# DESIGN OF COMPUTATION-AWARE MODE DECISION SCHEME FOR H.264/AVC

*Chuan-Yu Cho, Nicole Shih-Yu Huang, and Jia-Shung Wang*

Department of Computer Science, National Tsing Hua University, Hsinchu, TAIWAN

**ABSTRACT**

Mode decision is a kernel function of current video standards, especially while dealing with massive coding modes as the latest H.264/AVC variable block size partitions. The mode decision process is extreme time consuming because it contains the kernel motion estimation and intra prediction modules. By applying the concepts of using priority coding orders to encode a sequence, this paper concludes a set of mode decision strategies with observing among priority lists. A computation-aware mode decision scheme is then proposed to the H.264/AVC. As the results demonstrated, coding with priority orders can achieve computation-aware ability automatically and the priority info is very helpful for developing software-based computation-aware mode decision algorithms.

## 1. INTRODUCTION

Mode decision strategy affects the coding efficiency of a video codec very deeply because it deals with the kernel rate-distortion function. Especially while developing a video codec like the latest H.264/AVC, which includes up to 259 different block partition modes, intra prediction modes, and considerable multiple reference frames. Not only substance computations are involved in the mode decision processes, but also the most intelligent rate-distortion criteria are engaged. Therefore, to develop a successful video codec, it is necessary to put most effort on the mode decision processes.

The mode decision is a process to compromise between distortion and bit-stream size, that is, allocate bits to encode each coding block depending on its corresponding distortion. In a traditional block-by-block based coding algorithm, the rate-distortion performance is limited by the coding order, i.e., only local and coded blocks could be used as references to allocate the bit budgets. This bit allocation mechanism could be very unfair on visual quality without the consideration of entire frame variations, such as a motion or background block.

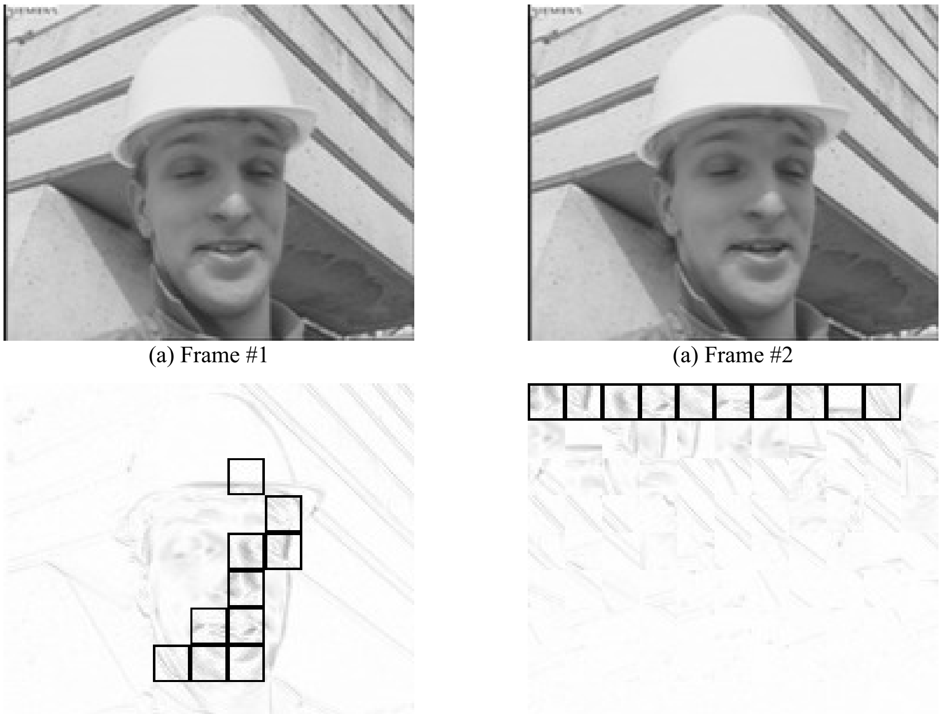
To overcome the problem of lacking view on global variations, some R-D algorithms, like MPEG-2 TM5, are developed by referring the coded results in pervious frames to update the current bit allocation strategies. However, the performance is not so good while at the initial stage or a shot change is happened. In fact, plenty of useful information is available for better mode decision strategies. [1] illustrates the idea of using initial mean square errors (MSEs) as a predictor to concentrate on processing the most distorted blocks, i.e., put effort on those blocks as much as possible. By focusing on those high MSE blocks, a computation-aware scheme can be obtained easily by simply omitting the small distortion blocks. In [2], some analyses on the MSE based priority lists are discussed, too. Base on those statistical analyses an adaptive search pattern switching scheme is proposed as a fast block matching algorithm.

