An HVS-based Adaptive Computational Complexity Reduction Scheme for H.264/AVC Video Encoder using Prognostic Early Mode Exclusion

Muhammad Shafique, Bastian Molkenthin, and Jörg Henkel
Karlsruhe Institute of Technology, Chair for Embedded Systems, Karlsruhe, Germany
{shafique, molkenth, henkel} @ informatik.uni-karlsruhe.de

Abstract—**The H.264/AVC video encoder standard significantly improves the compression efficiency by using variable block-sized Inter (P) and Intra (I) Macroblock (MB) coding modes. In this paper, we propose a novel Human Visual System based Adaptive Computational Complexity Reduction Scheme (ACCoReS). It performs Prognostic Early Mode Exclusion and a Hierarchical Fast Mode Prediction to exclude as many I-MB and P-MB coding modes as possible (up to 73%) even before the actual Rate Distortion Optimized Mode Decision (RDO-MD) and Motion Estimation while keeping a good quality. In the best case, ACCoReS processes exactly one MB Type and one corresponding near-optimal coding mode, such that the complete RDO-MD process is skipped. Experimental results show that compared to state-of-the-art approaches ([10], [22]-[26]), ACCoReS achieves a speedup of up to 9.14x (average 3x) with an average PSNR loss of 0.66 dB. Compared to exhaustive RDO-MD, our ACCoReS provides a performance improvement of up to 19x (average 10x) for an average 3% PSNR loss.**

I. Introduction

The advanced video coding standard H.264/AVC [1] was developed by the Joint Video Team (JVT) to provide a bit rate reduction of 50% as compared to MPEG-2 with similar subjective visual quality [7]. However, this improvement comes at the cost of significantly increased computational complexity [8], thus posing serious challenges on high-throughput (real-time) encoder implementations. High computational complexity of H.264 is mainly due to its complex *Prediction*, *Motion Estimation* (ME) and *Rate Distortion Optimized Mode Decision* (RDO-MD) processes that operate on multiple (variable) block sizes (as shown in Fig. 1). Large effort has been made in developing fast algorithms in ME for H.264 to reduce its complexity [15], [21]. However, RDO-MD is the most critical functional block in H.264, as it determines the number of ME iterations. Therefore, it becomes the primary research focus for complexity reduction.

A *Macroblock* (MB, i.e. 16x16 pixels) in a video frame can be divided into 16x16, 16x8, 8x16, or 8x8 blocks. Each 8x8 block can be further divided into 8x4, 4x8, or 4x4 sub-blocks. Altogether, there are 7 different block types. An MB can be encoded using one of the following two *MB Types*(see Fig. 1):

a) ***Intra-Predicted*** **(I-MB):** prediction is performed using the reconstructed pixels of the neighboring MBs in the current frame.
b) ***Inter-Predicted*** **(P-MB):** prediction is performed using the reconstructed pixels of the MBs in the previous frame. In this case ME has to be performed for various block size combinations (altogether 20 different ME combinations per MB are evaluated in RDO-MD [11]). An example scenario is presented in Fig. 1.

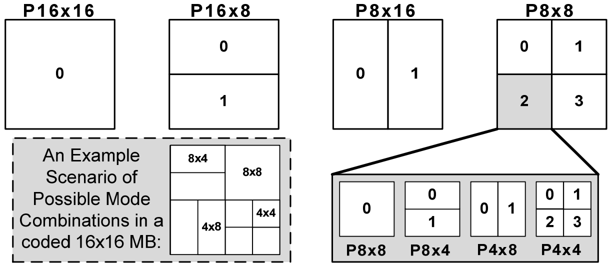

P16x16
0
P16x8
0
1
P8x16
0
1
P8x8
0
1
2
3
An Example Scenario of Possible Mode Combinations in a coded 16x16 MB:
8x4
8x8
4x8
4x4
0
P8x8
0
1
P8x4
0
1
P4x8
0
1
2
3
P4x4

Fig. 1: Variable Block Sizes for Inter Prediction used in H.264/AVC

Each *MB Type* can be predicted using one of the following coding modes[1] with variable block sizes.

$$Mode_P \in \{SKIP, P16\times16, P16\times8, P8\times16, P8\times8, P8\times4, P4\times8, P4\times4\}$$

$$Mode_I \in \{I16\times16, I8\times8, I4\times4\}$$

RDO-MD in H.264 processes all possible P-MB and I-MB mode combinations in all possible block sizes. It employs a *Lagrange-based cost function* that minimizes the *Distortion* (D) for a given *Rate* (R), as given below:

$$J(c,r,Mode|QP) = D(c,r,Mode|QP) + \lambda_{Mode} * R(c,r,Mode|QP)$$

'R' is the number of bits required to code the *'Mode'* and *'D'* is computed using *Sum of Absolute Transformed Differences* (SATD) or *Sum of Absolute Differences* (SAD) with respect to the current *'c'* and the reconstructed *'r'* MBs. λ is the Quantization Parameter (QP)-based *Lagrange Multiplier*, such that: $\lambda=0.85*2^{(QP-12)/3}$. The mode that provides the *best prediction* (i.e. minimizes the above-mentioned cost function) is chosen as the final coding mode. This process is called *exhaustive* RDO-MD. However, the *exhaustive* RDO-MD process is extremely compute-intensive (all of the coding modes for an MB are investigated before a decision about the actual coding mode is made), thus practically infeasible in real-world performance and/or power-critical embedded systems. Note, RDO-MD execution for P-MB modes is far more complex than that for I-MB modes due to the compute-intensive ME process. This fact becomes critical when after the RDO-MD the final coding mode comes out to be an I-MB mode, thus in this case the complete ME comes out to be unnecessary. To address the limitations of *exhaustive* RDO-MD, fast RDO-MD schemes are employed.

The basic idea of fast RDO-MD scheme is to select a set of coding mode candidates (which is much smaller than the set of all modes) such that the computational requirements of the RDO-MD process are significantly reduced while keeping the visual quality close to that of *exhaustive* RDO-MD. Several efforts have been made to reduce the computational complexity of H.264 by using various fast RDO-MD algorithms, such as fast P-MB MD [10]-[11], [19]-[26], fast *SKIP* MD [18], fast I-MB MD [12]-[13], and the combination of the above [18], [20]. However, most of the state-of-the-art approaches ([10]-[14], [18]-[26]) deploy a similar philosophy as they sequentially process mode by mode and exclude the modes depending upon the output of previously evaluated modes i.e., modes are not excluded in the fast RDO-MD until some ME is not done. Therefore, these approaches suffer from a limitation that – in worst case – all possible coding modes are evaluated. In average case, still significant (more than half of all) modes are computed or even in the best case at least one mode from both P-MB and I-MB is processed (see [10]-[11]). In any case, ME is always processed, thus the computational requirements of the state-of-the-art are still far too high, which makes them infeasible for embedded systems. **This begets the need for a *Computational Complexity Reduction Scheme*** that can adaptively exclude as many coding modes as possible from the candidate mode set at run time even before starting the actual fast RDO-MD.

A. Our Novel Contribution:

In this paper, we propose *a novel Adaptive Computational Complexity Reduction Scheme (ACCoReS)* for H.264 encoder that performs a ***Prognostic Early Mode Exclusion***, which curtails the set of possible coding modes, based on the properties of the *Human Visual System* (HVS). It incorporates an *HVS-based MB Categorization* by exploiting different video frame statistics, motion-field statistics, and modes of previously encoded neighboring MBs. The QP-based thresholds are used for MB categorization and are modeled & formulated on MATLAB using polynomial curve fitting. *Prognostic Early Mode Exclusion* is followed by a ***Hierarchical Fast Mode Prediction*** that further reduces the curtailed set of coding modes using the spatial and temporal statistics of video sequence. At the last stage, a ***Sequential***

[1] In this paper, we only target *Baseline* and *Main* profiles, therefore, we do not consider I8x8. However, the decisions and steps for I4x4 in our proposed scheme are scalable for I8x8.

