Journal of Marine Science and Technology

Volume 23 | Issue 5 Article 16

MOTION VECTOR PREDICTION USING RESELECTION AMONG CANDIDATE SET

S. T. Uou
Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.

Eric Y. T. Juan
Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.

Jim Z. C. Lai
Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C., zclai@mail.ntou.edu.tw

Recommended Citation

Uou, S. T.; Juan, Eric Y. T.; and Lai, Jim Z. C. (2015) "MOTION VECTOR PREDICTION USING RESELECTION AMONG CANDIDATE SET," *Journal of Marine Science and Technology*: Vol. 23: Iss. 5, Article 16.
DOI: 10.6119/JMST-015-0604-2
Available at: https://jmstt.ntou.edu.tw/journal/vol23/iss5/16

Journal of Marine Science and Technology, Vol. 23, No. 5, pp. 727-731 (2015)
DOI: 10.6119/JMST-015-0604-2

MOTION VECTOR PREDICTION USING RESELECTION AMONG CANDIDATE SET

S. T. Uou, Eric Y. T. Juan, and Jim Z. C. Lai

Key words: H.264, MPEG-4 AVC, motion vector prediction.

ABSTRACT

For video coding standards with high compression ratio and good image quality, prediction is an essential step. If the prediction is more accurate, then lower bit-rate will be achieved. In this paper, a method is proposed to improve the coding efficiency for motion vectors using predicted motion vector candidate set (PMVCS). PMVCS consists of the motion vectors of blocks, which are the spatial and temporal neighbors of an encoding block. Using the proposed method, a better predicted motion vector can be obtained, which will result in fewer bits allocated for motion vectors. Compared with JIM18.4, the proposed method can reduce the bit allocation for motion vectors by 1.37% in average using the image sequence "Bus." The proposed method can reduce the bit allocation for motion vector of error correction technique by about 0.31% using the same image sequence.

I. INTRODUCTION

Video coding consists of encoding and decoding. Video encoding is the process of reducing data to benefit storage and transmission. Video decoding is the operation which recovers an image sequence from the encoded or compressed data stream. Because the video data of high definition requires a very large amount of storage space, a high compression video coding is desired. H.264/AVC (ITU-T and ISO/IEC 2003; Wiegand et al., 2003; Xiao et al., 2005) is a popular format for HD television, DVD/Blue-Ray, and internet streaming media. Comparing with MPEG-II, H.264/AVC can provide high quality videos with lower bandwidth.

H.264/AVC is a block oriented motion compensation based video compression standard developed by the Joint Video Team (JVT), which consists of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC JTC1 Moving Picture Experts Group (MPEG). Therefore, the ITU-T H.264 standard and the ISO/IEC MPEG-4 Advanced Video Coding (AVC) standard have the same technology which is jointly maintained by ITU and MPEG.

In H.264/AVC, motion information requires a significant portion of encoded bit stream, particularly in the case of low bit rates. Therefore, a good prediction of motion vector is desired. This paper presents a method to improve the prediction of motion vectors for H.264/AVC. To improve prediction, we will modify the error correction technique presented by Yang et al. (2010). This paper is organized as follows. Section 2 describes H.264/AVC. Section 3 presents the method proposed in this paper. Section 4 shows the experimental results and section 5 presents the concluding remarks.

1. H.264/AVC

H.264/AVC is a format for video compression. It consists of the video encoder and decoder. The H.264/AVC encoder transfer image sequences into bit streams and it includes prediction, transformation and entropy encoding. The H.264/AVC decoder inverts encoded data into original video signal and it consists of entropy decoding, inverse transformation and reconstruction.

For the encoder, the predictive values are obtained by motion estimation (ME) and motion compensation (MC). The residual value is obtained by subtracting the predictive value from the current one. After residual values pass through integer transform and quantization (Q), the encoder performs entropy encoding to obtain the encoded bit stream. For the decoder, the encoded bit stream is gone through inverse quantization (IQ) and inverse integer transform to produce the residual values. The predictive values are also generated by the decoder side. The original image is obtained by adding the residual values to the predictive ones.

