

# Joint motion vector encoding scheme with a pooled macroblock type

Bongsoo Jung and Byeungwoo Jeon, MEMBER SPIE

Sungkyunkwan University, Department of Electronic and Electrical Engineering, 300 Chunchun-Dong, Jangan-Gu, Suwon 440-746, Korea

**Abstract.** Motion vectors correlate very well with other neighboring motion vectors. Thus, many macroblocks have zero residual motion vectors within their blocks after differential pulse coded modulation using their individually predicted motion vectors. Motivated by this observation, we develop a new joint encoding scheme of motion vectors by defining a new macroblock coding mode called *pooled zero motion vector difference coding* to jointly code such cases more efficiently. Experimental results with several well-known video test sequences verify that the proposed method improves the coding efficiency up to 6.2% compared to the H.264|advanced video coding (AVC). © 2008 Society of Photo-Optical Instrumentation Engineers.  
[DOI: 10.1117/1.3046713]

Subject terms: motion vector coding; pooled coding scheme; H.264; MPEG-4 Part 10 AVC.

Paper 080635LR received Aug. 11, 2008; revised manuscript received Oct. 29, 2008; accepted for publication Nov. 4, 2008; published online Dec. 24, 2008.

## 1 Introduction

In the H.264|advanced video coding (AVC) standard, the motion-compensation process selects the best motion-compensation block size among many different sizes<sup>1</sup> ranging between  $16 \times 16$  and  $4 \times 4$ . To signal the selected block size, the *inter\_coding\_mode* defines and signals the following macroblock types: SKIP,  $P16 \times 16$ ,  $P16 \times 8$ ,  $P8 \times 16$ , and

$P8 \times 8$ . The *inter\_coding\_mode* of  $P8 \times 8$  can be further divided into  $P8 \times 4$ ,  $P4 \times 8$ , and  $P4 \times 4$  in a hierarchical fashion. The Joint Model (JM) H.264|AVC encoder<sup>2,3</sup> estimates motion vectors for each block size and selects the one that performs best in rate-distortion optimization. This fine block structure brings higher coding efficiency, but it generates up to 16 motion vectors, which means that there are more data to encode for motion vector information. In general, decreasing the block size for motion compensation also decreases the prediction error. However, small motion-compensation block sizes are not frequently decided by rate-distortion optimization because of increased motion vector data. The current method for coding motion vectors by the H.264|AVC is still inefficient because it uses at least two bits to signal the zero residual motion vector, even if it is the same as its predicted motion vector. Furthermore, the zero residual motion vectors are encoded individually.

We propose a new joint motion vector coding scheme that can represent the case very efficiently when the motion

vector of each block is the same as the individual predicted motion vector. This is achieved by pooled representation of the zero residual motion vectors using a new *macroblock\_type* called PallzeroDMV mode.

## 2 Proposed Method

The compressed bitstream consists mainly of three kinds of information: texture information, motion information, and other information, such as headers. Usually, the texture information that represents transformed coefficients of pixel data takes up most of the compressed bits. However, the bit amount of motion information is not negligible unless the motion is extremely simple or the texture is very complex with low quantization parameter (QP).

Figure 1 shows the relative bit portions of motion information in total compressed bits with four different QP values. The sequence *Foreman* has a lot of motion, so it requires a large number of bits to represent that motion, and its relative portion increases as the bit rate decreases. When QP=40, about 47% of the total bits represent the motion information. The motion vectors of H.264|AVC are coded differentially with respect to their motion vector predictors, which are the medians of three motion vectors of the spatially adjacent causal blocks, as shown in Fig. 2.

H.264|AVC does differential motion vector coding. The motion vector residual **mvd** is calculated as:

$$\mathbf{mvd} = \mathbf{mv}_E - \mathbf{pmv}, \quad (1)$$

$$\mathbf{pmv} = \text{median}(\mathbf{mv}_A, \mathbf{mv}_B, \mathbf{mv}_C),$$

where  $\mathbf{mv}_E$  is the motion vector of the current block E, and  $\mathbf{pmv}$  is the median of its neighboring motion vectors, as shown in Fig. 2. If block C is not available when computing Eq. (1), then it is replaced by block D. Since motion vectors correlate closely with their spatial neighbors, more often than not the **mvd** is a zero vector. To maximize the coding efficiency of motion vectors, it is critical to effectively encode the cases that result in a zero vector of **mvd** because it happens very frequently. The current standard in H.264|AVC in this regard is to use 2 bits per block (one bit to indicate a zero value of the *x* component of **mvd**, and the other one bit for its *y* component).

In the JM H.264|AVC reference software,<sup>4</sup> motion vector selection is achieved by minimizing the rate-distortion optimization criterion:

$$J = D + \lambda R, \quad (2)$$

$$R = R_{mv} + R_{ref},$$

where  $R_{mv}$  is the rate of the motion vector difference (i.e., **mvd**), and  $R_{ref}$  is the rate of the reference frame index. The macroblock mode is decided by minimizing the rate-distortion cost among all block partitions ( $16 \times 16, \dots, 4 \times 4$ ).

