $4 \times 4$  pixels block it contains. The thresholding process provides two types of blocks: *active* and *non-active*.

It may happen that some blocks that were classified as active because their MAE with respect to  $F_{n-1}^i$  exceeded the threshold, could obtain MAE values with respect other references below the threshold. In that case, the active block is switched to k-non-active block, where  $k$  specifies the number of the used reference frame. The binary activity image  $B_n$  becomes  $B_n^K$ , which contain *active* blocks, *non-active* and *k-non-active* (e.g., *2-non-active*, *3-non-active* and *4-non-active* if  $K = 4$ ).

The multi-reference feature is used to swap the maximum number of active blocks by k-non-active blocks: the more active blocks are, the less computational load reduction is achieved; whereas non-active and k-non-active lead to the selection of smaller sub-sets of modes (as it is

shown in Sect. 4) and then obtaining higher reduction of computational cost.

The main reason of doing this once the activity has been already analyzed and thresholded, apart from computational cost of the algorithms, is that it is much better to have a large number of non-active blocks rather than a mixture of non-active and k-non-active. This is because, as it is shown in Sect. 4, the more non-active blocks, the more use of SKIP-Mode, which is computed only taking into account the first reference frame, and is the most efficient prediction Mode when used in low activity MBs. If k-non-active blocks are considered from the beginning of the system, many non-active blocks would become k-non-active, and less MBs will be coded with only SKIP-Mode, resulting in a loss of computation cost saving.

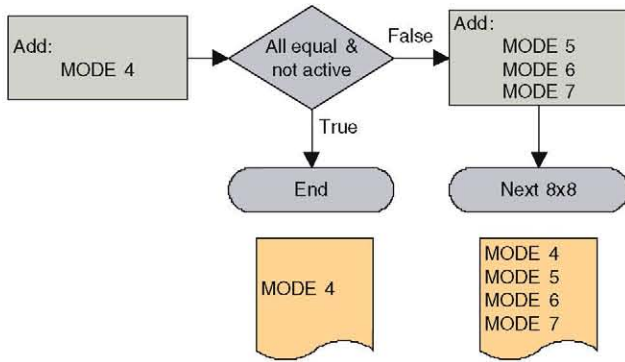
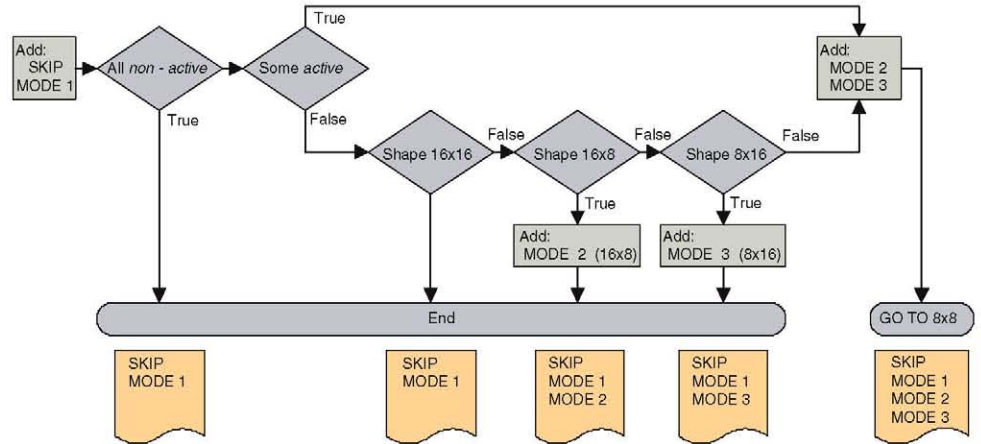
## 4 Selection of the sub-set of modes

The next step of the proposed scheme is the selection of an appropriate sub-set of inter-prediction modes to be computed for each MB, based on the activity information that it contains. This selection is done based on the analysis of the  $16 \times 4 \times 4$  blocks of each MB, classified as active, non-active and k-non-active.

