

Journal of Electronic Imaging

SPIEDigitalLibrary.org/jei

Adaptive search range adjustment and multiframe selection algorithm for motion estimation in H.264/AVC

Yingzhe Liu
Jinxiang Wang
Fangfa Fu



Adaptive search range adjustment and multiframe selection algorithm for motion estimation in H.264/AVC

Yingzhe Liu
Jinxiang Wang
Fangfa Fu

Microelectronics Center
Harbin Institute of Technology
Harbin 150001, China
E-mail: liuyingzhe616@163.com

Abstract. *The H.264/AVC video standard adopts a fixed search range (SR) and fixed reference frame (RF) for motion estimation. These fixed settings result in a heavy computational load in the video encoder. We propose a dynamic SR and multiframe selection algorithm to improve the computational efficiency of motion estimation. By exploiting the relationship between the predicted motion vector and the SR size, we develop an adaptive SR adjustment algorithm. We also design a RF selection scheme based on the correlation between the different block sizes of the macroblock. Experimental results show that our algorithm can significantly reduce the computational complexity of motion estimation compared with the JM15.1 reference software, with a negligible decrease in peak signal-to-noise ratio and a slight increase in bit rate. Our algorithm also outperforms existing methods in terms of its low complexity and high coding quality. © The Authors. Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.JEI.22.2.023031](https://doi.org/10.1117/1.JEI.22.2.023031)]*

1 Introduction

Motion estimation (ME) is used in many video coding standards, such as MPEG-1, MPEG-4, and H.264/AVC, to remove interframe redundancies. There are two main factors that determine the performance of ME. One is the size of the search range (SR), and the other is the number of reference frames (RFs). The values of these two factors are set manually in the configuration file of the encoder and are used as fixed parameters in the whole ME process. Although larger SR and RF values enable better coding performance, they also impose a heavy encoding complexity. In the overall coding system, about 50% of the computation time and 90% of the memory access are dedicated to ME.^{1,2} Indeed, the memory access bandwidth required to read reference pixels from external memory has gradually become the bottleneck of real-time applications.

Because the coding video differs considerably, fixed SR and RF settings are not suitable for all situations. For example, if the video sequence is motionless, a small SR and single RF are sufficient to determine the best motion vector (MV). On the other hand, if the sequence contains a high degree of movement, we can adaptively adjust the above

settings to achieve better video quality. In such cases, dynamic SR adjustment and multiframe selection algorithms have become important in improving coding efficiency and satisfying the needs of real-time applications.

Many algorithms that adaptively adjust the SR size and select the RFs have been proposed. For instance, the influence of the SR on the compression rate and coding quality has been examined in an extensive series of experiments.³ The maximum-likelihood estimator has been used to model the probability density function of the motion vector difference (MVD), with the optimal SR then calculated for a given hitting probability.^{4,5} However, the integral operation and likelihood estimator make the algorithm somewhat complicated. In Ref. 6, images are divided into motionless and moving regions, and different SR selection methods are given for each. Using the spatial correlation, the SR can be set as a function of the maximum of the neighboring matching errors.⁷ Similarly, the block can be classified as motionless, slow, or violent based on the magnitude of the predicted motion vector (PMV), allowing a 3×3 , 7×7 , or 9×9 search area to be selected accordingly.^{8,9} In Ref. 10, three different SRs are selected based on certain thresholds. However, with only a few fixed SRs and thresholds, the adaptability of these algorithms^{8–10} is insufficient to cope with the diversity of video sequences. In Ref. 11, the correlation between different block partitions is used to predict the SR size. The SR of the current block has been calculated based on the relationship between the probability density function of MVD and the MV distribution of the neighboring blocks.¹² An analysis of the relationship between the compression ratio and the size of the SR has been shown to allow a smaller SR for macroblocks (MBs) with smaller compression ratios.¹³ In Ref. 14, a linear SR model is defined based on the 1-bit image plane.

Existing multiframe selection algorithms are generally classified into three categories. In the first, the MV composition technique is used to reduce the calculation load of multiframe ME, which uses the MVs from the previous RFs to generate those for the current RF. The activity dominant vector selection criterion can be used to obtain the composited MV.¹⁵ The MVs of two successive frames can synthesize the MV of the current RF,¹⁶ with only a small range around the composited MV needed for ME. In Ref. 17, the MVs with the same characteristics are first merged and

Paper 13062 received Feb. 6, 2013; revised manuscript received May 4, 2013; accepted for publication May 23, 2013; published online Jun. 21, 2013.

then the two most relevant MVs are selected to generate the MV of the current block. To apply the MV composition algorithm, the MVs of different block types must be stored for further calculation. Therefore, this approach inevitably involves a large additional memory space.

