# IMPROVING FLEXIBLE MACROBLOCK ORDERING OF H.264/AVC

*Li Song, Xin Ma*

Institute of Image Communication and Information Processing, Shanghai Jiao Tong University
song_li@sjtu.edu.cn, milanblood@hotmail.com

## ABSTRACT

H.264/AVC is the latest video coding standard from ITU-T. It provides superior and efficient video coding at low bit rates. Flexible Macroblocks Ordering (FMO) is an error–resilient tool included in H.264/AVC. In this paper, a novel technique for adaptive classification of macroblocks into two slice groups using FMO method is proposed. With our adaptive macroblocks allocation map, simulation results have shown the improved FMO works better with the error concealment tool in H.264/AVC. Simulations also show that our approach performs nearly as well as the chess-board pattern but has better coding efficiency.

***Index Terms***—Video coding, Error resilience, H.264/AVC

## 1. INTRODUCTION

Nowadays, with the explosion of streaming video services, how to maintain the end-to-end video's quality in the error prone environments has attracted more and more attention. As the most state-of-the-art video compression standard, H.264/AVC not only offer the better coding efficiency compared with other ones, but also provide some new error resilient tools. The error resilient tools in H.264/AVC standard include the slices of a picture, the intra block refresh, the multiple reference picture selection, the data partition and the parameter sets (PS) for sequence and picture, the flexible macroblock ordering (FMO), and the redundant slice (RS). Among them, the PS, FMO, and RS tools are introduced by H.264/AVC for the first time.

Although the error resilient tools can improve the video's robustness in transmission, they usually come with a certain overhead or cost. In [1], adaptive macroblock slice grouping scheme is proposed to classify the macroblocks of a frame into three categories to support unequal error protection. However, classification is performed in multi-pass iterating manner. This paper focuses on extending chess pattern FMO tools of current H.264/AVC to achieve better compromise between coding performance and the robustness, but without incurring complex computing.

Firstly, the existing FMO tools in H.264/AVC standard will be overviewed in section 2, and then in section 3, a novel adaptive FMO technique is proposed to improve the coding performance of the existing FMO mode in H.264, while keep robustness of video transmission provided by the chess pattern FMO mode. The experiments and the related results are given in section 4.

## 2. REVIEW OF THE FMO IN H.264/AVC

In video coding standards without FMO tools, macroblock (MB)s in a picture is coded by raster scan order. By using FMO mode in H.264/AVC [2], the MBs can be coded in different order by classifying the MBs into different slice groups according to the specific MB map (the macroblock allocation map, *MBAmap*), Every slice group can include one or several slices and coding order of MB can be flexible. Compared with the no FMO mode, FMO gives MB the arbitrary coding order which can make the MBs in one specific slice scattering in the whole picture, rather than in consecutive order. Video coding without FMO, a slice is lost will make consecutive MBs in the picture lost together. With FMO video coding, this situation can be avoided by setting specific MB maps in slice group classification.

Most efficient and popular error concealment techniques, either spatial or temporal error concealment uses the existing neighbor information to recover lost block, and these method's performance are mostly dependent on the amount of the available neighbor information [3]. The consecutive MBs loss is more difficult to recover than the scattering MBs loss. Videos coded with FMO are more robust than those coded without FMO. Among all 6 FMO patterns defined in H.264/AVC, the chess pattern is widely used to increase robustness, which classifying all MBs into two slice groups as the chess-board. With this FMO type, the missing macroblock can be reconstructed in a very effective way using interpolation based on surrounding (available) sample values. This also keeps the drifting of prediction under control compared to the case no FMO was used [4]. Fig.1 gives the example of one slice loss in no FMO mode and chess pattern FMO mode. We can find that the chess pattern FMO can offer more neighbor information for the lost MBs to be recovered.

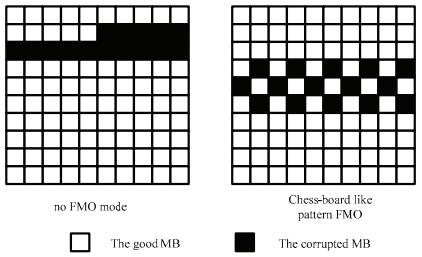


**Fig. 1 No FMO mode and chess-board like pattern FMO**

H.264/AVC's coding performance is largely gained from the spatial prediction among the adjacent MBs, include the intra block prediction, the motion vector prediction and the context based entropy coding etc. Therefore, coding with FMO like chess pattern, where one MB's four neighbor MBs are all in other slice groups, will decrease spatial correlations among the MBs in one slice and result in the poor coding efficiency.