The FMD algorithm, aims at achieving: (1) great computational load reduction at the ME stage and (2) little RD deviation compared with a complete analysis of modes. It exploits the fact that for low activity MBs, the best subdivision mode usually slightly differs from other subdivision modes in terms of RD cost. Therefore, we propose to limit the number of modes to be computed at the ME phase, with the consequent reduction of processing time, since the RD cost achieved with an appropriate selection of the restricted sub-set of modes is very close to the optimal RD cost.

Figure 13 shows a block scheme of the selection algorithm, which is described as a flow chart where inter-prediction modes are enabled or left as disabled. The enabled modes are those added to the sub-set of modes for that MB. There are paths on the flow chart that lead to an early termination, where selection ends and no more modes are enabled. In Fig. 13, the lists of enabled modes are shown for each early termination point. Fig. 14 shows the flow chart for each  $8 \times 8$  MB-partition, as these can be further sub-divided into sub-MB partitions. Therefore, for each MB, Fig. 13 shows the process of enabling modes of large blocks (SKIP-Mode, Mode 1, 2 and 3), while Fig. 14 represent the process for each  $8 \times 8$  MB partition, that enables or not the modes of small blocks (Mode 4, 5, 6 and 7). The complete scheme, that is applied to each single MB, is described as follows:

**Fig. 13** Mode decision block diagram for each MB. SKIP and Mode 1 are firstly added to the list of enabled modes. After that, several conditions are tested, resulting in the addition of Mode 2 or Mode 3 to the list. This list is depicted for each of the possible terminations of the algorithm



**Fig. 14** Block diagram of the selection of modes for each  $8 \times 8$  MB partition. The result is Mode 4, or Mode 4, 5, 6 and 7

- (1) *Initialization*: SKIP-Mode and Mode 1 are firstly enabled.
- (2) *Condition all non-active*: If all the 16 blocks inside the MB under analysis have been classified as non-active, no more modes are enabled, and the process ends with the list: SKIP-Mode and Mode 1. The distortion obtained with this selection of modes is controlled because all blocks were classified as non-active, so there is low activity between the original MB (belonging to  $F_n$ ) and the MB located in the previous frame ( $F'_{n-1}$ ) at the same position. Otherwise, the algorithm continues.
- (3) *Condition some active*: If there is not any active block at the MB, but there are non-active or some k-non-active (with  $k$  from 2 to  $K$ ), Mode 1, 2 and 3 are sequentially tested to find whether the distribution of these types of blocks inside the MB matches with the sub-division of Mode 1 ( $16 \times 16$ , all blocks are classified as the same type) Mode 2 ( $16 \times 8$ , the 8 upper-blocks are the same type, and the 8 bottom-blocks are another type) or Mode 3 ( $8 \times 16$ , the 8 left-blocks are the same type and the 8 right-blocks are another type). Table 2b, c show examples of this

distribution matching for Mode 2 and Mode 3, respectively, where black regions represent non-active blocks, and grey regions correspond to some k-non-active. These are early termination points if the conditions are verified, so the selection of the sub-set ends with the enabled modes.

- (4) Otherwise, there are at least one active block in the MB. Mode 2 and Mode 3 are enabled.
- (5) *Initialization*: Mode 4 is firstly enabled.
- (6) *Condition all equal and not active*: If their  $4 \times 4$  blocks are non-active or k-non-active, it is supposed that subdivision Modes 5, 6 and 7 will not offer RD cost reduction compared with Mode 4, so the algorithm reaches the early termination point for the current  $8 \times 8$  MB partition under analysis.
- (7) Only if the MB partition contains one or more active blocks, Modes 5, 6 and 7 are enabled. This is the end of the algorithm for the current  $8 \times 8$  MB partition.

