ITU - Telecommunications Standardization Sector
STUDY GROUP 16 Question 6
Video Coding Experts Group (VCEG)
36th Meeting: San Diego, USA, 8-10 October, 2008

Document VCEG-AJ21
Filename: VCEG-AJ21.doc

Question: Q.6/SG16 (VCEG)

Source: Jaeil Kim, Taeyoung Na, Changhee Kim, Bumshik Lee, Munchurl Kim
Information and Communications Univ.
119 Munji street, Yuseong-Gu, Daejeon, Korea
Tel: +82-42-866-6246
Email: jaeil1203@icu.ac.kr
tyna@icu.ac.kr
changhee21c@icu.ac.kr
bslee@icu.ac.kr
mkim@icu.ac.kr

Changseob Park, Sangjin Hahm, Injoon Cho
Korean Broadcasting System
11 Yoidodong, Youngdeungpo-Gu, Seoul, Korea
Tel: +82-2-781-5963
Email: changseob@kbs.co.kr
cashy@kbs.co.kr
sylphide@kbs.co.kr

Title: **Enlarging MB size for high fidelity video coding beyond HD**

Purpose: Proposal

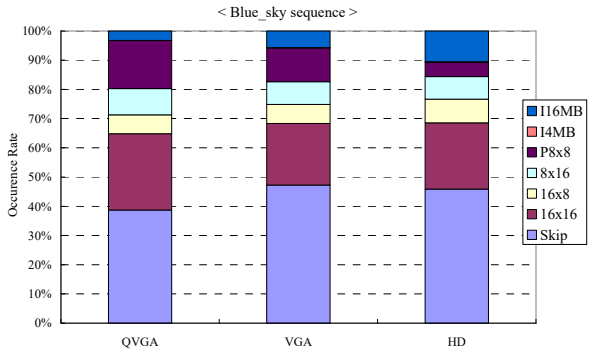
Abstract

In this contribution, we propose to extend the architecture of H.264 codec with enlarged MB sizes. Most of video codecs have lived long with MB based coding, which is now limiting to the coding efficiency toward the video coding of beyond High Definition (HD) resolutions. Even if the High profile of H.264, called FREXt tools, for high bit rate contents has been developed, it still maintains the size of the MB to be 16x16 for ME and MC based predictive coding as well as 16x16 intra coding. To limit the MB size into 16x16 might be one of the factors to restrict the RD performance in the high resolution video. Thus, we attempt to improve the coding efficiency of H.264 by enlarging MB size to the maximum 128x128. This attempt has been made with the extension to JM reference software. For this, the followings are to be newly defined, redefined or extended: new MB mode types for enlarged MB, CBP for new MBs or block indexing etc. In this proposal, the extended architecture with the enlarged MB structure are first proposed for ME&MC but new designs of transform kernels, intra prediction, luma/chroma DC transforms, CABAC context models and de-blocking filters for the enlarged MB are not yet addressed.

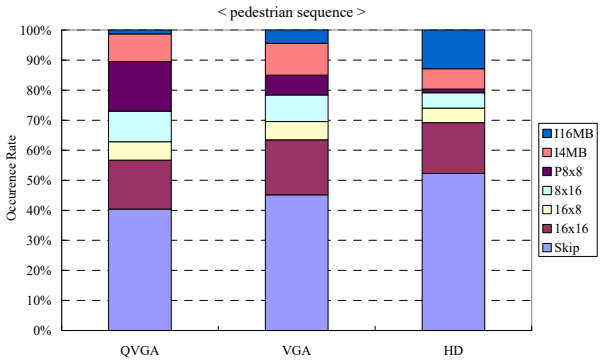
1. Introduction

The demand for HD or beyond HD such as ultra high definition videos of 4k and 8k has been increasing with increased network bandwidth and developed display technology. However, the current video encoder has much limitation to support HD or beyond HD video contents. Since the characteristics of the video signals depends heavily upon spatial resolutions, it is necessary to devise an appropriate video coding architecture to reflect the video characteristics with different sizes, especially the high resolution video contents. The High profile of H.264/AVC provides several tools such as 8x8 integer transform, 8x8 intra prediction and 8 to 12 bit-depth processing etc., which allows for better RD performance for high resolution video, for example, SD or HD video contents. Nevertheless, the maximum MB size for ME and MC is maintained with 16x16 pixels. It may not be efficient for the HD or beyond HD since the MB size of 16x16 pixels is relatively small for high resolution video contents. Fig.1 shows the relative portions of the final block mode selections according to the spatial resolutions of different video contents. These statistics of block mode distributions have been collected from the experiments with H.264