2. Prediction

In H.264/AVC, a frame can be classified into either an intra frame or inter frame. Each frame is further divided into a set of macroblocks (MBs), where a MB is a region of 16 × 16 pixels. For the intra frame, a MB can be encoded using the information from the same frame as shown in Fig. 1. For the inter frame, a MB is compressed using the information from current and previous encoded frames. The inter prediction finds the motion vector which is the best matched block from

Paper submitted 05/27/14; revised 10/31/14; accepted 06/04/15. Author for correspondence: Jim Z. C. Lai (e-mail: zclai@mail.ntou.edu.tw).
Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.

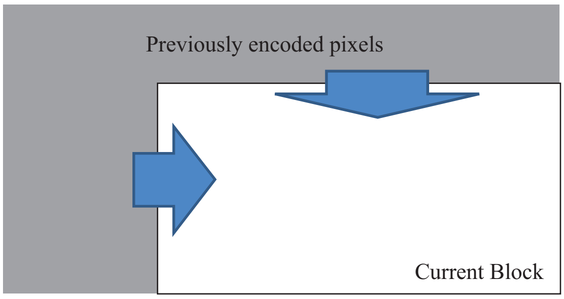

Fig. 1. Inra prediction.

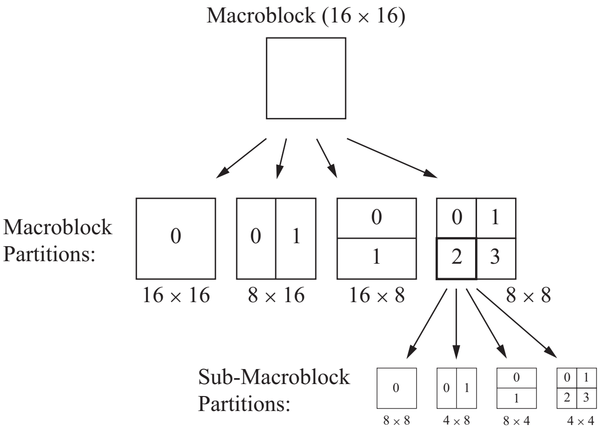

Fig. 2. Block partitions.

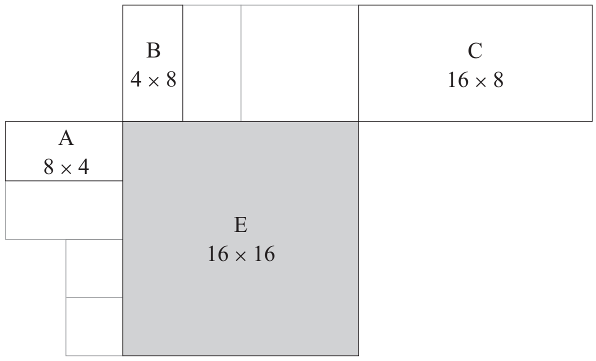

Fig. 3. Current and neighboring partitions with different partition sizes.

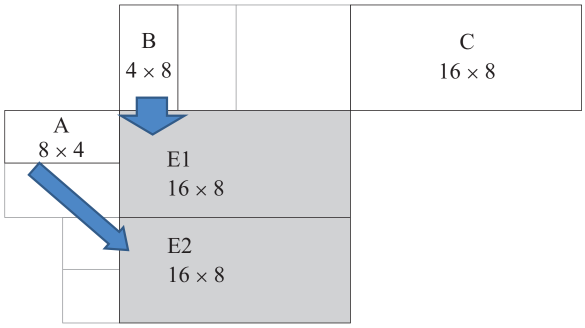

Fig. 4. Case (2) of ADMV.

the reference frames for the current block.

To improve the efficiency of prediction, a MB is split to some smaller partitions: 16 × 16, 16 × 8, 8 × 16, 8 × 8, 8 × 4, 4 × 8 and 4 × 4 as shown in Fig. 2. There are three types of MBs: I MB, P MB and B MB. The I MB uses the intra prediction from the neighbors in the current frame. The P MB is predicted from previously encoded frames which are before the current frame. The B MB is like the P MB and its reference frames can be from before or after the current frame in the display order.