978-3-9810801-6-2/DATE10 © 2010 EDAA

RDO Mode Elimination – as a fast RDO-MD – is used that eradicates the unlikely coding modes depending upon the output of previously processed coding mode. Note: *ACCoReS* facilitates the integration of previously researched fast RDO-MD schemes (e.g. [10]-[14], [18]-[26]) in the stage of *Sequential RDO Mode Elimination*.

The principal distinctions of our proposed *ACCoReS* compared to the state-of-the-art approaches are the *Prognostic Early Mode Exclusion* and the *Hierarchical Fast Mode Prediction* that **exclude more than 70% of the possible coding modes even before starting the fast RDO-MD and ME** while keeping the bit rate and distortion loss imperceptible (see Section VI). The ultimate goal of our *ACCoReS* is to predict exactly one *MB Type* and one corresponding near-optimal mode in the best case, thus skipping the complete RDO-MD and ME. Therefore, the computational load is significantly decreased. Up to the best of our knowledge, none of the state-of-the-art has done it before.

Our results show that the proposed *ACCoReS* results in a significant performance improvement (up to 19x for QCIF and 14.5x for CIF videos). This benefit comes at the cost of an average 3% PSNR loss and insignificant (compared to the benefit) processing overhead of computing video frame statistics (as we will see in Section D).

Paper Organization: Section II presents the related work. Section III presents a case study for video analysis considering the HVS properties. Section IV presents our *ACCoReS* in detail followed by the QP-based thresholding in Section V. Section VI presents the results and evaluation followed by conclusion in Section VII.

II. Related work

The fast RDO-MD algorithms either simplify the used cost function or reduce the set of candidate modes iteratively depending upon the output of the previous mode computation. [10] uses *Mean Absolute Difference* (MAD) of MB to reduce the number of candidate block types in ME. On average, it processes 5 out of 7 block types. [23] uses the RD-cost of neighboring MBs to predict the possible coding mode for the current MB. Similar approach is targeted by [24] and [26] that use the residue texture or residue of current and previously reconstructed MB for fast P-MB RDO-MD. [25] uses the mode information from previous frame to predict the modes of MBs in the current frame. [22] provides a fast SKIP and P16x16 prediction as an early predicted mode option. In [11], smoothness and SAD of the current MB are exploited to extend the *Skip* prediction and exclusion of smaller block mode types. Even if all conditions are satisfied, still 152 out of 168 RD costs are evaluated (*Luminance* component only), else all RD costs are evaluated as the *exhaustive* RDO-MD.

[12] exploits the local edge information by creating an edge map and an edge histogram for fast I-MB RDO-MD. Using this information, only a part of available I-MB modes are chosen for RDO, more precisely 4 instead of 9 I4x4 and 2 out of the 4 I16x16 are processed. The fast I-MB MD scheme in [13] uses partial computation of the cost function and selective computation of highly probable modes. I4x4 blocks are down-sampled and the predicted cost is compared to variable thresholds to choose the most probable mode.

A limited work has been done that jointly performs fast MD for both I-MB and P-MB. In [14], a scalable mode search algorithm is developed where the complexity is adapted jointly by parameters that determine the aggressiveness of an early stop criteria, the number of re-ordered modes searched, and the accuracy of ME steps for the P-MB modes. At the highest complexity point, all P-MB and I-MB modes are processed with highest ME accuracy. [17] proposes a scalable fast RDO-MD for H.264 that uses the probability distribution of the coded modes. It prioritizes the MB coding modes such that the highly probable modes are tried first, followed by less probable ones.

Unlike the related work, **our scheme performs** an extensive mode-exclusion before fast RDO-MD and ME thus providing a significant reduction in the computational complexity. Our proposed scheme in most of the cases (up to 70%) skips the complete RDO process and predicts the near-optimal coding mode and *MB Type* (see Section VI) that up to the best of our knowledge, related work have not achieved yet. We will now present the analytical case study for finding the important spatial and temporal video statistics as used by *ACCoReS*.

III. Analytical Case Study of video sequences

Although the digital image and video processing fields are built on a foundation of mathematical and probabilistic formulations, human intuition and analysis play the central role in the choice of one technique vs. another [3]. Therefore, important properties of the *Human Visual System* (HVS) are considered in the scope of work to account for subjective quality. Some important HVS properties are as follows (see [3], [4], and [5] for details):

a) The perceived brightness is a function of contrast and light intensity. Visual system tends to overshoot and undershoot at the boundary of regions of different intensities.
b) The total range of distinct intensity levels that an eye can discriminate simultaneously is rather small when compared with the total adaptation range. Below that level, all stimuli are perceived as indistinguishable blacks.
c) The Human eye is more sensitive to brightness compared to color.
d) At low levels of illumination, vision is carried out by activity of the *Rods* (part of the human eye; they are not involved in color vision), therefore, under low ambient light human eye can only extract the luminance information.
e) Moving objects capture more attention of the eye compared to the stationary objects.

We have carried out an extensive investigation of several video sequences [9] to subjectively learn the HVS response to different statistics of video frames and their corresponding coding modes. Fig. 2 shows the coding mode distribution (P-MBs in green and I-MBs in purple) for the 7th frame of *American Football* sequence encoded with *exhaustive* RDO-MD using JM13.2 software [2]. This analysis revealed that MBs with high texture and fast motion (e.g. fast moving players) are more probable to be encoded as I4x4 or P8x8 and blow. On the contrary, homogeneous or low-textured MBs with slow motion (e.g. grassy area) are more probable to be encoded as P16x16, P16x8, or P8x16 because the *Motion Estimation* (ME) has high probability to find a good match. Similar behavior was found in various other video sequences leading to the conclusion that majority of coding modes of a video frame can truly be predicted using spatial and temporal statistics of the current and previous video frames.

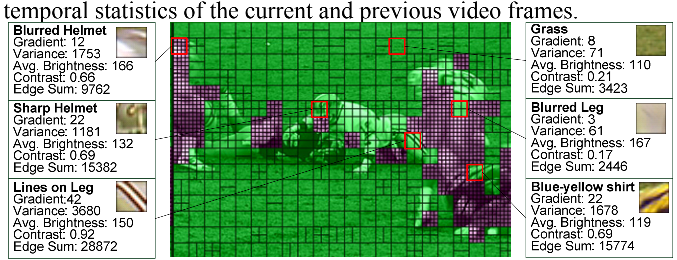


Fig. 2: Mode Distribution & Video Statistics in the 7th Frame of American Football

From our analytical study, we have learnt that five primitive characteristics of a video frame are sufficient to categorize an MB, thus to predict a probably correct coding mode. The decision of *which video frame property to choose* can be made considering the tradeoff between computational overhead and the provided precision in the early mode-prediction.

a) **Average Brightness** is used to categorize an MB as *dark* or *bright*. It is the average of luminance values *I(i,j)* of an MB (Eq. 1).

$$\mu_{MB} = \left(\sum_{i=0}^{15}\sum_{j=0}^{15}\left(I(i,j)\right)+128\right)>>8 \qquad \text{(Eq. 1)}$$

b) **Contrast** is the difference in visual properties that makes an object distinguishable from the background and other objects. In our scheme – due to its simplicity – we have used a modified version of *Michelson Contrast* [6] as shown in Eq. 2.