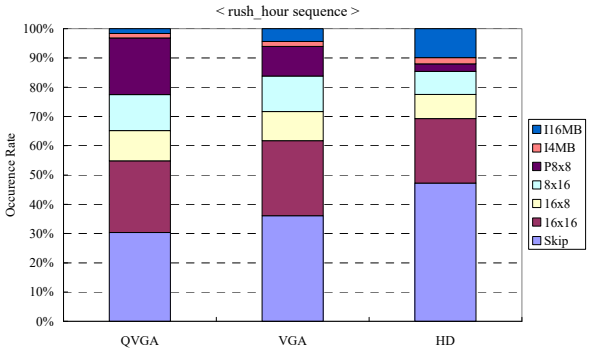
reference software, Joint Model ver. 12.0, for *Blue Sky*, *Pedestrian*, *Rush Hour* and *Station* sequences in QVGA, VGA and HD resolutions. As shown Fig.1, it can be noticed that the portions of the large block modes such as SKIP and 16x16 modes increase in proportion to the video resolution sizes. The mode selection trend shows consistent results for all the sequences with different resolutions and different QP values.



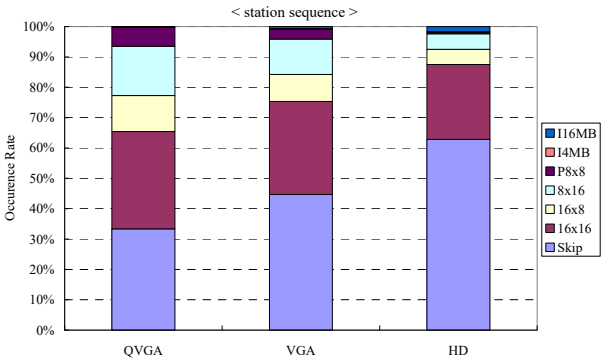
(a) Blue Sky Sequences



(b)Pedestrian Sequences



(c) Rush Hour Sequences



(c) Station Sequences

Fig. 1 Block mode distributions versus spatial resolutions of video

On the other hands, the proportions of the block modes in small spatial resolutions such as P8x8 and I4x4 significantly are reduced, especially for the HD sequences. It implies that relatively large block modes are more advantageous to the high resolution videos in terms of the Rate-distortion optimization. Motivated from these observations, we propose to enlarge the macroblock (MB) of 16x16 for HD or beyond HD video sequences.

Recently, similar works have been introduced in [1-4], and Ma *et. al*. has analyzed the advantage of larger MB with a 16x16 interger transform, a modified RD model and enlarged search ranges for ME in terms of RDO costs for high resolution videos. In this contribution, we attempt to extend with enlarged MB sizes the JM reference software architecture such that the fair comparison and backward compatibility can be supported. Furthermore, the extensions to the current syntax and the required side information for enlarged MB sizes are addressed. In Section 2, how to design the enlarged MB are first explained. Its subsequent works such as new mode definition, CBP for newly enlarged MB's, block indexing and deblocking filters are described.

2. Proposed Architecture

In order to enlarge the MB size, the extensions to H.264 (JM) should be made with a new definition of the added block modes for mb_type, the modification of the CBP definition and block indexing. Fig.2 shows a subsequent chain of the H.264 (JM) elements which are affected by enlarging the MB size. Enlarging the MB size entails the necessity of redefining block indexes of sub-blocks. Subsequently, motion estimation and mode decision are conducted based on the newly numbered block indexing. After the

mode decision is completed, entropy coding is performed. Finally deblocking filters with the block indexing modified are applied.

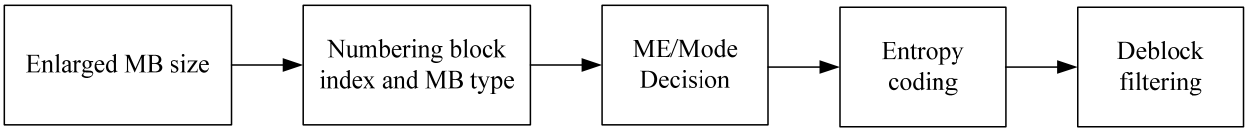


Fig. 2 H.264 (JM) elements affected by the enlarged MB's

Table 1 lists the H.264 elements that must be modified by enlarging the MB sizes. As aforementioned, this is not the exhaustive list. In Table 1, *N* indicates the maximum pixel size for the enlarged MB and can be increased with power of 2 such as 32, 64 or 128.