The proposed algorithm offers a high range of possibilities of reducing the set of modes to be computed, but most of them are very similar: the result is that SKIP-Mode and Mode 1 and almost always Mode 2 and 3 are enabled, unless the specific early termination cases aforementioned are reached. For each  $8 \times 8$  MB partition Mode 4 is enabled, and Modes 5, 6 and 7 are only enabled if the 4 blocks of the  $8 \times 8$  MB partition are not of the same type. Table 2 shows examples of the sub-set of modes for some typical MBs: Table 2a shows the case with 16 non-active blocks belonging to the current MB; Table 2b, c are the cases where the different non-active and k-non-active blocks coincide with the sub-division form of Mode 2 and 3, respectively. Cases in Table 2d–h show MBs with different distributions of active, non-active and k-non-active.

As it will be shown in the results section, this selection algorithm ensures minimal RD deviation compared with the complete analysis of all inter prediction modes.



**Table 4** Rate-distortion deviation and computation reduction high QPs

Sequence	QP = 32, 36, 40, 44		
	APSNR	ABR	$\Delta T$ (%)
Foreman (CIF)	0.000	0.016	25.82
Coastguard (CIF)	-0.006	0.174	34.75
Tempete (CIF)	-0.007	0.170	33.42
M. & Daughter (QCIF)	-0.013	0.284	47.17
Container (CIF)	0.002	-0.036	48.67
Mobile (CIF)	0.000	0.006	8.23
News (QCIF)	-0.002	0.029	37.29
Silent (QCIF)	-0.012	0.062	50.30
Akiyo (QCIF)	-0.003	0.012	62.08
Paris (CIF)	-0.014	0.013	35.84
Claire (QCIF)	-0.028	0.143	56.48
Stefan (CIF)	-0.004	0.041	6.58
Average	-0.009	0.102	36.24

44). Low QP values are expected to produce larger bit rates and low distortions as the quantization step decrease, while high QP values correspond to low bit rates and high distortions. As it can be observed, the results depend on the activity level of each sequence. There are some sequences, like *Mother & Daughter* where the achieved  $\Delta T$  reaches 65.19%. These great results are due to that these sequences produce thresholded activity images with just a low percentage of the blocks classified as active. Therefore ME for many MBs will be quickly computed with SKIP-mode and Mode 1. High activity sequences like *Foreman*, *Coastguard* or *Tempete* obtain ME time reduction values between 26.13 and 35.88% depending on the set of QPs

**Table 5** Deviation of PSNR for different QPs (dB)

Sequence	QP							
	16	20	24	28	32	36	40	44
Foreman	-0.04	-0.04	-0.03	-0.01	-0.01	0.00	-0.01	0.00
Coastguard	-0.11	-0.10	-0.08	-0.05	-0.01	-0.01	0.00	0.00
Tempete	-0.04	-0.05	-0.06	-0.05	-0.04	-0.03	-0.01	-0.01
M. & Daughter	-0.09	-0.11	-0.13	-0.14	-0.08	-0.03	0.00	0.00
Container	-0.06	-0.05	-0.02	-0.02	-0.01	-0.01	0.00	0.00
Mobile	-0.01	0.00	0.00	0.00	0.00	0.00	-0.01	0.00
News	-0.03	-0.03	-0.03	-0.03	-0.03	0.00	0.00	0.00
Silent	-0.02	-0.02	-0.02	-0.03	-0.02	-0.01	0.00	0.00
Akiyo	-0.32	-0.20	-0.14	-0.19	-0.12	-0.10	-0.01	0.00
Paris	-0.01	-0.04	-0.03	-0.04	-0.04	-0.03	-0.01	0.00
Claire	-0.60	-0.59	-0.38	-0.26	-0.16	-0.07	-0.04	0.00
Stefan	-0.02	-0.03	-0.03	-0.04	0.00	-0.02	0.00	0.00
Average	-0.08	-0.07	-0.06	-0.05	-0.04	-0.02	-0.01	0.00

even when they are sequences with high amounts of activity. This is due to that even when there are not MBs with 16 non-active values, there will probably be a significant number of  $8 \times 8$  MB partitions with its four  $4 \times 4$  blocks set as non-active or K-non-active, consequently not-enabling Mode 5, 6 and 7.