$$C_{MB} = \left[\max_{0<(i,j)<16} I(i,j) - \min_{0<(i,j)<16} I(i,j)\right]>>8 \qquad \text{(Eq. 2)}$$

c) **Variance** is a measurement for statistical dispersion (Eq. 3), thus it is used as descriptor of smoothness or measurement of texture. If all samples have the same brightness, then it is a flat/smooth area and the corresponding *Variance* is zero.

$$\sigma_{MB}^{2} = \sum_{i=0}^{15}\sum_{j=0}^{15}\left(I(i,j)-\mu_{MB}\right)^{2} \qquad \text{(Eq. 3)}$$

d) **Gradient:** *Gradient* is defined as the rate of change of luminance. In our case, it measures the average rate of change of luminance over a whole 16x16 MB, vertically (Gx) and horizontally (Gy). Therefore, it is regarded as an approximation of texture. The first order *Gradient* (G) along a particular direction is approximated by using the difference between two pixel along that direction (Eq. 4).

$$G_x = (\sum_{i=0}^{15}\sum_{j=0}^{15}\left|{}^{\partial f}\!/\!{}_{\partial x}\right| + 128) >> 8\ , \quad {}^{\partial f}\!/\!{}_{\partial x} = I(i,\ j) - I(i-1,\ j)$$

$$G_y = (\sum_{i=0}^{15}\sum_{j=0}^{15}\left|{}^{\partial f}\!/\!{}_{\partial y}\right| + 128) >> 8, \quad {}^{\partial f}\!/\!{}_{\partial y} = I(i,\ j) - I(i,\ j-1) \qquad \text{(Eq. 4)}$$

$$G = (|G_x| + |G_y| + 1)\ /\ 2$$

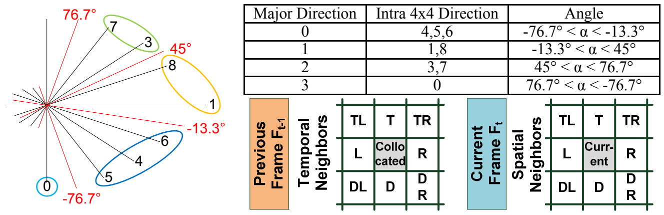
e) **Texture and Edges:** In addition to *Gradient*, a more precise edge detection – operating on a finer granularity – is required to predict the smaller coding modes more precisely. A *Sobel Edge Filter* is applied to obtain the magnitude and the direction of edges for every 4x4 sub-block. The *Sobel Edge Filter* has the advantage of providing both differencing and smoothing effect. The total edge values for a 4x4 sub-block, 8x8 block, and 16x16 MB are computed using Eq.5.

$$S_x = (\sum_{i=0}^{3}\sum_{j=0}^{3}\begin{pmatrix} I(i+1,\ j-1) + 2I(i+1,\ j) + I(i+1,\ j+1) \\ -I(i-1,\ j-1) - 2I(i-1,\ j) - I(i-1,\ j+1) \end{pmatrix}$$

$$S_y = (\sum_{i=0}^{3}\sum_{j=0}^{3}\begin{pmatrix} I(i-1,\ j+1) + 2I(i,\ j+1) + I(i+1,\ j+1) \\ -I(i-1,\ j-1) - 2I(i,\ j-1) - I(i+1,\ j-1) \end{pmatrix} \qquad \text{(Eq. 5)}$$

$$S_{4x4} = |S_x| + |S_y|; \quad S_{8x8} = \sum_{k=0}^{3} S_{4x4}^{k}; \quad S_{16x16} = \sum_{k=0}^{3} S_{8x8}^{k}$$

The direction angle (in degrees) with respect to the x-axis is calculated as α_{4x4}=(180°/π)*$\tan^{-1}(G_y/G_x)$. It is used to classify an edge into one of the following four directional groups (Fig. 3).



Major Direction	Intra 4x4 Direction	Angle
0	4,5,6	-76.7° < α < -13.3°
1	1,8	-13.3° < α < 45°
2	3,7	45° < α < 76.7°
3	0	76.7° < α < -76.7°

Fig. 3: Directional Groups with respect to the Edge Direction Angle and Notion of Spatial and Temporal Neighboring Macroblocks

In addition to the spatial statistics of video sequences, we have considered the temporal statistics (i.e., *SAD, Motion Vector-MV, and Coding Modes of the spatial and temporal neighboring MBs,* see the notion of Fig. 3) to corroborate the early prediction decision.

IV. Our Adaptive Computational Complexity Reduction Scheme (ACCoReS)

Fig. 4 presents the overview of our proposed *ACCoReS*. The step-by-step procedure is given below:

Step-1: First, the *HVS-based categorization* of MBs is performed using the spatial and temporal video statistics. The *QP-based thresholds* (as discussed in Section V) are used for this categorization.

Step-2: Afterwards, a *Prognostic Early Mode Exclusion* for I-MB and P-MB coding modes is incorporated that excludes the highly unlikely modes. In many cases the curtailed set of modes is left with either I-MB or P-MB modes, especially for low-motion sequences.

Step-3: *Hierarchical Fast Mode Prediction* further analyzes this curtailed set of modes and provides a set of candidate coding modes.

Step-4: In the last step, *Sequential RDO Mode Elimination* is done. It processes the candidate coding modes one-by-one starting from the bigger partitions. After a mode is processed, it is evaluated for the termination condition or to exclude further irrelevant modes. Now we will explain each processing stage of *ACCoReS* in detail.

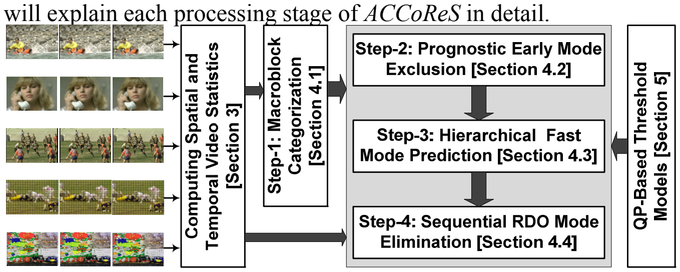

Computing Spatial and Temporal Video Statistics [Section 3]
Step-1: Macroblock Categorization [Section 4.1]
Step-2: Prognostic Early Mode Exclusion [Section 4.2]
Step-3: Hierarchical Fast Mode Prediction [Section 4.3]
Step-4: Sequential RDO Mode Elimination [Section 4.4]
QP-Based Threshold Models [Section 5]

Fig. 4: Our Adaptive Computational Complexity Reduction Scheme (ACCoReS)

A. Step-1: HVS-Based Macroblock Categorization

Video Frame Statistics based Categorization: Depending upon their spatial statistics, MBs can fall in one or many of the following categories:

Average Brightness (μ_{MB})	very dark (μ_{VD}), dark (μ_D), bright (μ_B), very bright (μ_{VB})
Contrast (C_{MB})	low (C_L), high (C_H) contrast
Variance (σ^2_{MB})	very low (V_{VL}), low (V_L), high (V_H) variance
Gradient (G_{MB})	very low (G_{VL}), low (G_L), high (G_H) gradient
Edge (S_{MB})	low (S_L), highly (S_H) edged