## 3. THE PROPOSED ADAPTIVE FMO METHOD

As discussed above, the FMO mode can provide the better robustness than the no FMO mode, but the coding performance could drop due to many smaller or dispersed regions. Here we propose a new adaptive FMO scheme that keep the chess pattern FMO's robustness while improve its coding performance.

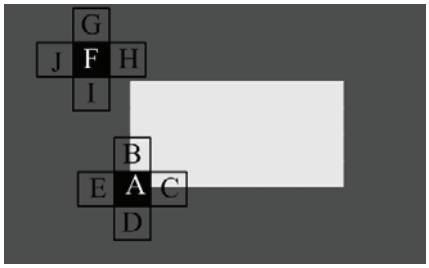


**Fig. 2 The two different type MBs**

The provision for dynamic formation of slice groups is exploited by the proposed approach. Specifically, macroblocks are classified into two slice groups with respect to their relative importance. Group A is those MBs with strong edges or complex texture (*A-type* MBs), and the other MBs in the picture form group B (*B-type* MBs). As shown in fig. 2, the MB *A*, *B*, *C* are *A-type* MBs, and *D*, *E*, *F*, *G*, *H*, *I*, *J* are *B-type* MBs. When an *A-type* MB is lost, either in intra frame or inter frame, a concealment algorithm may then leverage neighboring *A-type* MBs to conceal the missing blocks much more effectively. For example, correct recovery of the lost *A* will depend more on *B* and *C*, rather than *D* and *E* in fig. 2. For spatial error concealment, *B* and *C* help to predict the MB *A*'s structure while *D* and *E* fails. In the case of temporal error concealment, *B* and *C*'s spatial and motion information also include better motion information according to [5] , which are more consistence with *A*'s motion. If neither *B* nor *C* is available, A may not be correctly recovered just by *D* and *E*, since we cannot distinguish whether *A* is at the corner of the rectangle or in a plain area like *D* or *E*. But for the *B-type* MBs, it can be easily recovered only by little neighbor information, as demonstrated in fig. 2, the lost *F* can be recovered by any one of its four neighbor MBs, *G*, *H*, *I* and *J*.

Based on the above analysis, there's no need to equally disperse all MBs in a frame like chess pattern FMO in H.264/AVC. Only *A-type* MBs need to be rearranged in chess-board like pattern to ensure necessary surrounding information available when error happens. In other words, we hope to preserve the spatial correlation between the MBs in the same group as much as possible.

H.264/AVC standard provides the explicit FMO mode for users to use their own FMO types for different slices, so we can make use of this mode for our adaptive FMO mode. The detail of our method will be discussed in the following parts, include the *A-type* MB detecting, the Slice Group determination and the creation and adjustment of the *MBAmap*.

### 3.1 The MB classification

As discussed above, an *A-type* MB is the one has strong edges or complex textures. In our method, we use the gradient information of a frame to detect such MBs. The gradient image *G(f)* of a frame *f* can be calculated according to [6] as:

$$G(f) = \frac{1}{3}\sum_{i=1}^{3}[((f \oplus B_i) - (f \Theta B_i)) \Theta B_{i-1}] \quad (1)$$

where $B_i$ is a group of square structuring element, with the size of (2i+1)x(2i+1). $\oplus$ and $\Theta$ denote the dilation and erosion respectively. The example of the gradient image is shown in figure 3(b). Then the average gradient value of a MB $B(i,j)$ at MB position $(i,j)$ is calculated by

$$G(i,j) = (\sum_{(x,y)\subset B(i,j)} G(f(x,y))) / 256 \quad (2)$$

where $G(f(x,y))$ denotes the pixel gradient value at position $(x,y)$ in frame *f*, $G(i,j)$ is the average gradient value of the MB $B(i,j)$. We set a threshold to detect *A-type* MBs according to the average gradient value of a MB. If the average gradient value greater than the threshold, we think the MB is a *A-type* MB:

$$B(i,j) \subset \begin{cases} A\text{-}type\ \text{MB} & if\ (G(i,j) > th_f) \\ B\text{-}type\ \text{MB} & otherwise \end{cases} \quad (3)$$

$th_f$ is the threshold determined by experiments. The example of the *A-type* MBs has been shown in figure 3(c). One can see *A-type* MBs mainly correspond to regions with edges.