Inter prediction uses the temporal correlation between a frame and its neighboring encoded frames. The offset between the location of the current block and the predicted block in the reference frame is called predicted motion vector, which is denoted as PMV. The offset between the location of the current MB and the best matched block in the reference frames is called motion vector, which is denoted as MV. The best matched block has the minimum SAD (sum of absolute difference) between the current block and the corresponding partition in reference frames. The difference between the PMV and MV is called motion vector difference, which is denoted as MVD. MVD is encoded using entropy coding.

Determining PMV depends on the partition size and the availability of the motion vectors of neighboring encoded blocks. Let E be the current macroblock/sub-macroblock; A, B and C are, respectively, the left, top and top-right macroblock/sub-macroblock adjacent to E as shown in Fig. 3. The algorithm of determining predicted motion vector is referred to as ADPMV and is presented as follows:

ADPMV

(1) For the current partition excluding 16 × 8 and 8 × 16, set PMV = the median of motion vectors of A, B and C.
(2) For the 16 × 8 partitions, set PMV of the upper partition = MV of B and the PMV of the lower partition = MV of A as shown in Fig. 4.
(3) For the 8 × 16 partitions, set PMV of the left partition = MV of A and PMV of the right partition = MV of C.
(4) For a skipped MB, the PMV is generated as in case (1). This MB is encoded in 16 × 16 inter mode.

II. PROPOSED METHODS

1. PMV and MVD

PMV is predicted from the MVs of neighboring blocks by the encoder and decoder. The MVD is the difference between the PMV and MV of the current block. MVD instead of MV is sent to the encoder to be encoded. If the PMV is predicted accurately, a small MVD will be obtained and encoded using fewer bits. The goal of the proposed method is to minimize MVD such that fewer bits are allocated to encode MVD. The proposed method is added to inter prediction in H.264/AVC standard to predict motion vectors.

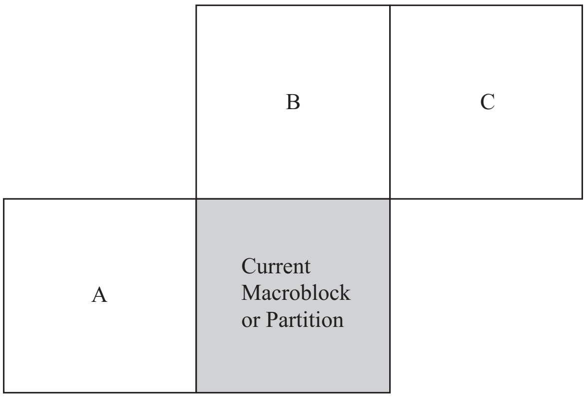

Fig. 5. Neighbors of current block.

2. Candidate Set of Motion Vectors

In an image sequence, there is a highly spatial and temporal correlation between motions of neighboring blocks. Therefore, the spatial correlation is used to determine PMV in H.264/AVC. If the current block and any of neighboring blocks do not belong to the same object, the predicted motion vector using the ADPMV algorithm can be different significantly from the motion vector of the current block. To solve this problem, the proposed method uses alternative temporal or spatial correlated motion vectors to determine the better PMV. The proposed method creates a predicted motion vector candidate set (PMVCS) for PMV as shown below:

$$\text{PMVCS} = \{\mathbf{mv}_{\text{H.264}}, \mathbf{mv}_{pre}, \mathbf{mv}_A, \mathbf{mv}_B, \mathbf{mv}_C, \mathbf{mv}_{(0,0)}\} \quad (1)$$

where $\text{mv}_{\text{H.264}}$ is the PMV determined using ADPMV adopted in the H.264/AVC standard; mv_{pre} is the MV of a block, in the previous frame, having the same block position as the current one; mv_A, mv_B, and mv_C, respectively, are the motion vectors of blocks A, B and C in the current frame as shown in Fig. 5; and $\text{mv}_{(0,0)} = (0, 0)$.