Combinations of the above-defined categories are used to predict the MB content characteristics as follows:

$$\begin{aligned} High_{MB}^{Textured} &= (S_H \,\&\, V_H) \| (S_H \,\&\, G_H) \| (G_H \,\&\, V_H) \\ S_{MB}^{StrongThick} &= !V_H \,\&\, S_H \,\&\, \mu_B \,\&\, G_H \\ S_{MB}^{StrongThin} &= !\,\mu_B \,\&\, G_H \,\&\, V_H \,\&\, (!\,\mu_D) \\ S_{MB}^{ManyThin} &= S_H \,\&\, \mu_B \,\&\, G_H \,\&\, V_H \end{aligned} \qquad \text{(Eq. 6)}$$

Directional Statistics: An edge direction is called *dominant* if the edge sum belonging to an edge direction group *'i'* (see Fig. 3) significantly contributes to the total edge sum of this MB.

$$EDir_{MB}^{Dominant} = \begin{cases} 1, & S_i > \varepsilon * S_{MB};\ \ i \in \{0, 1, 2, 3\} \\ 0, & Otherwise \end{cases} \qquad \text{(Eq. 7)}$$

$$EDir_{MB}^{Vt} = S_3 > 0.5 * S_0 \qquad EDir_{MB}^{Hz} = S_1 > 0.5 * S_0$$

Motion-Field Statistics are obtained using the motion characteristics of the neighboring MBs as follows:

$$\begin{aligned} SAD_{MB}^{Spatial} &= (SAD_L + SAD_{TL} + SAD_T)\ /\ 3 \\ SAD_{MB}^{Neighbors} &= (SAD_L + SAD_T + SAD_{TR} + SAD_{MB}^{Collocated})\ /\ 4 \end{aligned} \qquad \text{(Eq. 8)}$$

Coding-Mode-Field Statistics are obtained considering the coding modes of the spatial (in the current frame F_t) and temporal (in the previous frame F_{t-1}) neighboring MBs encoded as an I-MB.

$$\begin{aligned} INb_{Spatial} &= isI(MB_L^{Ft}, MB_T^{Ft}, MB_{TL}^{Ft}, MB_{TR}^{Ft}) \\ INb_{Temporal} &= isI(MB_R^{Ft-1}, MB_{DR}^{Ft-1}, MB_D^{Ft-1}, MB_{DL}^{Ft-1}) \\ INb_{TemporalTotal} &= INb_{Temporal} + isI(MB_{Collocated}^{Ft-1}) \\ &\quad + isI(MB_L^{Ft-1}, MB_T^{Ft-1}, MB_{TL}^{Ft-1}, MB_{TR}^{Ft-1}) \\ INb_{Total} &= INb_{Spatial} + INb_{Temporal} + isI(MB_{Collocated}^{Ft-1}) \end{aligned} \qquad \text{(Eq. 9)}$$

B. Step-2: Prognostic Early Mode Exclusion

The *Prognostic Early Mode Exclusion* scheme starts with a classification of MBs into two distinct groups using Eq. 10:

- *Group-A*: High-textured MB containing medium to fast motion
- *Group B*: Flat, homogenous regions with slow motion

$$Group_{MB} = \begin{cases} A, & (High_{MB}^{Textured} \,\&\, (\mu_B \| C_H)) \| V_H \| EDir_{MB}^{Dominant} \\ & \|(S_{MB}^{StrongThick} \| S_{MB}^{StrongThin} \| S_{MB}^{ManyThin}) \\ & \|(High_{MB}^{Textured} \,\&\, SAD_{MB}^{Collocated} > Th_{SAD}) \\ & \|(\mu_{VB} \| (INb_{Total} > Th_{I4})) \,\&\, (!\,G_L) \,\&\, (!\,V_L)) \\ & \|((INb_{Spatial} > Th_{I4}) \,\&\, (S_{16x16} > Th_{Edge})) \\ B, & Otherwise \end{cases} \qquad \text{(Eq. 10)}$$

Fig. 5 and Fig. 7 present the pseudo-codes of *Prognostic Early Mode Exclusion* for both *Group-A* and *Group-B*, respectively. In case of *Group-A*, I16x16 is excluded (line 3) due to high texture and the best choice would most probably be P8x8 or I4x4. However, exclusion of P16x16 at this point is critical as a wrong exclusion may result in a significantly increased bit rate. Therefore, the exclusion decision of P16x16 is performed in the *Hierarchical Fast Mode Prediction* step. Lines 4-7 and 8-11 check for slow motion using the motion statistics of the spatial neighboring MBs and exclude the smaller block partitions and I4x4 (line 5, 9). Lines 12-15 detect a high texture and hectic motion region. In this case, I4x4 coding mode is selected and all other modes are excluded.

In case of *Group-B*, a more sophisticated scheme systematically excludes the most unlikely modes. Lines 3-5, 6-12, 13-27 check for slow motion, flat and homogenous region, respectively. In these cases, I4x4, P8x8 and smaller partition modes are excluded. If a homogenous MB is stationary, P16x16 is predicted to be the most probable coding mode; otherwise, I16x16 is additionally processed (line 8). Lines 15-18, 19-25, 28-31 detect low motion and dark low-to-medium texture to exclude I4x4 mode; otherwise, I4x4 mode is re-enabled to avoid significant visual quality loss. Lines 33-39 assure that modes with smaller block partitions are only excluded if low motion and/or low textured are detected.