Above applications portray that priority info exists in video sequences naturally, and could be easy extracted via simple pre-processing operates. As a result, we focus on the idea of using priority info in a video sequence to help with the mode decisions in H.264/AVC. We construct priority lists with the initial MSEs at the origins of coding blocks, and then observe the statistical relations between priority lists and the H.264/AVC coding modes. The relations are concluded as three observed rules and are used in our further mode decision scheme design for H.264/AVC.

The proposed mode decision algorithm comes out with a three-stage algorithm, which corresponding to the observed characters in priority lists. Furthermore, according to the stage-based design and priority coding order, the proposed mode decision scheme is also automatically a computation-aware scheme, and, therefore, is suit for software-based H.264 codec designs, such as DSP and PC platforms.

The rest of this paper is organized as follows: Section 2 presents the idea of extracting priority info from a video sequence, and then the statistical analyses of extracted priority lists are portrayed in Section 3. The Section 4 gives the proposed three-stage mode decision scheme and then

This research is supported in part by the National Science Council, Taiwan, R.O.C., under Contract NSC-94-2213-E-007-068 and 94-EC-17-A-04-S1-044.



(a) Frame #1 (a) Frame #2

(c) the difference between (a) and (b) (d) A priority list for frame #2

Fig. 1. Illustration for constructing a priority list.

experimental results are given in Section 5. Conclusion marks are drawn at the end section.

## 2. EXTRACTING OF PRIORITY INFO

Priority info can be found by applying preprocesses on the video sequences or by observing on the coding info in pervious frames. For example, the coding modes and motion vectors in pervious coded frames can be used as priority info while making mode decision of current coding frame. Also, we can use more sophisticated technologies such as object tracking or image segmentation algorithms to obtain different priority info by distinguishing foreground and background blocks. In this section, we demonstrate a simple but efficient way to obtain priority info and then build up a priority list for advanced mode decisions.

Consider the first two consequent frames in *Foreman* sequence in the Fig. 1 (a) and (b). Fig. 1 (c) is a residual picture of the MSEs between Fig. 1 (a) and (b). Since MSE is often used to evaluate the quality of video coding results with the quality matrix, Peak Signal-to-Noise Ratio (PSNR), we define the priorities of blocks by their MSEs. A block with higher MSE has higher priority because it has higher chances to reduce more MSE; on the contrary, the smaller MSE blocks have lower priorities, and as a result of contributing fewer MSE improvements even with a lossless encoding. Accordingly, we construct a priority list as in the Fig. 1 (d); the blocks in the priority list are descending sorted according to their MSEs. Here, the top 10 priority blocks in the residual picture, Fig. 1 (c), and the corresponding top 10 priority blocks appear in the priority list are highlighted by showing up their block boundaries.

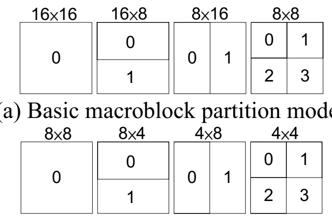
With above procedures, we can construct a priority list for every inter-frame, and since the higher MSE blocks at the beginning of the proposed priority list are more important, most of bits and computations should be allocated for these high priority blocks. Furthermore, the priority info in a priority list is not always fixed; the priorities could be changed after performing motion estimation and intra prediction procedures. Therefore, an update procedure is embedded to our proposed algorithms for always concentrating on the highest priority block.

The following section summarizes the observations on the relations of priority lists and the H.264/AVC mode decision results.

## 3. OBSERVATIONS ON PRIORITY LISTS

To understand the characteristics of priority lists, we first scrutinize the relations between the coding modes and the block-orders in the current priority list. Then, we discuss the initial block-order variation in the priority lists of two successive frames.

One of the most important features in H.264 is the variable block size partition modes. As shown in Fig. 2, the H.264 allows four basic macroblock partition modes in Fig. 2 (a). Four additional sub-block partitions are used while the best basic marcoblock partition mode is set to 8×8. That is, for each macroblock, it has up to 259 different partition combinations.




(a) Basic macroblock partition modes

(b) 8×8 Sub block partition modes

Fig. 2 Macroblock partition modes in H.264 [3].