In PMVCS, $\mathbf{mv}_{pre}$ is adopted due to the temporal correlation; $\mathbf{mv}_A$, $\mathbf{mv}_B$ and $\mathbf{mv}_C$ are inserted for using the information of spatial correlation. As shown in (Laroche et al., 2008), when block A and the current block are belong to the same object, $\mathbf{mv}_A$ will be the better PMV. Similarly, if block B/C and the current block are belong to the same object, $\mathbf{mv}_B/\mathbf{mv}_C$ will be the better PMV. In the case that the current is part of static background, $\mathbf{mv}_{(0,0)}$ is obviously a better selection for PMV.

Instead of using $\mathbf{mv}_{\text{H.264}}$ as the PMV, the proposed method determines PMV from the candidate set PMVCS. Let MV, PMV and DMV of the current block be denoted as (mv_x, mv_y), (pmv_x, pmv_y), and (dmv_x, dmv_y), respectively. Suppose that $\mathbf{mv}_{\text{H.264}} = (tmv_x, tmv_y)$. Let the set $TS = \{\mathbf{mv}_{pre}, \mathbf{mv}_A, \mathbf{mv}_B\}$. For each current block, denote *xcount* as the number of motion vectors, in *TS*, whose *x* component value is larger than the *y* component value. Note here that $xcount \le 3$. If $xcount < 2$, the proposed method sets $pmv_x = tmv_x$ and $dmv_x = mv_x - pmv_x$ and determines a candidate motion vector $\mathbf{CVT} = (cmv_x, cmv_y)$ from PMVCS such that $|cmv_x - mv_x|$ is minimum. At this step, we can set $dmv_y = mv_y - cmv_y$. If $xcount \ge 2$, the proposed method sets $pmv_y = tmv_y$ and $dmv_y = mv_y - pmv_y$. The candidate motion vector $\mathbf{CVT} = (cmv_x, cmv_y)$ is selected from PMVCS such that $|cmv_y - mv_y|$ is minimum and set $dmv_x = mv_x - cmv_x$.

Now, we would like to present the modified ADPMV (MADPMV) for the encoder and decoder as follows:

MADPMV (for the encoder)

(1) For a current block: determine mv_{pre}, mv_A, mv_B, and mv_C; use ADPMV to find $\text{mv}_{\text{H.264}}$; and set PMVCS = $\{\text{mv}_{\text{H.264}}, \text{mv}_{pre}, \text{mv}_A, \text{mv}_B, \text{mv}_C, \text{mv}_{(0,0)}\}$.
(2) For a current block:
  (2.1) Set $TS = \{\text{mv}_{pre}, \text{mv}_A, \text{mv}_B\}$.
  (2.2) Determine *xcount*, where *xcount* is number of motion vectors, in *TS*, whose *x* component value is larger than or equal to the y component value.
  (2.3) If $xcount < 2$: (a) set $pmv_x = tmv_x$ and $dmv_x = mv_x - pmv_x$, where tmv_x is the *x* component of $\text{mv}_{\text{H.264}}$; (b) find a candidate motion vector CVT = (cmv_x, cmv_y) from PMVCS such that $|cmv_x - mv_x|$ is minimum; (c) set $dmv_y = mv_y - cmv_y$.
  (2.4) If $xcount \ge 2$: (a) set $pmv_y = tmv_y$ and $dmv_y = mv_y - pmv_y$, where tmv_y is the y component of $\text{mv}_{\text{H.264}}$; (b) find a candidate motion vector CVT = (cmv_x, cmv_y) from PMVCS such that $|cmv_y - mv_y|$ is minimum; (c) set $dmv_x = mv_x - cmv_x$.