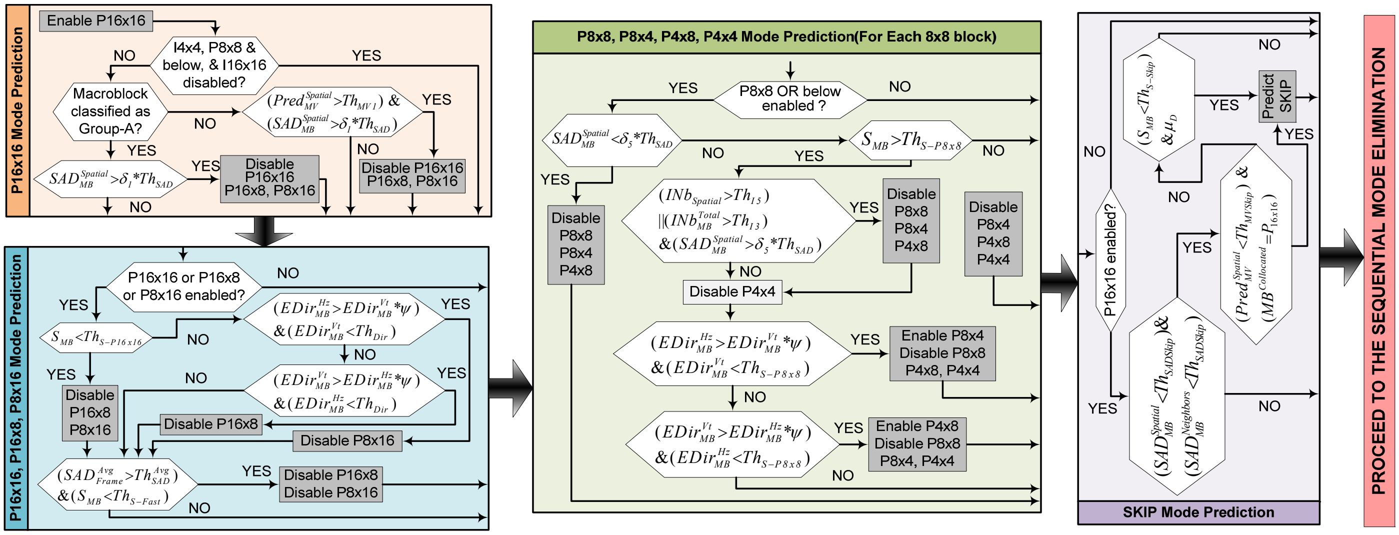

P16x16 Mode Prediction
Enable P16x16
I4x4, P8x8 & below, & I16x16 disabled?
NO
YES
Macroblock classified as Group-A?
NO
$(Pred_{MV}^{Spatial} > Th_{MV1})$ & $(SAD_{MB}^{Spatial} > \delta_1 * Th_{SAD})$
YES
NO
YES
$SAD_{MB}^{Spatial} > \delta_1 * Th_{SAD}$
YES
Disable P16x16 P16x8, P8x16
Disable P16x16 P16x8, P8x16
NO
P16x16, P16x8, P8x16 Mode Prediction
P16x16 or P16x8 or P8x16 enabled?
NO
YES
$S_{MB} < Th_{S-P16x16}$
NO
$(EDir_{MB}^{Hz} > EDir_{MB}^{Vt} * \psi)$ $\&(EDir_{MB}^{Vt} < Th_{Dir})$
YES
NO
YES
Disable P16x8 P8x16
NO
$(EDir_{MB}^{Vt} > EDir_{MB}^{Hz} * \psi)$ $\&(EDir_{MB}^{Hz} < Th_{Dir})$
YES
Disable P16x8
Disable P8x16
$(SAD_{Frame}^{Avg} > Th_{SAD}^{Avg})$ $\&(S_{MB} < Th_{S-Fast})$
YES
Disable P16x8 Disable P8x16
NO
P8x8, P8x4, P4x8, P4x4 Mode Prediction(For Each 8x8 block)
YES
P8x8 OR below enabled ?
NO
$SAD_{MB}^{Spatial} < \delta_5 * Th_{SAD}$
NO
$S_{MB} > Th_{S-P8x8}$
NO
YES
YES
Disable P8x8 P8x4 P4x8
$(INb_{Spatial} > Th_{I5})$ $\|(INb_{MB}^{Total} > Th_{I3})$ $\&(SAD_{MB}^{Spatial} > \delta_5 * Th_{SAD})$
YES
Disable P8x8 P8x4 P4x8
Disable P8x4 P4x8 P4x4
NO
Disable P4x4
$(EDir_{MB}^{Hz} > EDir_{MB}^{Vt} * \psi)$ $\&(EDir_{MB}^{Vt} < Th_{S-P8x8})$
YES
Enable P8x4 Disable P8x8 P4x8, P4x4
NO
$(EDir_{MB}^{Vt} > EDir_{MB}^{Hz} * \psi)$ $\&(EDir_{MB}^{Hz} < Th_{S-P8x8})$
YES
Enable P4x8 Disable P8x8 P8x4, P4x4
NO
NO
P16x16 enabled?
YES
$(SAD_{MB}^{Spatial} < Th_{SADSkip})\&$ $(SAD_{MB}^{Neighbors} < Th_{SADSkip})$
NO
YES
$(Pred_{MV}^{Spatial} < Th_{MVSkip})$ & $(MB^{Collocated} = P_{16x16})$
NO
$(S_{MB} < Th_{S-Skip})$ $\& \mu_D$
YES
Predict SKIP
YES
NO
SKIP Mode Prediction
PROCEED TO THE SEQUENTIAL MODE ELIMINATION

Fig. 6: Processing Flow of our Level-2 Hierarchical Fast Mode Prediction

```
GROUP-A: High-textured MB containing medium-to-fast motion
M = {P16x16, P16x8, P8x16, P8x8, P8x4, P4x8, P4x4, I16x16, I4x4}
// Initialize the possible coding modes with all modes
M ← M \ {I16x16};                                  // Exclude I16x16
If $(SAD_{MB}^{Spatial} < \delta_3 * Th_{SAD})$ Then
    M ← M \ {P8x8, P8x4, P4x8, P4x4, I4x4};        // Exclude I4x4, P8x8 and below
    return;                                        // Go to Step-3 (Section C)
End If
If $(Pred_{MV}^{Spatial} < Th_{MV1}) \& (SAD_{MB}^{Spatial} < \delta_4 * Th_{SAD})$ Then
    M ← M \ {I4x4};                                // Exclude I4x4
    return;                                        // Go to Step-3
End If
If $\begin{pmatrix} ((INb_{TemporalTotal} > Th_{I1}) \& (INb_{Spatial} > Th_{I2})) \| ((Pred_{MV}^{Spatial} \\ > Th_{MV2}) \& ((SAD_{MB}^{Collocated} > Th_{SAD}) \| (INb_{Total} > Th_{I3})) \end{pmatrix}$ Then
    M ← M \ {P8x8, P8x4, P4x8, P4x4};              // Exclude P8x8 and below
    return;                                        // Go to Step-3
End If
return;                                            // Go to Step-3
```

Fig. 5: Pseudo-Code of Group-A for Prognostic Early Mode Exclusion

C. Step-3: Hierarchical Fast Mode Prediction

Our *Hierarchical Fast Mode Prediction* (Fig. 6) performs a more refined second-level mode exclusion to obtain a set of candidate coding modes, which is later evaluated by the RDO-MD process with an integrated *Sequential RDO Mode Elimination* mechanism.

P16x16 Mode Prediction: If all modes except P16x16 are already excluded, then P16x16 is processed unless *SKIP* mode is detected in the last step of Fig. 6. On the contrary, P16x16 is excluded if the MB has fast motion and high texture.

P16x16, P16x8, P8x16 and P8x8 Mode Prediction: Based on the assumption *"the pixels along the direction of local edge exhibit high correlation, and a good prediction could be achieved using those neighboring pixels that are in the same direction of the edge"*, the main edge direction is investigated to split the MB accordingly. Hence, if the main edge direction is determined to be horizontal or vertical, P16x8 or P8x16 block type is chosen, respectively. A very small edge sum points out the presence of a homogeneous region, so only the P16x16 is processed.

Sub-P8x8 Mode Prediction: In case the SAD of the neighboring MBs is too high, P4x4 mode is predicted. In case the dominating horizontal or vertical edge direction is detected, P8x4 or P4x8 partition is selected, respectively.