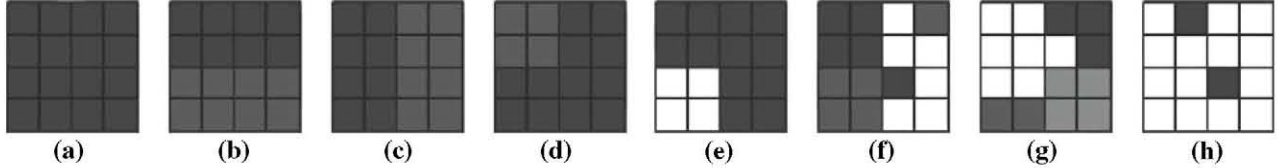
The RD deviations are very low, i.e., APSNR and ABR values are usually very low: 1.044% of ABR and -0.062 dB of APSNR for low QPs and 0.102% of ABR and -0.009 dB of APSNR for high QPs. These values clearly represent a negligible reduction of coding performance, thus keeping the high efficiency of the H.264/AVC encoder for real-time implementations which use our proposed FMD strategy.

Tables 5 and 6 show the specific values of PSNR and bitrate miss-rate obtained for each QP value, whereas Table 7 shows the specific values of  $\Delta T$  for each sequence and each QP value. It can be seen a decrement of the ME time reduction as the QP increases, whereas PSNR and bitrate miss-rate is very low in almost all cases. This is due to the distortion introduced into the reconstructed sequence. Simulations with QP values higher than 32 typically produce PSNRs values below 28 dB, which is a very poor quality. In these cases the activity image contains larger differences between the original  $F_n$  and the reconstructed  $F_{n-k}^i$ , mainly due to quantization noise rather than activity or movement inside the sequence. As the active area increases in these activity images, the ME time reduction achieved with our proposal is reduced to achieve a good RD performance.

Therefore, FMD is recommended for low QPs due to its low RD deviation and great ME time reduction, with the optimum results found at QP = 32, with an average value of -45.21% for all the sequences as it is observed in Table 7.

**Table 2** Mode decision for different MB examples

Mode	Skip	Modes 1,2,3			8x8 Up-Left		8x8 Up-Right		8x8 Down-Left		8x8 Down-Right	
		1	2	3	4	5,6,7	4	5,6,7	4	5,6,7	4	5,6,7
(a)	•	•	-	-	-	-	-	-	-	-	-	-
(b)	•	•	•	-	-	-	-	-	-	-	-	-
(c)	•	•	-	•	-	-	-	-	-	-	-	-
(d)	•	•	•	•	•	-	•	-	•	-	•	-
(e)	•	•	•	•	•	-	•	-	•	•	•	-
(f)	•	•	•	•	•	-	•	•	•	-	•	-
(g)	•	•	•	•	•	•	•	•	•	•	•	-
(h)	•	•	•	•	•	•	•	•	•	•	•	•



## 5 Results and discussion

The proposed algorithm has been implemented into the reference software model JM 12.1 to make experimental tests.

The proposed scheme of adaptive sub-set of modes for each MB according to the activity contained in it achieves the mentioned objectives as it will be shown in the obtained results of Rate, Distortion and ME Time saving, using typical standard test sequences.

Let note that the first requisite of the proposed scheme was to perform fast algorithms that do not introduce by themselves significant computational load. As it would be useless a ME time reduction if more computation is required at the FMD algorithm, we have always selected the alternatives of analysis that perform faster results. Tests have demonstrated the algorithm computational cost is lower than 1% of ME time, so that we consider it negligible for further result comparisons.