MADPMV (for decoder)

(1) For a current block: determine mv_{pre}, mv_A, mv_B, and mv_C; use ADPMV to find $\text{mv}_{\text{H.264}}$; and set PMVCS = $\{\text{mv}_{\text{H.264}}, \text{mv}_{pre}, \text{mv}_A, \text{mv}_B, \text{mv}_C, \text{mv}_{(0,0)}\}$.
(2) For a current block:
  (2.1) Set $TS = \{\text{mv}_{pre}, \text{mv}_A, \text{mv}_B\}$.
  (2.2) Determine *xcount*, where *xcount* is number of motion vectors, in *TS*, whose *x* component value is larger than or equal to the *y* component value.
  (2.3) If $xcount < 2$: (a) set $pmv_x = tmv_x$ and $mv_x = dmv_x + pmv_x$; (b) find a candidate motion vector CVT = (cmv_x, cmv_y) from PMVCS such that $|cmv_x - mv_x|$ is minimum; (c) set $mv_y = dmv_y + cmv_y$.
  (2.4) If $xcount \ge 2$: (a) set $pmv_y = tmv_y$ and $mv_y = dmv_y + pmv_y$; (b) find a candidate motion vector CVT = (cmv_x, cmv_y) from PMVCS such that $|cmv_y - cmv_y|$ is minimum; (c) set $mv_x = dmv_x + cmv_x$.

When the motion vectors of neighboring blocks are not available, set zero motion vectors as their corresponding motion vectors for the proposed method: MADPMV. In this case, we will obtain $\mathbf{CVT} = \text{mv}_{\text{H.264}} = \text{mv}_{(0,0)}$ and obtain $dmv_x = dmv_y = 0$.

III. EXPERIMERNTAL RESULTS

In this section, we will compare the proposed method with the error correction technique (Yang et al., 2010) and the

Table 1. Hardware and software environment.

Software	JM 18.4
CPU	Intel Core i7-2600 @3.40GHz
Ram	8GB
OS	Windows 7 SP1
Compiler	Microsoft Visual Studio 2010

Table 2. Simulation conditions.

Profile	66 (baseline)
Prediction structure	IPPP
Frame rate	30
Reference frames	5
Search range	32
RDO	On
Quantization parameter (QP)	28, 32, 36, 40

Table 3. Reduction of bits allocated for motion vectors using the image sequence "Bus".

QP	Error Correction Technique	Proposed Method
28	1.03%	1.30%
32	1.15%	1.70%
36	0.91%	1.16%
40	1.14%	1.32%
average	1.06%	1.37%

original JM18.4, which is based on H.264/AVC key technical area (KTA) standard. Four tested CIF image sequences: "Bus," "Tempete," "Mobile," and "Highway" are used in the experiments. These four image sequences consist of 150 frames. Tables 1 and 2 indicate the hardware/software environment and the main simulation conditions, respectively.

The percentages of reduction of bits allocated for motion vectors using the proposed method and error correction technique (Yang et al., 2010), compared with the original JM18.4 for various QP (quantization parameter) values are shown Table 3, when the image sequence "Bus" is used. Table 3 shows that compared with JM18.4, the proposed method and error correction technique can obtain fewer bits allocated for motion vectors due to the image sequence "Bus" has many horizontal or vertical motions. Compared with JIM18.4, the proposed method can reduce the bit allocation for motion vectors by 1.37% in average. The proposed method can reduce the bit allocation for motion vector of error correction technique by about 0.31%.

Table 4 presents the percentages of bit allocation for motion vectors using the proposed method and error correction technique for the image sequence "Tempete," which consists of many complex motions. Table 4 shows that compared with JM18.4, the proposed method and error correction technique can still obtain better predicted motion vectors to allocate few bits for motion vectors in the case of complex motions.

Table 4. Reduction of bits allocated for motion vectors using the image sequence "Tempete".

QP	Error Correction Technique	Proposed Method
28	0.82%	0.55%
32	0.41%	0.35%
36	0.60%	0.70%
40	0.61%	0.01%
average	0.61%	0.41%

Table 5. Reduction of bits allocated for motion vectors using the image sequence "Mobile".