### 3.2 The *MBAmap* construction

If there is an *A-type* MB lost, we hope that more neighboring MBs with similar structure can be used to conceal the error. Therefore, all *A-type* MBs in a frame are divided into 2 slice groups like chess type pattern:

$$B(i,j) \subset \begin{cases} slicegroup\ 0 & if\ ((i+j)\%2 == 0) \\ slicegroup\ 1 & otherwise \end{cases} \quad (4)$$

For the *B-type* MBs, according to the above discussion, we can combine them with one of the above two slice group, for example, *slice group 1*, to increase spatial correlation between the MBs. The final *MBAmap* is shown in figure 3(d).

### 3.3 The *MBAmap* adjustment

In H.264/AVC standard, the explicit MB map and the slice group information of each MB are written in the Picture Parameter Set (PPS). So adoption of the explicit MB map will incur additional overhead. In order to avoid the unnecessary overhead, we further adjust the MB map obtained in previous step.

If the ratio of the *B-type* MBs is small, the coding gain of the *MBAmap* will be limited. In this situation, we adopt the chess pattern MB map as below:

$$M_a(f)=\begin{cases}M(f) & if\ (r_n<th_n)\\ M_c(f) & otherwise\end{cases} \quad (5)$$

Where $M(f)$ denote the *MBAmap* obtained in section 3.2, $M_c(f)$ is the chess pattern MB map, $M_a(f)$ is the adjusted *MBAmap*, $r_n$ is the ratio of the *B-type* MBs in frame *f* and $th_n$ is the predefined threshold.

For inter coding that consecutive frames are static or few motion, error concealment mainly get rewards from previous zero-motion macroblocks. So we also adopt the standard chess MB map to avoid the extra overhead. Specifically, whether a MB $B(i,j)$ is a static one is determined by the mean absolute difference (MAD) between $B(i,j)$ and the MB in the previous frame at the same position $(i,j)$. If the MAD is less than the predefined threshold, the MB will be defined as the static MB. The ratio of the static MBs in the frame is used to decide whether the frame is a static frame. Specially, *MBAmap* is adjusted as follows for inter frame:

$$M_a(f)=\begin{cases}M(f) & if((r_n<th_n)or(r_s>th_s))\\ M_c(f) & otherwise\end{cases} \quad (6)$$

Where $r_s$ denote the ratio of the static MBs in *f*, $th_s$ is the predefined threshold.

After performing the above three steps, we get the final MB allocation map for a frame to be encoded. Such adaptive MB coding order will be helpful both spatial prediction at encoder and error concealments at decoder than the standard chess pattern FMO.

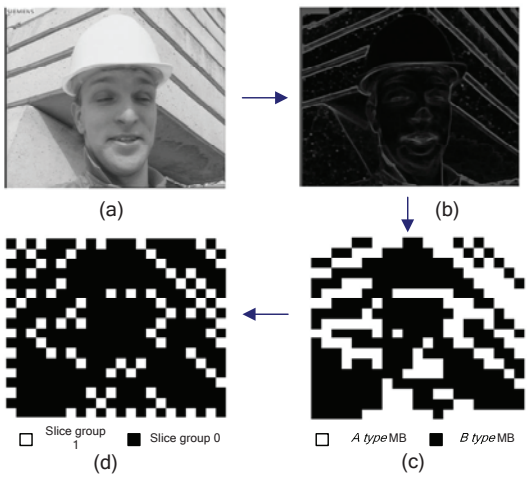


**Fig. 3 The process of the creation of the MB map for the 1st frame in foreman sequence: (a) is the frame to be encoded, (b) is the gradient image of (a), (c) is the *A* and *B-type* MB detection results of (a), (d) is the resultant *MBAmap***