We note that in low bit-rate applications, motion vector data may take a more dominant bit portion than texture data. In this case, the inefficient motion vector coding will lead to significantly lower coding efficiency. In the H.264|AVC standard, the DIRECT mode coding in B frames need not transmit the motion vector because the decoder can infer it by using the motion vectors of previ-

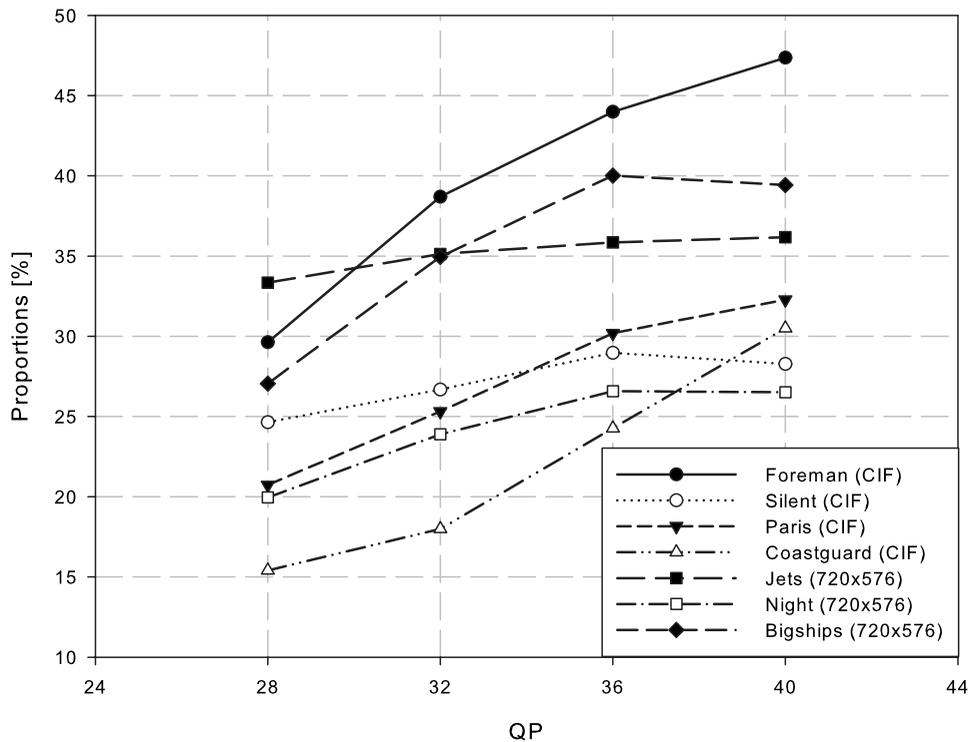


Fig. 1 Bit proportion of motion vector data in compressed bitstream (H.264|AVC@baseline).

ously decoded macroblocks. Similarly, the P\_SKIP mode in P frames sends neither the motion vector nor residual texture information. The motion vector is inferred by the median motion vector predictor of the  $16 \times 16$  macroblock in the same way. This is possible because the motion vector of the current macroblock correlates well with those of neighboring macroblocks. These ideas inspired us to develop our new motion vector coding method based on a pooled representation of zero residual motion vectors. In sixteen  $4 \times 4$  motion compensation blocks, we make a pooled representation of the 16 zero **mvds**. The cases of macroblock segmentation in the smallest block size happen frequently only if they can override the rate burden of motion vector data since smaller block size for motion compensation brings small prediction error. We signal the pooled **mvd** by defining a new macroblock\_type mode in P(Predictive) slices. This mode is called PallzeroDMV, and it indicates that the following conditions are jointly satisfied:

- The block size for motion compensation is  $4 \times 4$ .
- The reference frame is the one closest in the reference frame memory to the current picture
- **mvd**=0 for each of the 16 blocks in the macroblock.

For the PallzeroDMV mode, the rate-distortion criterion is proposed as follows:

$$J_{PallzeroDMV} = D_{PallzeroDMV} + \lambda R_{PallzeroDMV}, \quad (3)$$

$$R_{PallzeroDMV} = R_{mv} + R_{ref} = 0,$$

where  $D_{PallzeroDMV}$  is the distortion introduced by all sixteen  $4 \times 4$  motion compensation blocks in the PallzeroDMV mode. Here,  $R_{mv}$  and  $R_{ref}$  have zero rate, since PallzeroDMV mode does not send a motion vector nor a reference frame index to the decoder.

The proposed PallzeroDMV mode is easily accommodated in the H.264 encoder by adding the additional mode, as shown in Table 1. If all sixteen  $4 \times 4$  block motion vectors are equal to each predicted motion vector, then the total number of bits in H.264|AVC to represent all motion vectors is 32 bits. However, by implementing the proposed macroblock mode PallzeroDMV, it can be represented using only 3 bits for the macroblock\_type. This is possible because the zero differential motion vectors are pooled and jointly represented by only one macroblock\_type. Under the PallzeroDMV mode, a decoder can infer the 16 motion vectors as individual predicted motion vectors (PMVs), and the reference index as zero. Note that the proposed PallzeroDMV mode coding does not require any extra motion search to calculate the rate-distortion cost of the PallzeroDMV mode.