Baseline profile has been used for the tests (only I and P slices), with search range  $16 \times 16$  over five reference frames and the fast ME algorithm enabled

. Tests have been made for a wide range of QP values following the common test conditions and the method of calculation of average PSNR differences between RD-curves proposed by Bjöntegeard. This is an interpolation method to build PSNR-Rate curves to compare results of the reference encoder and the proposed scheme of encoder with the FMD algorithm. The results are *APSNR* (dB), *ABR* (%) and  $\Delta T$ (%): *APSNR* is the Average PSNR difference in dB obtained between the RD curves of the reference encoder and the encoder with our proposed FMD strategy; *ABR* is the Average Bit Rate deviation in percentage values. These average differences are the differences between the integrals of the RD curves divided by the integration interval; and  $\Delta T$  means the ME time

reduction in percentage compared with the reference encoder. It is calculated as the average value of the ME time reductions achieved with each QP value. These percentage values are obtained as:

$$\Delta T(\%) = \frac{T_{\text{reference}} - T_{\text{FMD}}}{T_{\text{reference}}} \times 100 \quad (17)$$

where  $T_{\text{reference}}$  is the ME time obtained with the reference encoder, and  $T_{\text{FMD}}$  is the ME time with the encoder using the FMD algorithm proposed in this work. Negative values for *APSNR* mean that less PSNR value is achieved with the proposed FMD scheme compared with the reference simulation and for *ABR*, negative values mean lower bit rate.

Tables 3 and 4 show these results for two sets of QP values: low-QPs (16, 20, 24, 28) and high-QPs (32, 36, 40,

**Table 3** Rate-distortion deviation and computation reduction low QPs

Sequence	QP =16, 20, 24, 28		
	APSNR	ABR	$\Delta T$ (%)
Foreman (CIF)	-0.039	0.882	26.13
Coastguard (CIF)	-0.105	1.523	35.88
Tempete (CIF)	-0.069	1.013	28.43
M. & Daughter (QCIF)	-0.199	3.814	65.19
Container (CIF)	-0.029	0.622	58.76
Mobile (CIF)	-0.069	0.695	3.70
News (QCIF)	-0.029	0.401	58.88
Silent (QCIF)	-0.017	0.247	47.03
Akiyo (QCIF)	-0.023	0.312	64.71
Paris (CIF)	-0.026	0.543	29.54
Claire (QCIF)	-0.108	2.143	67.97
Stefan (CIF)	-0.034	0.342	5.78
Average	-0.062	1.044	41.00

**Table 6** Deviation of bitrate for different QPs (%)

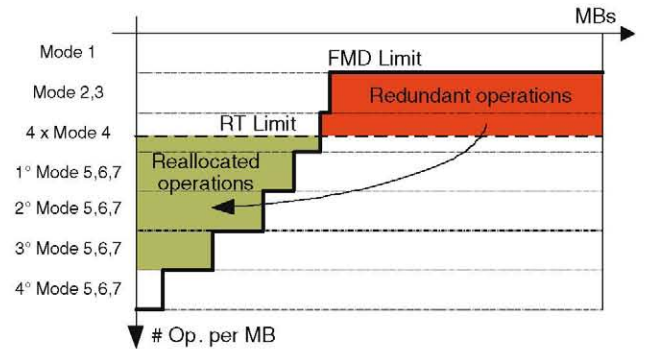
Sequence	QP							
	16	20	24	28	32	36	40	44
Foreman	0.52	0.18	-0.02	-0.14	-0.06	-0.05	0.00	0.00
Coastguard	0.31	0.32	0.25	0.19	-0.03	-0.02	0.01	0.00
Tempete	0.25	0.32	0.30	0.13	-0.12	-0.09	-0.03	0.01
M. & Daughter	2.08	2.34	1.08	-0.55	-0.32	-0.14	0.00	0.00
Container	-0.14	-0.48	-0.25	-0.16	-0.02	-0.02	0.00	0.00
Mobile	0.03	0.01	0.02	0.01	0.00	0.00	-0.01	-0.01
News	0.02	0.00	0.00	-0.01	-0.02	0.00	0.00	0.00
Silent	0.01	0.00	-0.01	0.00	-0.02	0.00	-0.01	0.00
Akiyo	-0.38	-1.67	-1.60	-2.69	-2.02	-2.47	-0.22	0.00
Paris	0.73	0.16	0.07	-0.12	0.00	-0.32	0.00	0.00
Claire	2.19	-0.57	-3.35	-1.75	-2.90	-1.42	-1.21	0.00
Stefan	0.22	-0.15	-0.06	0.00	0.00	-0.61	0.00	0.00
Average	0.65	0.43	0.27	-0.14	-0.36	-0.39	-0.16	0.00