QP	Error Correction Technique	Proposed Method
28	1.56%	1.70%
32	1.10%	0.95%
36	1.48%	1.54%
40	0.24%	0.67%
average	1.10%	1.22%

Table 6. Reduction of bits allocated for motion vectors using the image sequence "Highway".

QP	Error Correction Technique	Proposed Method
28	0.05%	1.65%
32	-0.23%	0.77%
36	0.72%	-0.29%
40	-1.06%	-1.19%
average	-0.13%	0.24%

Table 4 indicates that proposed method is more effective in the case of lower quantization parameter (QP) value. That is, the proposed method is better when a better image quality with higher bit-rate is desired.

Table 5 presents the reductions of bit allocation for motion vectors of the proposed method and error correction technique, compared with JM18.4, using the image sequence "Mobile". The image sequence "Mobile" has many horizontal or vertical motions. From Table 5, we can find that compared with JM18.4 or error correction technique, the proposed method can obtain the better predicted motion vectors, since fewer bits are allocated for motion vectors. Compared with error correction technique, the proposed method can reduce bit allocations for motion vectors by 0.12% in average. The proposed method can reduce allocated bits for motion vectors of JM18.4 by 1.22% in average.

Table 6 gives the reductions of bit allocations for motion vectors of the proposed method and error correction technique, compared with JM18.4, using the image sequence "Highway." This image sequence has complex motions. For some QP values, the proposed method or error correction technique may increase allocated bits for motion vectors, compared with JM18.4. However, compared with JM18.4 or error correction technique, the proposed method can still obtain few bits, in

Table 7. Average reduction of bits allocated for motion vectors using QCIF image sequences.

Sequence	Average bit rate
Bus	1.28%
Tempete	0.38%
Mobile	1.19%
Highway	0.21%

average, allocated for motion vectors, which implies the proposed method can obtain better predicted motion vectors.

Table 7 presents the average reduction of bits allocated for motion vectors for the proposed method, compared with the KTA standard, using image sequences of different resolution (QCIF). From Table 7, we can that the proposed method obtains a better performance, in terms of bits allocated to motion vectors, for QCIF image sequences; although the improvement is reduced a little bit. This is due to a larger portion of QCIF image sequences are boundary blocks and motion vectors of neighboring blocks for these blocks are not available.

IV. CONCLUSIONS

The proposed method uses a predicted motion vector candidate set (PMVCS) to obtain better predicted motion vectors. PMVCS consists of motion vectors having high spatial or temporal correlations to the motion vector of the current one. The better predicted motion vectors will result in fewer bit allocations. Compared with JM18.4 or error correction technique, the proposed method can provide fewer bits allocated for motion vectors using an image sequence having many horizontal or vertical motions. For an image sequence consisting complex motion, the proposed method can still obtain the better prediction vectors compared with JM18.4 or error correction technique. Using the image sequence "Bus" with QP = 32, the proposed method can reduce the bits allocated for motion vectors of JM18.4 and error correction technique by 1.70% and 0.55%, respectively.

REFERENCES

Joint Video Team of ITU-T and ISO/IEC JTC 1 (2003). Draft ITU recommendation and final draft international standard of Joint Video Specification (ITU-TRec. H.264 ISO/IEC 14496-10 AVC). Document JVT-G050r1.

Laroche, G., J. Jung and B. Pesquent-Popescu (2008). RD optimized coding for motion vector predictor selection. IEEE Transaction on Circuits and Systems for Video Technology 18, 1247-1257.

Wiegand, T., G. J. Sullivan, G. Bjontegaard and A. Luthra (2003). Overview of the H.264/AVC video coding standard. IEEE Transactions on Circuits and Systems for Video Technology 13, 560-576.

Xiao L., S. Rahardja, K. P. Lim, Z. G. Li, D. Wu and S. Wu (2005). Fast mode decision algorithm for intra prediction in H.264/AVC video coding. IEEE Transactions on Circuits and Systems for Video Technology 15, 813-822.

Yang W. et al. (2010). An efficient motion vector coding algorithm based on adaptive predictor selection. Proc. IEEE Int. Symp. Circuits and Systems, 2175-2178.