Table 1. H.264 elements to be modified for enlarged MB's

	H.264/AVC	Proposed enlarged MB
MB size	16x16	NxN
ME size	16x16 ~ 4x4	NxN ~ 4x4
Integer transform	4x4	4x4
CBP computation	8x8	(N/2xN/2), ···, 8x8
The number of SKIP mode	1	$\log_2 N - 3$
Blocking indexing	Fig.4	Fig.5
The number of mode for Inter	8	$4 \cdot \log_2 N - 8$

2.1 Hierarchical block partition structures for enlarged MB's

The proposed block modes with enlarged MB are depicted in Fig. 3. As shown in Fig.3, the MB mode partitions constitute a hierarchical structure in a top-down manner. For example, if the maximum MB size is set to 64x64, the resulting number of the partition levels is 4. The sub-block of 32x32 size constitutes the 2nd level hierarchy which can be partitioned further down to 16x16. Similarly, the 4th level hierarchy ends up with 8x8 block mode. Thus, for the case of 64x64 MB, 13 block types and 3 SKIP modes for 64x64, 32x32 and 16x16 block size can be possible. The number of block modes depends on the maximum MB size as shown in Table 1.

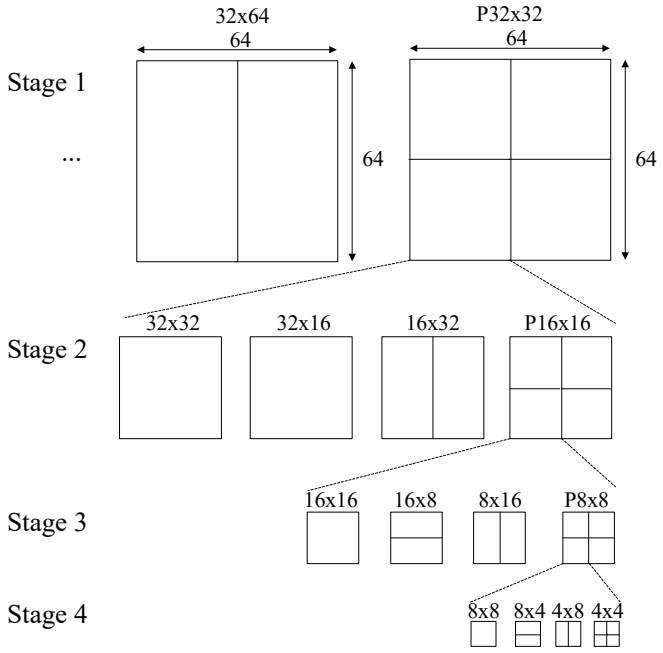


Fig. 3 An example of hierarchical block partitions for enlarged MB of 64x64

2.2 block modes for my_type

The mb_type in the MB syntax of the current H.264 is defined for the MB size of 16x16 and its partitions. The small numbers (0, 1) of mb_type are assigned for the large block modes (SKIP, Inter_16x16) while large numbers (5, 6, 7) are assigned for the block modes (Inter_8x4, Inter_4x8, Inter_4x4) of relatively small sizes. Similarly, in the proposed enlarged MB, the numbering of mb_type should be changed by assigning small numbers to the large block modes such that total bit amounts to represent mb_type become efficient. The mb_type according to the enlarged MB size is defined in Table 2.

Table 2. Possible partition block modes (mb_type) for enlarged MB's of 128x128, 64x64 and 32x32

128x128 MB size	SKIP_128x128(0), 128x128(1), 128x64(2), 64x128(3), SKIP_64x64(4), 64x64(5), 64x32(6), 32x64(7), SKIP_32x32(8), 32x32(9), 32x16(10), 16x32(11), SKIP_16x16(12), 16x16(13), 16x8(14), 8x16(15), 8x8(16), 8x4(17), 4x8(18), 4x4(19),
64x64 MB size	SKIP_64x64(0), 64x64(1), 64x32(2), 32x64(3), SKIP_ 32x32(4), 32x32(5), 32x16(6), 16x32(7), SKIP_16x16(8), 16x16(9), 16x8(10), 8x16(11), 8x8(12), 8x4(13), 4x8(14), 4x4(15),
32x32 MB size	SKIP_32x32(0), 32x32(1), 32x16(2), 16x32(3), SKIP_16x16(4), 16x16(5), 16x8(6), 8x16(7), 8x8(8), 8x4(9), 4x8(10), 4x4(11),

If an enlarged MB size is NxN, '0' is assigned to mb_type for the SKIP_NxN block mode. Hence, the largest number of mb_type is assigned to the smallest block size whose occurrence rate is the smallest among the other block modes.