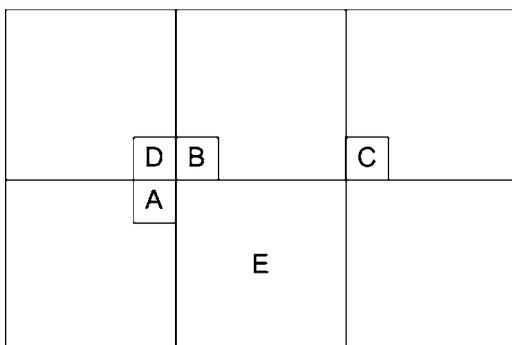


Fig. 2 Motion vector prediction.

**Table 1** Proposed *inter\_coding\_mode* of macroblock in P-slices.

<i>mb_type</i>	Code	No. of bits	Name of <i>mb_type</i>
0	"1"	1	P16×16
1	"010"	3	PallzeroDMV
2	"011"	3	P16×8
3	"00100"	5	P8×16
4	"00101"	5	P8×8
5	"00110"	5	P8×8ref0
Inferred	Run coding	—	P_SKIP

### 3 Experimental Results

To evaluate the performance of the proposed pooled coding scheme, we implemented it on the JM12.3 encoder reference software.<sup>3</sup> In the simulation, we considered the baseline profile; three reference frames; context adaptive variable length coding entropy coding; the IPPP coding structure; full motion search with ±32 search range; QP values of 28, 32, 36, 40; and a total encoding of 150 frames. We used *Foreman*, *Coastguard*, *Paris*, and *Silent* sequences of common intermediate format (CIF) size, *Jets*, *Night*, and *Bigships* sequences of 720×576 standard definition (SD) size; and *Night*, *ShuttleStart*, and *Crew* sequences of 720-p high definition (HD) size. The coding efficiency of the proposed method is compared with that of the H.264|AVC encoder under the identical conditions. For a numerical comparison, we use the Bjontegaard Delta bit rate (BDBR) and Bjontegaard Delta peak signal-to-noise ratio (PSNR) (BDPSNR).<sup>5</sup> The plus sign of BDPSNR and minus sign of BDBR indicate better PSNR and better bit rate reduction of the proposed method compared to those of its anchor case, respectively.

Table 2 shows that the proposed motion vector coding method is better than that of the current H.264|AVC by an average of 3.83% in BDBR and 0.13 dB in BDPSNR. In particular, the proposed method has much higher coding efficiency than that of H.264|AVC baseline in the *Foreman* and the *ShuttleStart* sequences. Based on the bit portion statistics in Fig. 1, this is somewhat expected—the *Foreman* has a lot of motion, and its motion information consumes a relatively large portion of the compressed bit budget. Therefore, a more efficient motion vector coding method has made a distinctive difference. These results verify that our proposed motion vector encoding approach using the new pooled zero differential motion vector coding is more efficient than the H.264|AVC video coding standard.

**Table 2** Coding performance of proposed method.

Sequence	BDPSNR (dB)	BDBR (%)
<i>Foreman</i> (CIF, 30 fps)	0.28	-5.91
<i>Coastguard</i> (CIF, 30 fps)	0.09	-2.58
<i>Paris</i> (CIF, 15 fps)	0.08	-1.53
<i>Silent</i> (CIF, 15 fps)	0.17	-3.99
<i>Jets</i> (SD, 30 fps)	0.20	-4.33
<i>Night</i> (SD, 30 fps)	0.13	-3.09
<i>Bigships</i> (SD, 30 fps)	0.18	-4.52
<i>Night</i> (HD, 30 fps)	0.15	-3.56
<i>ShuttleStart</i> (HD, 30 fps)	0.19	-6.16
<i>Crew</i> (HD, 30 fps)	0.10	-2.64
<i>Average</i>	0.15	-3.83

### 4 Conclusion

We propose a new motion vector coding method using pooled zero differential motion vector representation. It is realized by defining a new macroblock\_type called the PallzeroDMV mode. Experimental results show that the proposed coding method achieves a BDBR gain of 1.53% ~ 6.16% with no increase in complexity.

### Acknowledgments

This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) NRL program grant funded by the Korean government [MEST; ROA-2006-000-10826-0(2008)].

### References

1. Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264|ISO/IEC 14496-10 AVC)," Document JVT-G050r1 (2003).
2. T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 688–703 (2003).
3. JM12.3 Test Model CODEC ISO/IEC MPEG and ITU-T VCEG Joint Video Team (Online). Available: [http://bs.hhi.de/suehring/tml/download/old\\_jm/jm12.3.zip](http://bs.hhi.de/suehring/tml/download/old_jm/jm12.3.zip).
4. K.-P. Lim, G. Sullivan, and T. Wiegand, "Text description of joint model reference encoding methods and decoding concealment methods," ISO/IEC MPEG and ITU-T VCEG Joint Video Team, Document JVT-N046 (2005).
5. G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ISO/IEC MPEG and ITU-T VCEG Joint Video Team, Document VCEG-M33, Austin, TX (2001).