## 4. EXPERIMENT RESULTS

To test our proposed adaptive FMO mode's robustness and the coding performance, we compare our method with the chess pattern FMO and the no FMO mode in H.264/AVC. We take 3 test sequences in our experiment, including *foreman* (150frames), *football* (250frames) and *hall* (250frames); all the sequences are CIF (352x288) resolution at 30fps. We use the H.264 reference software JM10.2 with baseline profile to encode these sequences with the GOP size equal to 12, where the first frame of GOP is coded as Intra frame, and the remaining frames are coded as the P frames with one reference frame. The QP is 30 and slice size is 30 MBs, where each slice is transmitted as a packet. The predefined *A-type* detection thresholds $th_f$ = 20. Although the optimal $th_n$ and $th_s$ may be different for sequences according to their own contents and characteristic, we use adjustment threshold $th_n$ = 0.1, $th_s$ = 0.95 for all the test sequence to avoid increasing complexity. The encoded sequences suffer from different transmission environments whose packet loss rate (PLR) are 3%, 5%, 10%, 20% respectively by the packet loss simulator[7]. Table 1 gives the results of the sequences for three different methods.

The error concealment method used in our experiment is the combination of the intra interpolation method and the EBME (external boundary matching error) described in [3], which is a little different from the methods used in JM 10.2[8]. In JM 10.2, the error concealment methods are a combination of the intra interpolation method and the BMA (boundary match algorithm). But in [3], it has been shown that the EBME algorithm performs better than BMA, while has the same computation complexity. So we replace the BMA with EBME in JM 10.2 in our experiment. The results of the error concealment in different error environments for the three FMO modes have been shown in table 2. Fig. 4 also gives the example of the subjective quality comparison between three FMO modes. The RD performances with different QP are illustrated for *foreman* in Fig. 5.

From the experiment results, we can find that the proposed adaptive FMO mode can save about 3%-5% bits for the whole sequence and 8%-16% bits for the intra coded pictures, compared with the chess type FMO mode. In H.264/AVC, encoding of the intra coded picture is entirely dependent on the spatial correlation among the adjacent MBs, our proposed adaptive FMO mode can save more bits in intra coded frame than in the inter frames as preserving more spatial correlation in the MB map. The error concealment results under different packet loss rate show that our proposed adaptive FMO mode performs nearly as well as chess pattern FMO but has better coding efficiency and are much better than those in no FMO mode.

And the added computation complexity in our method lies mostly in counting the gradient information of the frame, which is minor compared with the computation complexity of the standard encoding process and result in only about 1%-2% increase of the encoding time in our experiment.

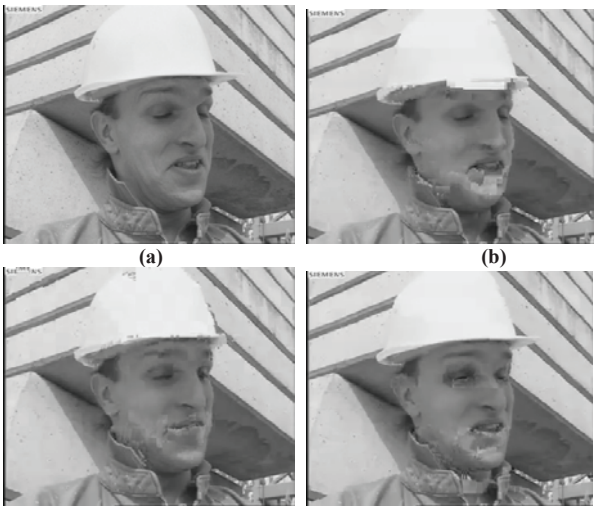

**Fig. 4 The recovered 75th frame of *foreman* with PLR = 20%, (a) no error frame (36.26dB) (b) no FMO mode (24.92dB), (c) chess pattern FMO (29.82dB), (d) proposed method (30.12dB).**

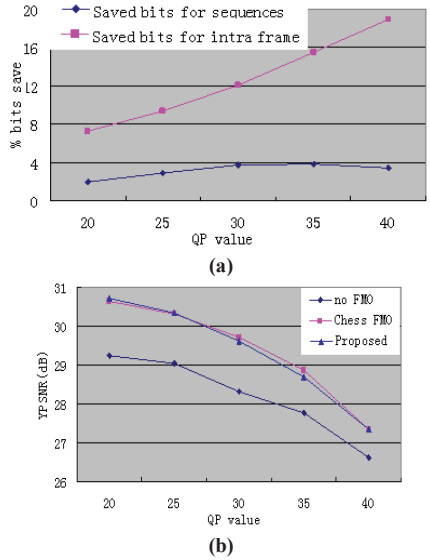


**Fig. 5 The RD performance comparison of three modes for *foreman* under different QP with the PLR=20%, (a) coding gain (saved bits), (b) *PSNR* with same error concealment method.**