2.3 Block indexing

As the MB size is enlarged, the block indexing for sub-MB and 4x4 integer transform should be modified. Fig.4 shows the block indexing for the MB of 16x16 size in the 4:2:0 YCbCr format. Once the sub-blocks are partitioned, 4x4 blocks are indexed inside one 8x8 block. However, for an enlarged MB size, the block indexing for the sub-MBs are changed accordingly. Fig. 5 shows an example of the block indexing for the MB size of 32x32. The block indexing for the enlarged MB is numbered hierarchically in a top-down manner. The encoding order according to the proposed block indexing is also shown in Fig.5.

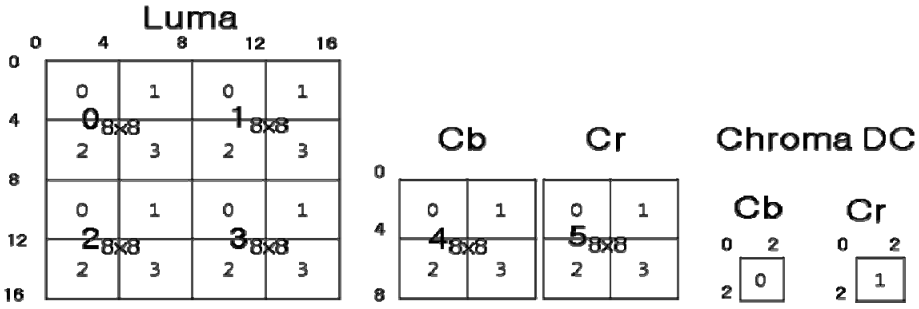


Fig. 4 Block indexing for 16x16 MB

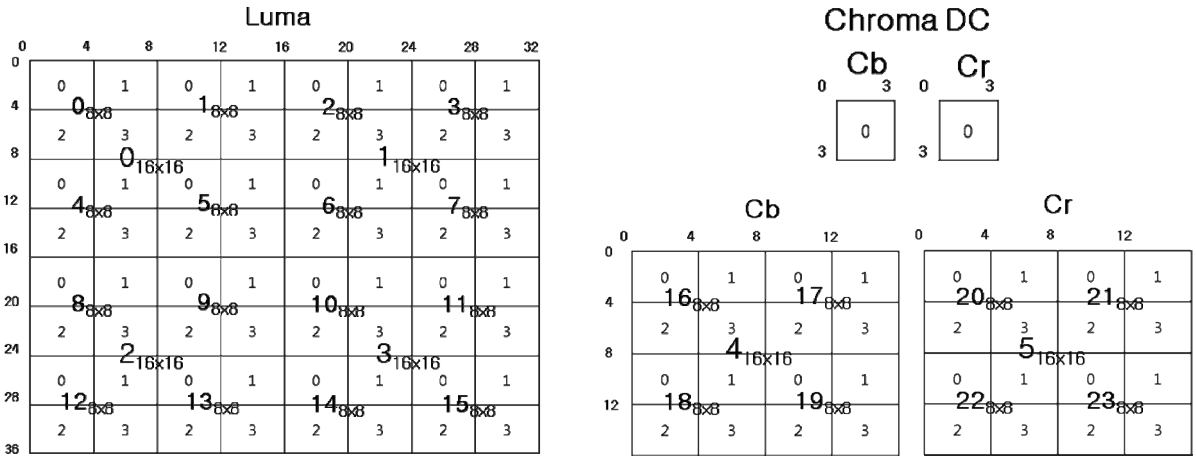


Fig. 5 Block indexing for 32x32 MB

2.4 New SKIP modes in each mode stage.

For the enlarged MB sizes, new SKIP modes are added to the block modes. The total number of SKIP modes in the current H.264 is only one for 16x16 block size while the number of SKIP mode in the enlarged MB depends on the MB size N. Since the block modes are constructed in a hierarchical tree structure, several SKIP modes may be added when the shapes of the block modes are square, for example, 32x32, 64x64, and 128x128 etc. The added SKIP modes can play an important role of enhancing the coding efficiency for high resolution video contents. By incorporating one SKIP mode into each squared block partition, the decoding syntax might be accordingly modified such that the SKIP mode is firstly checked at the beginning of parsing each squared block partition [3]. Fig. 6 shows an example of syntax parsing for enlarged MB with SKIP modes in each level of squared block modes.