Based on the observations on the relations between priority lists and the block coding modes, the following statistical characteristics are concluded:

1) A block is ordered at the top of the priority list, which is a large MSE block, has very high probability to be encoded as an 8×8 mode. On the contrary, the blocks at the tail end are usually encoded with simple coding mode, such as 16×16, 8×16 and 16×8.
2) If a block changes its order within 10% of its order in the pervious priority list, the block has very high chances to have the same coding mode as its pervious one. Otherwise, the coding mode tends to be a simple coding mode if the block order is moved from the head side to the tail side of the list, or the complex coding mode with sub-block partitions is used while the block moves form tail end to the head end.
3) If a block have gotten significant improvement after performing from the first to the second stage of the proposed algorithm, then it is possible that we can get further improvements by applying further complex coding modes on it. Otherwise, the block is considered as a non computation-distortion efficient block, which means we can not improve its MSE performance by using more complex coding modes.

According to above three observations, a progressive three-stage mode decision algorithm is proposed in the following section.

## 4. PROPOSED MODE DECISION SCHEME

Fig. 3 gives an example for the proposed three-stage progressive algorithm, and follows by the stage details.

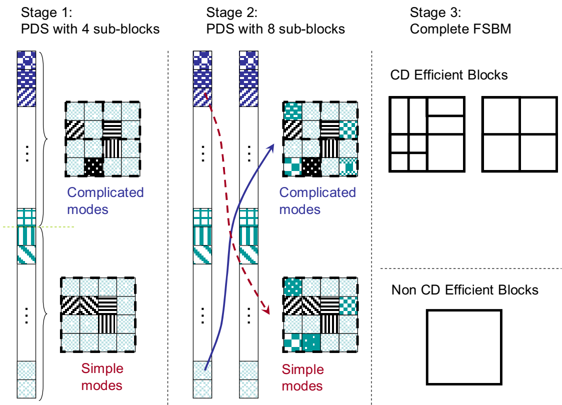



Fig. 3 The proposed 3-stage progressive algorithm.

Stage 1: Start FSBM using the 8×8 or 16×16 partition, and the first four maximum MSEs 4×4 sub-blocks at (0,0) are used for partial distortion search (PDS) [4].

Stage 2: The second stage uses the first eight maximum 4x4 sub-blocks to perform a further PDS algorithm. According to the second characteristic we mentioned above, the corresponding coding modes are chosen depending on whether the order variations are larger than 10% of the priority list length or not.

Stage 3: Perform complete FSBM with more complex coding modes if the blocks have significant improvements form stage 1 to stage 2. Otherwise, simple coding modes are used to deal with the non computation-distortion efficient blocks.

where in each stage, the up-merging post-processing is performed to merge up the block to a larger partition if the same motion vectors are found in successive sub blocks.

Due to the stage-based algorithm architecture, the computation-aware scheme in [1] can be easily integrated together. The 3-stage can be simply replaced with the motion estimation steps in the computation-aware scheme of [1], so that we can get a computation-aware mode decision scheme with progressive quality according to the available computation resource.

## 5. EXPERIMENTAL RESULTS

Experimental results regarding to our proposed mode decision algorithm is implemented on the JM9.3 reference encoder and is simulated with sequences, *Foreman*, *Salesman* and *Mobile&Calendar*, in CIF formats. For each sequence, the GOP structure is organized as IPPP…, and the number of reference frame is set to 1. All inter coding modes are enabled. The quantization parameter, QP, is fixed to 28, and the proposed method is evaluated with the exhaustive coding mode decision process in JM.

The following Table I summarizes that the proposal mode decision algorithm can reach a close performance to the H.264/AVC exhaustive search with scalable complexity usages. The maximum PSNR difference is only within 0.63 dB, which only happens in *Foreman* sequence. However, this degradation is very worth because it takes only 1/64 SAD computations comparing to the exhaustive full search algorithm. In addition, since the mode decision is actually different to the original H.264 exhaustive one, our bit rate is slightly increased, but still in a tolerable percentage with the keeping of high visual quality.

Table I Number of 4×4 SADs vs. PSNR and bit rate

| # of 4×4 SADs ( 1,000,000 ) | *Salesman* | | *Foreman* | | *Mobile&Cal.* | |
|---|---|---|---|---|---|---|
| | (a) | (b) | (a) | (b) | (a) | (b) |
| 46.06 (FS) | 35.93 | 0 | 37.15 | 0 | 34.18 | 0 |
| 21.59 | 35.95 | 5.16 | 37.08 | 6.53 | 34.13 | 2.78 |
| 11.45 | 35.93 | 5.11 | 37.01 | 5.89 | 34.08 | 2.60 |
| 5.68 | 35.89 | 4.07 | 36.93 | 5.20 | 34.04 | 3.41 |
| 2.85 | 35.87 | 2.28 | 36.91 | 4.47 | 33.99 | 3.28 |
| 1.43 | 35.85 | 2.51 | 36.89 | 5.70 | 33.98 | 2.73 |
| 0.72 | 35.84 | 2.99 | 36.88 | 5.72 | 33.97 | 2.91 |

*(a) PSNR (dB), and (b) Increased bit-rate (%).