**Table 1 The encoding results for different FMO modes**

| sequence | Mode | Ypsnr | Bits/frame | Bits/intra frame |
|---|---|---|---|---|
| foreman | no FMO | 35.78 | 14520.2 | 58984.0 |
| | Chess FMO | 35.78 | 16761.2 | 76968.0 |
| | proposed | 35.77 | 16125.8 | 67641.2 |
| | Bits saving | | **3.8%** | **12.1%** |
| football | no FMO | 35.00 | 41717.7 | 77950.9 |
| | Chess FMO | 34.92 | 46523.4 | 97970.7 |
| | Proposed | 34.96 | 44905.3 | 89510.6 |
| | Bits saving | | **3.5%** | **8.6%** |
| hall | no FMO | 36.64 | 10625.1 | 57562.7 |
| | Chess FMO | 36.63 | 12457.6 | 79018.7 |
| | Proposed | 36.63 | 11818.1 | 66316.4 |
| | Bits saving | | **5.1%** | **16.1%** |

**Table 2 The PSNR results of error concealment**

| sequence | PLR | no FMO | Chess FMO | proposed |
|---|---|---|---|---|
| foreman | **3%** | 33.93 | 34.59 | 34.48 |
| | **5%** | 32.57 | 33.73 | 33.46 |
| | **10%** | 30.67 | 32.31 | 32.15 |
| | **20%** | 28.33 | 29.72 | 29.62 |
| football | **3%** | 30.60 | 31.69 | 31.83 |
| | **5%** | 28.29 | 30.03 | 29.83 |
| | **10%** | 25.55 | 27.53 | 27.39 |
| | **20%** | 22.23 | 23.90 | 23.79 |
| hall | **3%** | 35.47 | 35.88 | 35.77 |
| | **5%** | 34.16 | 34.75 | 35.00 |
| | **10%** | 32.58 | 33.98 | 33.99 |
| | **20%** | 29.42 | 31.37 | 31.49 |

## 5. CONCLUSION

In this paper, we propose an adaptive FMO mode for H.264/AVC standard, which is more efficient than the chess type FMO used in H.264/AVC in coding performance and offer the nearly same robustness as the chess pattern FMO does.

## 6. ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (60702044, 60632040 and 60625103).

## 7. REFERENCES

[1] N. Thomos, S. Argyropoulos, N. V. Boulgouris and M. G. Strintzis, "Robust Transmission of H.264/AVC Streams Using Adaptive Group Slicing and Unequal Error Protection," *EURASIP J. Applied Signal Processing*, pp. 1-13, 2006, Article ID 51502.

[2] P. Lambert, W. De Neve, Y. Dhondt and R. Van de Walle, "Flexible macroblock ordering in H.264/AVC", *J. Vis. Comm. Image Represent.* , pp. 358-375, 2006, 17, (2)

[3] D. Agrafiotis, D. Bull, and C. N. Canagarajah, "Enhanced Error Concealment with Mode Selection", *IEEE Trans. Circuits Syst. Video Tech.*, August. 2006

[4] B. Katz, S. Greenberg, N. Yarkoni, N. Blaunstien and R. Giladi, "New Error-Resilient Scheme Based on FMO and Dynamic Redundant Slices Allocation for Wireless Video Transmission", *IEEE Trans. Broadcasting*, vol. 53, no. 1, pp308-319, March 2007.

[5] S. Y. Chien, Y. W. Huang, B. Y. Hsieh, S. Y. Ma and L. G. Chen, "Fast Video Segmentation Algorithm With Shadow Cancellation, Global Motion Compensation, and Adaptive Threshold Techniques", *IEEE Trans. Multimedia*, vol. 6, no. 5, pp732-748, October 2004

[6] D. Wang, "Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking", *IEEE Trans. Circuits Syst. Video Tech*, vol. 8, no. 5, pp539-546, September 1998.

[7] M. Luttrell, S. Wenger, M. Gallant, "New versions of packet loss environment and pseudomonas tools", *ITU-T SG16 Document Q15-I-09*, October, 1999

[8] Y.K. Wang, M.M. Hannuksela, V. Varsa, A. Hourunranta and M. Gabouj, "The Error Concealment Feature in the H.26L Test Model."*Proc. IEEE Int. Conf. on Image Proc.*, 2, 2002, 729-732.