Skip Mode Prediction: If SAD of an MB in P16x16 mode is significantly low, a perfect match could be very well predicted by ME. Such MBs are highly probable to be *SKIP*, thus saving complete ME computational load. Similarly, if the collocated MB is highly correlated with the current MB, then the probability of *SKIP* is very high e.g., the complete region is homogeneous.

```
GROUP-B: Flat, homogenous regions with slow-to-medium motion
M = {P16x16, P16x8, P8x16, P8x8, P8x4, P4x8, P4x4, I16x16, I4x4}
// Initialize the possible coding modes with all modes
If $(SAD_{MB}^{Spatial} <= \delta_1 * Th_{SAD})$ Then
    M ← M \ {P8x8, P8x4, P4x8, P4x4, I4x4};        // Exclude I4x4, P8x8 and below
End If
If $(V_{VL} \& G_{VL} \& (!S_{MB}^{StrongThick}) \& (!S_{MB}^{StrongThin}) \& (!S_{MB}^{ManyThin}))$ Then
    If $((SAD_{MB}^{Collocated} < \delta_3 * Th_{SAD}) \& (SAD_{MB}^{Spatial} < \delta_2 * Th_{SAD}))$ Then
        M ← M \ {I16x16};                          // Exclude I16x16
    End If
    M ← M \ {P8x8, P8x4, P4x8, P4x4, I4x4};        // Exclude I4x4, P8x8 and below
    return;                                        // Go to Step-3
End If
If $(V_L \& G_L \& S_L)$ Then
    M ← M \ {P8x8, P8x4, P4x8, P4x4, I16x16};      // Exclude I16x16, P8x8 and below
    If $((\mu_D \| C_L) \& (!High_{MB}^{Textured}))$ Then
        M ← M \ {I4x4};                            // Exclude I4x4
        return;                                    // Go to Step-3
    End If
    If $((Pred_{MV}^{Spatial} < Th_{MV1}) \& (SAD_{MB}^{Spatial} < \delta_2 * Th_{SAD}))$ Then
        M ← M \ {I4x4};                            // Exclude I4x4
    End If
    If $\begin{pmatrix} (SAD_{MB}^{Spatial} < \delta_5 * Th_{SAD}) \& (SAD_{MB}^{Neighbors} < \delta_5 * Th_{SAD}) \\ \& (!High_{MB}^{Textured}) \& (INb_{Spatial} > Th_{I4}) \end{pmatrix}$ Then
        M ← M \ {I4x4};                            // Exclude I4x4
        return;                                    // Go to Step-3
    End If
    return;                                        // Go to Step-3
Else
    If $((!High_{MB}^{Textured}) \& \mu_D \& G_L)$ Then
        M ← M \ {I4x4};                            // Exclude I4x4
        return;                                    // Go to Step-3
    End If
    Exclude I16x16 and Re-enable I4x4
    If $\begin{pmatrix} ((SAD_{MB}^{Spatial} < \delta_5 * Th_{SAD}) \& (SAD_{MB}^{Neighbors} < \delta_5 * Th_{SAD}) \& (!High_{MB}^{Textured}) \\ \& (INb_{Spatial} > Th_{I4})) \| ((Pred_{MV}^{Spatial} < Th_{MV1}) \& (isI(MB_{Collocated}^{Ft-1}))) \end{pmatrix}$ Then
        M ← M \ {I4x4};                            // Exclude I4x4
    End If
    If $(Pred_{MV}^{Spatial} > Th_{MV3})$ Then
        M ← M \ {P8x8, P8x4, P4x8, P4x4 };         // Exclude P8x8 and below
        return;                                    // Go to Step-3
    End If
    return;                                        // Go to Step-3
End If
```

Fig. 7: Pseudo-Code of Group-B for Prognostic Early Mode Exclusion

Moreover, if the MB lies in a dark region, the human eye cannot perceive small brightness variations. Thus, the insignificant distortion introduced by a forceful SKIP is tolerable here.

D. Step-4: Sequential RDO Mode Elimination

An integrated *Sequential RDO Mode Elimination* mechanism re-evaluates the candidate coding modes for sequential elimination, i.e. after P16x16 is processed, P16x8, P8x16, P8x8, and below are re-evaluated for elimination as specified in Fig. 6. However, for *Sequential RDO Mode Elimination*, the spatial SAD and MV values are replaced by the actual SAD and MV of the previously evaluated mode.

V. QP-Based Thresholding

As discussed in Section A, *QP-based thresholds* are used to categorize different features of video frames. For higher QP values, the effect of texture and motion becomes blurry due to the increased number of zero coefficients. It follows the fact that finding a good prediction is easier for ME, thus the number of injected I-MBs decreases. Therefore, with changing QP values, the thresholds (related to the decisions operating on the referenced frames) need to be adapted. We have performed extensive experimentation using different QPs (12 to 40) and several video sequences (only a small subset all of sequences used for validation in Section IV) to evaluate these thresholds. Afterwards, we have performed polynomial curve fitting using MATLAB to obtain threshold equations as a function of QP, see Eqs. 11-12[2]. Our empirical analysis revealed that only the thresholds for SAD, edge sum and MV (thus the major characteristics for motion and texture detection) react to the changing QPs. Table 1 presents the remaining thresholds (which are not affected by changing QPs).

$$TH_{SAD}=\begin{cases} 2500, & QP_{prev}<20 \\ 9000, & QP_{prev}\geq 40 \\ -0.3QP_{prev}^{3}+38.3QP_{prev}^{2}-115.9QP_{prev}+11897, & Otherwise \end{cases}$$

$$TH_{High}^{E}=\begin{cases} 10000, & QP_{prev}<20 \\ 13000, & QP_{prev}\geq 28 \\ -31.25QP_{prev}^{2}+1875QP_{prev}-15000, & Otherwise \end{cases} \qquad \text{(Eq. 11)}$$

$$TH_{Low}^{E}=\begin{cases} 8000, & QP_{prev}<20 \\ 10000, & QP_{prev}\geq 24 \\ 500QP_{prev}-200, & Otherwise \end{cases}$$

$$TH_{(MV1,MV2,MV3)}=\begin{cases} (20,45,30), & QP_{prev}\leq 28 \\ (30,55,40), & QP_{prev}\geq 36 \\ 1.25QP_{prev}-(15,10,5), & Otherwise \end{cases} \qquad \text{(Eq. 12)}$$

TABLE 1: THRESHOLDS AND MULTIPLYING FACTORS USED IN ACCORES

Thresholds								
Brightness	μ_{VD}	70	Variance	V_{VL}	0.5	Texture Edge	Th_{Dir}	1000
	μ_{D}	85		V_{L}	1.25		$Th_{S\text{-}Fast}$	5000
	μ_{B}	135		V_{H}	2		$Th_{S\text{-}Slow}$	1350
	μ_{VB}	175	Gradient	G_{VL}	5		$Th_{S\text{-}P16x16}$	500
Contrast	C_{L}	0.2		G_{L}	10		$Th_{S\text{-}P8x8}$	1000
	C_{H}	0.7		G_{H}	15		Th_{Edge}	200
Intra Neighbors	Th_{I1}	6	Intra Neighbors	Th_{I4}	1	SKIP	$Th_{MV\text{-}Skip}$	3
	Th_{I2}	4		Th_{I5}	2		$Th_{SAD\text{-}Skip}$	323
	Th_{I3}	5	Motion	Th^{Avg}_{SAD}	2500		$Th_{S\text{-}Skip}$	4096
Multiplying Factors								
Motion	δ_1	0.4	Motion	δ_4	0.6	Texture Edge	Ψ	2.5
	δ_2	0.6		δ_5	1.4		ε	0.7
	δ_3	0.5		δ_6	1			

VI. Results and Evaluation

For evaluation and validation of our proposed *ACCoReS*, we compare it with several state-of-the-art fast RDO-MD schemes and *exhaustive* RDO-MD. Common test conditions are: JM 13.2, IPPP, 1 reference frame, search range = 16. We have encoded various QCIF and CIF video sequences (low to fast motion) with different QPs (12, 16, 20, 24, 28, 32, 36, and 40) using an Intel 6600 (2.4 GHz, 2GB RAM, Windows XP) PC. **NOTE:** All speedup results include the overhead of *ACCoReS* and computation of video statistics in software.

A. Comparison with State-of-the-Art RDO-MD Schemes

We have compared our *ACCoReS* with several state-of-the-art fast RDO-MD schemes for quality (a positive Δ*PSNR* shows PSNR loss) and performance using the similar coding conditions as specified by the corresponding scheme. Table 2 shows that, compared to state-of the-art approaches ([10], [22]-[26]), *ACCoReS* achieves a speedup of up to 9.14x (average 3.05x) at the cost of an average PSNR loss of 0.66 dB. The significant speedup comes from the *Prognostic Early Mode Exclusion* and *Hierarchical Fast Mode Prediction* that curtails the set of candidate coding modes for further evaluation.