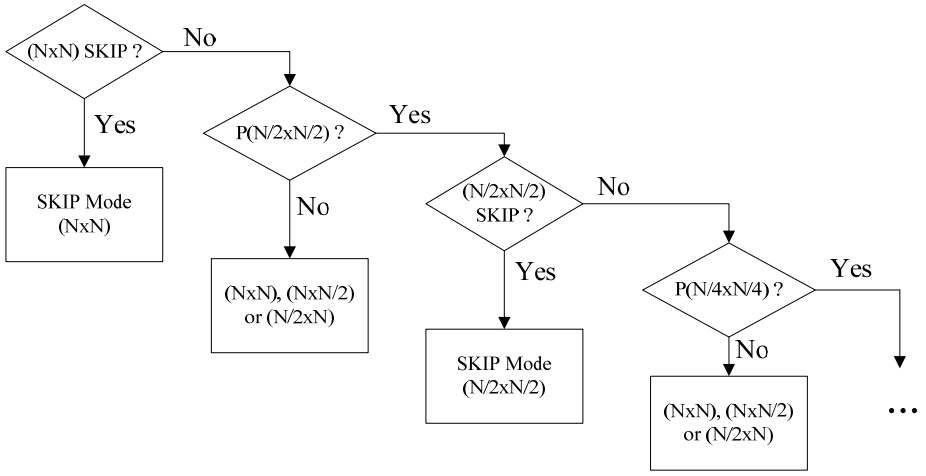


Fig. 6 MB syntax for the enlarged MB

2.5 The CBP value

The CBP value for the 16x16 MB can signal the information about all zero blocks based on 8x8 block in H.264. This is directly extended in a hierarchical manner to an enlarged MB where CBP_EMB (CBP for Enlarged MB) is newly defined to signal the context of quantized IT coefficients based on the NxN MB size. Fig. 7 shows an example of CBP_EMB for the enlarged MB of 32x32 size.

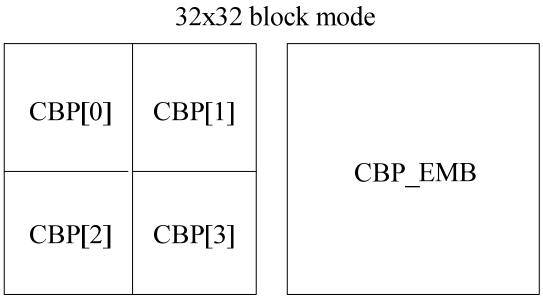


Fig. 7 CBP_EMB for 32x32 MB and CBPs for 16x16 blocks

2.6 Signaling issue for MB size information

An optimal way of signaling MB size information must be investigated by considering MB, Slice, SPS or PPS level.

3 Conclusion

We propose to extend the H.264 with an enlarged MB scheme for high resolution video coding such HD and beyond HD sequences. The statistics show that the enlarged MB can provide enhanced coding efficiency for high resolution video coding. This extension is definitely beneficial to the long-term living of H.264 for high resolution video coding beyond HD resolutions.

For enlarging MB size, the number of block partitions increases accordingly. This may cause to reduce the coding efficiency in terms of signaling the block mode information due to increase number of block partitions. Since the motivation of enlarging the MB size is targeted for high resolution video coding, the number of block modes can be reduced by eliminating the block partitions of small sizes which are rarely selected. This might be a reasonable approach because the interoperability between a very small resolution (QCIF) video and a very large resolution (Ultra HD) video may not be necessary in a single video codec with their application domains far separated.

4 References

[1] S. Naito and A. Koike, “Efficient coding scheme for super high definition video based on extending h.264 high profile,″ Visual Communications and Image Processing (VCIP) 2006, Jan 2006, San Jose, CA, USA

[2] Siwei Ma and C.-C. Jay Kuo, “High-definition Video Coding with Super-macroblocks,” Visual Communications and Image Processing 2007, Sep 16-19, 2007, San Antonio, USA,

[3] Jae-Chool Lee, Jianle Chen, Kyo-Hyuk Lee, and Woo-Jin Han, “Hierarchical Block Mode Structure Extension for Video Coding,” Conference 2008 on the Korean Society of Broadcast Engineers, p73-74, 2008

[4] Shun-ichi Sekiguchi, Shuichi Yamagishi, Yusuke Itani, Kohtaro Asai, Tokumichi Murakami., “On motion block size and partition for 4:4:4 video coding”, VCEG-AI31, July 2008, Berlin, Germany