From the view point of speedup ratio, the proposed algorithm provides a progressive computation-aware feature, which means the visual quality could be progressive improved depending on the available computations. In terms of 4×4 SAD calculations, we demonstrate the proposed algorithm can operate from 1 to 1/64 of the exhaustive full search scheme, and the speedup of motion estimation process could be speeded up to 20 times actually. However, due to the other parts of a H.264/AVC codec still takes considerable computations, the JM codec is finally speeded up for only three times. In this case, the motion estimation is no longer the bottleneck of the coding algorithm.

Table II Speedup performance

| # of 4×4 SADs ( 1,000,000 ) | *Salesman* | | *Foreman* | | *Mobile&Cal.* | |
|---|---|---|---|---|---|---|
| | (a) | (b) | (a) | (b) | (a) | (b) |
| 46.06 (FS) | | | | | | |
| 21.59 | 2.12 | 1.38 | 2.12 | 1.38 | 2.01 | 1.37 |
| 11.45 | 3.67 | 1.51 | 3.67 | 1.51 | 3.33 | 1.63 |
| 5.68 | 5.12 | 2.07 | 5.12 | 2.07 | 5.46 | 2.19 |
| 2.85 | 13.67 | 2.58 | 13.67 | 2.58 | 13.14 | 2.76 |
| 1.43 | 17.84 | 2.94 | 17.84 | 2.94 | 17.91 | 3.18 |
| 0.72 | 20.04 | 3.17 | 20.04 | 3.17 | 20.03 | 3.44 |

*(a) Motion estimation time speedup, and (b) Total time speedup.

In the last experiments, we show the performance of the hit-ratio to the correct partition modes. Two partition modes are defined as the same while they are either chosen as small or equal to 8×8 modes (8×8, 8×4, 4×8, 4×4), or large than 8×8 modes (16×16, 16×8, and 8×16). The following Fig.4 shows the hit ratio for *Salesman* sequence. We divide the priority list into three parts. As the figure shows, the hit ratio is as high as 90% in the head and tail parts of the priority list, but it is as low as 40% while in the middle part. Since the most high priority blocks are concentrated in the head of the priority list, the final output results are expected to have a good visual quality though the hit ratio is low in middle part.

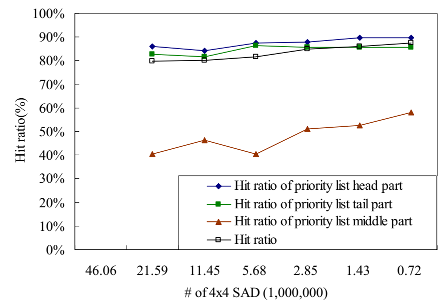



Fig 4. Number of 4×4 SAD vs. Hit ratio – Salesman.

## 6. CONCLUSIONS

As presented above, the proposed mode decision scheme portrays an efficient idea of using priority info to help with complicated mode decision problem in H.264/AVC. In addition, the proposed scheme is a computation-aware scheme, so that it is also quite suit for many mobile devices, which usually have multiple power saving operating modes and are equipped with software based coding algorithm.

This paper also brings the idea of using priority coding orders in video coding algorithm instead of traditional raster-scan and block-by-block coding manners. More different types of priority lists or characters might be further discovered, and might bring new thoughts to future video applications.

## 7. REFERENCES

[1] P.-L. Tai, S.-Y Huang, C.-T. Liu, and J.-S. Wang,“ Computation-aware scheme for software-based block motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.* Vol. 13, no 9, pp. 901-913, Sept. 2003.

[2] S.-Y. Huang, C.-Y. Cho and J.-S. Wang, “An Adaptive Fast Block Matching Algorithm Employing Profit List,” *IEEE Trans. Circuits Syst. Video Technol*, to be published.

[3] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra,”Overview of the H.264/AVC Video Coding Standard,” *IEEE Trans. Circuits Syst. Video Technol.* Vol. 13, no 7, pp. 560-576, July. 2003.

[4] C.K. Cheung and L.M. Po, “Normalized partial distortion search algorithm for block motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.,* vol. 10, no. 3, pp. 417-422, Apr. 2000.