TABLE 2: PERFORMANCE AND QUALITY COMPARISON OF OUR ACCORES WITH SEVERAL STATE-OF-THE-ART FAST MODE DECISION SCHEMES

	ΔPSNR [dB]	Speedup [x]	ΔPSNR [dB]	Speedup [x]		ΔPSNR [dB]	Speedup [x]
Sequence	**Jing'04 [10]**		**Salgado'06 [22]**		**Sequence**	**Kim'07 [23]**	
Mobile_CIF	1.25	9.14	1.02	1.72	*Mobile_CIF*	1.25	1.71
Paris_CIF	0.71	7.40	0.68	1.31	*Paris_CIF*	0.69	1.25
Foreman_CIF	0.52	6.13	0.40	1.54	*Foreman_CIF*	0.47	1.51
Sequence	**Wang'07 [24]**		**Yu'04 [25]**		**Sequence**	**Park'08 [26]**	
Paris_CIF	0.70	2.23	0.68	4.16	*Foreman_QCIF*	0.71	2.09
Foreman_CIF	0.49	2.24	0.47	3.57	*Container_QCIF*	0.58	2.84
Akiyo_CIF	0.25	1.44	0.24	2.58	*Salesman_QCIF*	0.79	2.11

B. Comparison with Exhaustive RDO-MD

Table 3 provides the comparison (average and maximum) of *ACCoReS* with the *exhaustive* RDO-MD for distortion, bit rate (a positive Δ*Bit Rate* shows the bit rate saving) and speedup. Each result for a sequence is the summary of 8 encodings using different QP values. The average PSNR loss is approximately 3%, which is visually imperceptible. However, our *ACCoReS* provides a significant reduction in the computational complexity i.e. performance improvement of up to 19x (average 10x) compared to the *exhaustive* RDO-MD. The major speedup comes from slow motion sequences (*Susie*, *Hall*, *Akiyo*, *Container*, etc.) as smaller block partitions and I-MB coding modes are excluded in the *Prognostic Early Mode Exclusion* stage.

TABLE 3: SUMMARY OF PSNR, BIT RATE, AND SPEEDUP COMPARISON FOR VARIOUS VIDEO SEQUENCES (EACH ENCODED USING 8 DIFFERENT QPS)

		AVERAGE			MAXIMUM		
	Sequence	ΔPSNR [%]	ΔBit Rate [%]	Speedup [x]	ΔPSNR [%]	ΔBit Rate [%]	Speedup [x]
CIF	*Bus*	3.35	6.69	9.07	4.63	12.00	11.56
	Susie	1.87	1.64	11.91	2.47	12.37	14.59
	Football	4.91	2.74	9.65	5.66	3.37	13.05
	Foreman	2.02	4.44	9.97	3.31	16.73	12.70
	Tempete	3.42	10.22	8.47	4.78	14.53	10.75
	Hall	1.82	6.79	12.33	4.34	29.92	14.81
	Rafting	4.29	4.51	9.72	4.84	5.67	12.62
	Mobile	3.38	6.42	8.52	5.05	11.61	10.99
	Am. Football	3.91	7.81	8.76	5.41	10.52	11.61
QCIF	*Akiyo*	0.61	-3.41	12.75	1.24	1.75	17.27
	Carphone	2.44	6.39	10.20	3.19	11.51	12.86
	Coastguard	2.53	4.58	9.35	4.04	11.32	12.53
	Container	1.06	-7.15	13.00	1.57	4.01	19.13
	Husky	4.83	5.73	7.71	6.18	7.44	10.31
	Miss America	0.73	-8.86	12.05	1.72	14.25	14.72
	News	1.77	-3.64	12.21	2.12	0.37	16.71

Fig. 8 presents the percentage mode exclusions with respect to the total possible mode combinations for a large set of video sequences. In the best case, up to 73% (average >50%) coding modes are excluded. Similar to Table 3, Fig. 8 also shows that the large number of modes are excluded in case of slow motion sequences (*Susie*, *Hall*, *Akiyo*, *Container*, etc.) due to the early exclusion of smaller block partitions and I-MB coding modes.

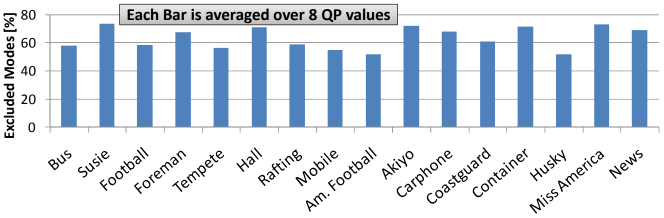


Fig. 8: Percentage mode excluded in our scheme for various video sequences

Fig. 9 shows the Rate-Distortion (R-D) curves of *ACCoReS* and *exhaustive* RDO-MD. The differences in R-D (PSNR loss of up to 5.76%) occur on PSNR values above 40-45 dB. These discrepancies are insignificant as the HVS is not able to recognize PSNR differences above 40-45 dB [3]-[5]. Mostly, *ACCoReS* achieves a much closer R-D as compared to *exhaustive* RDO-MD.

[2] Note: Eq. 11 is originally presented in [17] in context of a *Rate Control* but provided here for better understanding.

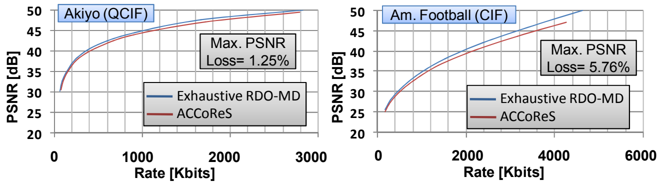


Fig. 9: Comparing Rate Distortion Curves for QCIF and CIF Sequences

C. *In-Depth Comparison with Exhaustive RDO*

Fig. 10 shows the in-depth comparison of *ACCoReS* with *exhaustive* RDO-MD for *Susie* sequence. It shows that *ACCoReS* suffers from an average PSNR loss of 0.8 dB (max: 1.4 dB, min: 0.19 dB), which is visually imperceptible (above 40 dB). However, *ACCoReS* achieves a significant reduction in the computational complexity, i.e. *ACCoReS* processes only 17% of SADs (reduced ME load which is the most compute-intensive functional block) compared to *exhaustive* RDO-MD. Red circles in the Fig. 10 show the region of sudden motion that causes disturbance in the temporal-field statistics. As a result, *ACCoReS* suffers from a higher PSNR loss but also provides high SAD savings. Moreover, *ACCoReS* maintains a smooth SAD computation curve, which is critical for embedded systems, while *exhaustive* RDO-MD suffers from excessive SADs. The PSNR curve shows that after frame 70, the mode prediction quality of *ACCoReS* improves due to the stability in the temporal-field statistics.

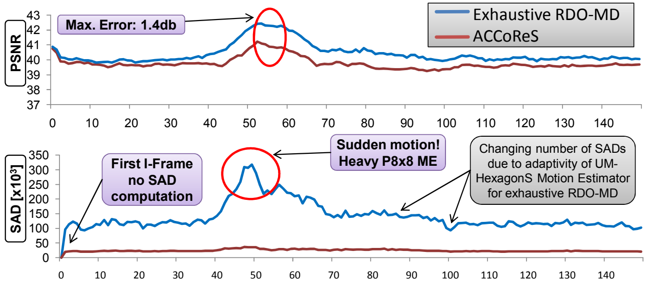


Fig. 10: Frame-Level in-depth Comparison for Susie Sequence

Fig. 11 shows the frame-wise distribution of correct mode selection by *ACCoReS* for *Susie* sequence. On average 74% of MBs are encoded with the correct mode (*MB Type* and the corresponding block size), i.e. as selected by the *exhaustive* RDO-MD. The correct modes predicted by *ACCoReS* range from 63% to 83%.