**Table 7** Motion estimation time reduction (%)

Sequence	QP							
	16	20	24	28	32	36	40	44
Foreman	25.42	26.06	26.10	26.80	28.68	24.54	24.88	25.19
Coastguard	35.26	37.51	36.98	33.78	33.29	36.32	35.30	34.09
Tempete	22.75	27.37	30.79	32.82	33.85	32.85	34.24	32.75
M. & Daughter	61.55	65.32	65.97	67.93	65.75	58.06	38.19	26.66
Container	55.97	56.83	57.69	64.53	69.48	60.98	29.51	34.70
Mobile	4.03	3.46	4.27	3.02	4.27	5.45	9.91	13.29
News	58.18	54.81	57.94	64.58	60.42	45.95	26.19	16.58
Silent	43.09	39.58	49.26	56.19	59.63	55.60	47.62	38.36
Akiyo	59.21	61.42	67.01	67.19	69.83	71.12	58.14	49.23
Paris	24.36	28.46	29.22	36.14	39.25	38.97	36.72	28.42
Claire	66.21	69.67	68.10	67.91	68.97	65.41	55.14	36.42
Stefan	5.53	5.23	6.17	6.21	9.12	6.61	5.45	5.17
Average	38.47	39.63	41.62	43.92	45.21	41.82	33.44	28.40

Additionally, a little variation of the algorithm provides even better results in terms of  $\Delta T$  by paying a bit more deviation in bitrate and distortion. It is achieved by not computing Mode 1 at the first step of the algorithm shown in Fig. 13. This way, for those MBs with 16 non-active values, only the SKIP-Mode is computed, achieving great reductions of computational cost (as no RD cost has to be computed). In average, this slight change increases the reduction  $\Delta T$  achieved by the proposed algorithm in 10%, while increasing the ABR to 1.320% and APSNR to 0.401 dB.

Another important use of this FMD strategy is as an intelligent tool for real-time applications with constrained

**Fig. 15** Reallocation of computational effort based on FMD results given a real-time restriction

number of operations between frames, which is depicted in Fig. 15.

Typically, the easiest solution to fit an encoder like H.264 into a real-time application is to limit the number of operations to be done at different levels, including the ME phase. Without any a priori knowledge, the only way to proceed is to limit the number of Modes to be computed for all MBs. This fixed limit is shown in Fig. 15 as the *RT Limit*.

However, when using the proposed FMD strategy, the existing redundancy of Modes for each MB is known, for any type of sequence, with low or high activity, previously to the ME phase. Therefore, it is easy to define an adaptive limit, which is shown in Fig. 15 as the *FMD Limit*, where the redundant operations are identified as those to be devoted to MBs where the RT Limit offers more operations than the MBs really need to provide good RD performance.

Therefore, it is possible to reallocate this amount of operations to MBs that need it to provide better RD

performance than with the fixed RT Limit, resulting in a wise strategy to maximize RD performance given real-time constraints.