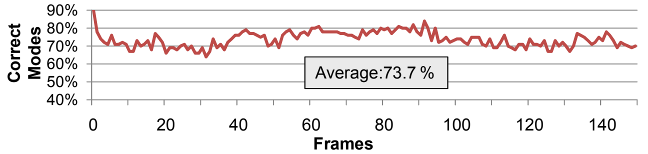


Fig. 11: Frame-Level in-depth evaluation of correct mode prediction

D. *Overhead of Computing Video Sequence Statistics*

The performance gain of our *ACCoReS* comes at the cost of additional computation of spatial and temporal video statistics. Experiments demonstrate that the PC-based software implementation of these statistics computations are 4.6% of the total encoding time using *ACCoReS*, which is already up to 19x smaller than the encoding time with *exhaustive* RDO-MD. Compared to the performance savings of our scheme, this overhead is negligible. We have also implemented various hardware accelerators for video statistics computation (considering H.264 encoder implementations for reconfigurable processors like [27]) to further reduce the processing overhead at the cost of additional hardware. Table 4 shows the area results for different accelerators synthesized for Xilinx Virtex-II xc2v3000 (ff1152) FPGA.

TABLE 4: AREA REQUIREMENTS OF HARDWARE ACCELERATORS FOR COMPUTING THE VIDEO SEQUENCE STATISTICS

Hardware Accelerators	AREA			Hardware Accelerators	AREA		
	Gate Eq.	Slice	LUTs		Gate Eq.	Slice	LUTs
Brightness	597	31	55	*Gradient*	1494	84	156
Contrast	1680	113	222	*Texture*	2190	129	250
Variance	767	47	50				

Note, the hardware processing of video statistics computation can be done in parallel, i.e. video statistics of next frame can be computed while the current frame is under encoding. The additional memory requirements are (#statistics)*#MBs*16bits, where (#spatial + #temporal statistics = 5 + 2). Note: Similar to fast RDO-MD schemes, ACCoReS (Fig. 5, Fig. 7, Fig. 8) is implemented in software for flexibility (as they are not fixed by the standard).

VII. CONCLUSION

We have presented a novel HVS-based *Adaptive Computational Complexity Reduction Scheme* (ACCoReS) that performs ***Prognostic Early Mode Exclusion** and **Hierarchical Fast Mode Prediction** to* curtail the set of possible coding modes. Compared to *exhaustive* RDO, our *ACCoReS* provides a performance improvement of up to 19x (average 10x) with an average 3% PSNR loss. *ACCoReS* excludes more than 70% of the possible coding modes even before starting the RDO-MD and ME. Our scheme is especially beneficial for low-cost performance and/or power-critical embedded systems where the available computational resources are limited. Our proposed scheme is quick and easy to be deployed in the real-world video encoding applications and exhibit a great industrial potential.

VIII. REFERENCES

[1] ITU-T Rec. H.264 and ISO/IEC 14496-10:2005 (E) (MPEG-4 AVC), "Advanced video coding for generic audiovisual services", 2005.

[2] H.264 Codec JM 13.2: http://iphome.hhi.de/suehring/tml/index.htm

[3] R. C. Gonzales, R. E. Woods, "Digital Image Processing", Prentice-Hall Inc., 2002.

[4] W. K. Pratt, "Digital Image Processing", John Willy & Sons Inc., 2001.

[5] Y. Wang, J. Ostermann, Y.-Q. Zhang, " Video Processing And Communications", Prentice-Hall Inc., 2002, Isbn 0-13-017547-1.

[6] Michelson, A. (1927). Studies in Optics. U. of Chicago Press.

[7] T. Wiegand, G. J. Sullivan, G. Bjntegaard, A. Luthra, "Overview of the H.264/AVC video coding standard", IEEE Transaction on Circuit and Systems for Video Technology (CSVT), vol. 13, pp. 560-576, 2003.

[8] J. Ostermann, et al., "Video coding with H.264/AVC: Tools, Performance, and Complexity", IEEE Circuits and Systems Magzine, vol. 4, no. 1, pp. 7-28, 2004.

[9] Arizona State University: http://trace.eas.asu.edu/yuv/index.html

[10] X. Jing, L.-P. Chau, "Fast Approach for H.264 Inter Mode Decision", Electronic Letters, pp. 1050- 1052, 2004.

[11] C. Grecos and M. Y. Yang, "Fast Inter Mode Prediction for P Slices in the H264 Video Coding Standard" IEEE Trans. on Broadcadting pp. 256- 263, 2005.

[12] F. Pan, et al., "Fast mode decision algorithm for intraprediction in H.264/AVC video coding" , IEEE CSVT, vol. 15, no. 7, pp. 813–822, 2005.

[13] B. Meng, et al., "Efficient Intra-Prediction Mode Selection for 4x4 Blocks in H.264", ICME, pp.III-521-III-524, 2003

[14] E. Akyol, D. Mukherjee, Y. Liu, "Complexity Control for Real-Time Video Coding", ICIP, pp.I-77-I-80, 2007.

[15] Y-W. Huang, et al., " Survey on Block Matching Motion Estimation Algorithms and Architectures with New Results", JVLSI, pp. 297-320, 2006.

[16] M. Shafique, B. Molkenthin, J. Henkel, "Non-Linear Rate Control for H.264/AVC Video Encoder with Multiple Picture Types using Image-Statistics and Motion-Based Macroblock Prioritization", ICIP, pp. 3429–3432, 2009.

[17] F. Pan, H. Yu, Z. Lin, "Scalable Fast Rate-Distortion Optimization for H.264-AVC", EURASIP Journal on Applied Signal Processing, Article ID 37175, pp. 1–10. vol. 2006.

[18] B. W. Jeon, J. Y. Lee, "Fast mode decision for H.264," in Joint Video Team (JVT) of ISO/IECMPEG & ITU-T VCEG 8th Meeting, Waikoloa, Hawaii, USA, December 2003, Document JVT-J033.

[19] K. P. Lim, et al., "Fast inter mode selection," in 9th JVT Meeting, San Diego, CA, USA, 2003, Document JVT-I020.

[20] E. Arsura, et al., "Fast macroblock intra and inter modes selection for H.264/AVC", ICME, pp. 378-381, 2005.

[21] M. Shafique, L. Bauer, J. Henkel, "3-Tier Dynamically Adaptive Power-Aware Motion Estimator for H.264/AVC Video Encoding", ISLPED, pp. 147-152, 2008.

[22] L. Salgado, M. Nieto, "Sequence Independent very fast Mode Decision Algorithm on H.264/AVC Baseline Profile", ICIP, pp.41–44, 2006.

[23] B-G. Kim, C-S. Cho, "A Fast Inter-Mode Decision Algorithm based on Macro-block Tracking for P Slices in the H.264/AVC Video Standard", ICIP, pp.V-301–V-304, 2007.

[24] X. Wang, et al., "Fast Mode Decision for H.264 Video Encoder based on MB Motion Characteristic", ICME, pp.372–375, 2007.

[25] A. C. Yu, "Efficient Block-Size Selection Algorithm for Inter-Frame Coding in H.264/MPEG-4 AVC", ICASSP, pp.III169–III172, 2004.

[26] I. Park, D. W. Capson, "Improved Inter Mode Decision based on Residue in H.264/AVC", ICME, pp.709–712, 2008.

[27] L. Bauer, M. Shafique, J. Henkel, "Efficient Resource Utilization for an Extensible Processor through Dynamic Instruction Set Adaptation", TVLSI, volume 16, issue 10, pp. 1295-1308, 2008.