## 6 Conclusions

In this paper we have introduced a new FMD strategy that dramatically reduces the computational cost of the ME phase of H.264 Baseline Profile (only P-Slices are used) by wisely selecting an appropriate sub-set of inter-prediction Modes to be computed for each MB. This selection is based on a detailed analysis of the activity level of each frame, which leads to the identification of the areas of the image where more coding effort has to be devoted to achieve high performance in terms of RD.

Real-time encoder implementations using this strategy will benefit from the identification of those areas obtaining better RD results. This improved coding efficiency is achieved by wisely distributing the available number of operations among the MBs according to their activity level.

**Acknowledgments** This work has been partially supported by the Ministerio de Educación y Ciencia of the Spanish Government under project TIN2004-07860 (Medusa) and by the Comunidad de Madrid under project S0505/TIC-0223 (Pro-Multidis).

## References

- Wiegand, T., Sullivan, G.J.: Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 560–576 (2003)
- Richardson, I.E.G.: H.264 and MPEG-4 Video Compression. Video Coding for Next-generation Multimedia, edn. Wiley, London (2003)
- Sullivan, G.J., Wiegand, T.: Rate-distortion optimization for video compression. *IEEE Signal Process. Mag.* **15**(6), 74–90 (1998)
- Wiegand, T., Girod B.: Lagrange Multiplier Selection in Hybrid Video Coder Control. In: *IEEE Proceedings ICIP01*, vol. III, pp. 542–545 (2001)
- Ates, H.F., Kaneberoglu, B., Altunbasak, Y.: Rate-Distortion and complexity joint optimization for fast motion estimation in H.264 video coding. In: *IEEE Proceedings ICIP06*, pp. 37–40, Atlanta, 8–11 October 2006
- Choi, I., Lee, J., Jeon, B.: Efficient Coding Mode Decision in MPEG-4 Part-10 AVC/H.264 Main Profile. In: *IEEE Proceedings ICIP04*, pp. 1141–1144 (2004)
- Choi, B.-D., Nam, J.-H., Hwang, M.-C., Ko, S.-J.: Fast motion estimation and Inter-mode Selection for H.264. *EURASIP J. Appl. Signal Process.* vol. 2006, pp. 1–8
- Nieto, M., Salgado, L., Cabrera, J.: Fast Mode Decision on H.264/AVC Main Profile Encoding Based on PSNR Predictions. In: *IEEE Proceedings ICIP06*, pp. 49–52, Atlanta, 8–11 October 2006
- Lin, Z., Yu, H., Pan, F.: A scalable fast mode decision algorithm for H.264. In: *IEEE International Symposium on Circuits and Systems 2006*, 21–24 May 2006

- Kim, C., Jay Kuo, C.-C.: Feature-based intra-/intercoding mode selection for H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.* **17**(4), 441–453 (2007)
- Weisstein, E.W.: Heaviside Step Function. From MathWorld. A Wolfram Web Resource: <http://mathworld.wolfram.com/HeavisideStepFunction.html>
- Niemisto, A.: A Comparison of Non-Parametric Histogram-Based Thresholding Algorithms: Presentation for 8002202 Digital Image Processing, vol. III, 27 October 2004
- Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. *J. Electron. Imaging* **13**(1), 146–165
- Rosin, P.: Thresholding for change detection. In: *Proceedings ICCV98*, pp. 274–279 (1998)
- Haralick, R.M., Shapiro, L.G.: *Computer and Robot Vision*, edn. Addison-Wesley Longman Publishing Co., Inc., Reading (1992)
- H.264/AVC Reference Software Model (JM12.1). <http://bs.hhi.de/suehring/tml/index.html>
- Sullivan, G.J.: Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-Resolution Progressive-Scan Source Material. Document ITU-T VCEG-N81, Santa Barbara, 24–27 September 2001
- Bjontegaard, G.: Calculation of Average PSNR Differences between RD-curves. Document ITU-T VCEG-M33, April 2001