In the second category, the candidate reference frames (CRFs) are predicted from the temporal and spatial neighbors. The priorities of all RFs can be computed from the spatial and temporal correlation of the RF index and MVs,¹⁸ or the coding modes and RFs from some reference region may be used to determine the CRFs for the current block.¹⁹ In Ref. 20, the initial RF is dynamically selected using the mean of the RF indexes of spatially neighboring MBs.

The third category speeds up the multiframe selection procedure by employing an early termination strategy. For instance, based on the spatial and temporal consistencies of the motion field, multiframe ME may only be selected for a small region of the MBs,²¹ or the experimentally obtained sum of absolute difference (SAD) curve could be used to skip the third and fourth RFs.²² In Ref. 23, the relationship between the SAD and the all-zero quantized transform coefficients was deduced. If the SAD is less than some threshold, the multiframe ME procedure should be terminated immediately. Alternatively, the MVs and SADs from previous RFs can be used to automatically confirm whether to search the last two RFs.²⁴

Different from the above-mentioned methods, we propose a different dynamic SR and multiframe selection algorithm for the H.264/AVC ME process. For the SR adjustment scheme, we first analyze the relationship between the PMV and the SR and then use this relationship to adjust the SR size. For the multiframe selection scheme, we focus on the correlation between different block sizes of the MB to adaptively select RFs. Simulation results show that the proposed algorithm achieves a higher speedup while maintaining almost the same video quality and total bit rate.

The rest of this paper is organized as follows. Sections 2 and 3 describe the algorithms for dynamic SR adjustment and multiframe selection, respectively. Our simulation results are presented in Sec. 4, and we give our conclusions in Sec. 5.

2 Proposed SR Adjustment Algorithm

2.1 Motivation and Analysis

The SR size can be successfully reduced using the algorithms mentioned in Sec. 1, but the efficiency can be further improved. In the H.264/AVC coding standard, the PMV decides the center of a search area. Therefore, the SR size depends on the location of the PMV more than any other factors, such as the maximum matching error⁷ or the motion activity.⁸ The relationship between the SR and the PMV is illustrated in Fig. 1. The vector mv_0 denotes the search center and is calculated as the median operator over the neighboring MVs, and mv_n is the final MV of block N obtained by ME. The vector mvd_n is the difference between mv_0 and mv_n . Intuitively, when PMV (mv_0) is accurate (i.e., when mv_0 is close to mv_n), fewer regions need to be searched before the final motion vector (mv_n) is obtained. Hence, we can consider an adaptive SR adjustment algorithm which depends on the PMV.

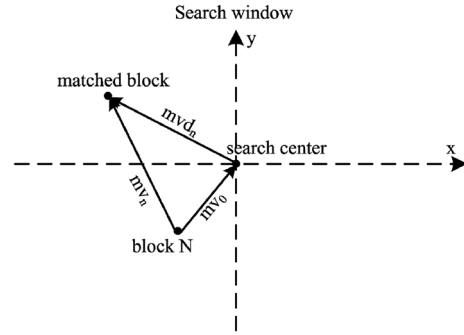


Fig. 1 Relationship between the search range (SR) and PMV.

To determine an MV within the search window, SAD has been widely used to measure the prediction distortion between the current and the candidate blocks. For a block of size $W \times H$ SAD is defined as

$$SAD[c, r(mv)] = \sum_{i=m, j=n}^{m+W, n+H} |curr(i, j) - ref(i + mv_x, j + mv_y)| \quad (1)$$

where $curr(i, j)$ denotes the pixel value of the current block and $ref(i + mv_x, j + mv_y)$ is the pixel value of the candidate block for a given motion vector $mv(mv_x, mv_y)$. In our scheme, we define the Dif metric to evaluate the accuracy of the PMV. In other words, we use Dif to measure the distance between the MV and its predictor. It is defined as follows:

$$Dif = |SAD_I[c, r(mv_0)] - SAD_I[c, r(mv_i)]|, \quad (2)$$

where $SAD_I[c, r(mv_0)]$ and $SAD_I[c, r(mv_i)]$ are the SADs computed at the PMV (mv_0) and the MV (mv_i) of block I , respectively. A small Dif value implies an accurate predictor. Thus, we may expect a high probability of a small SR. To reveal the relationship between Dif and the SR size, the experimental results from a variety of video sequences are shown in Fig. 2. In this figure, $|mvd_n|$ represents the magnitude of the difference between MV and PMV, and

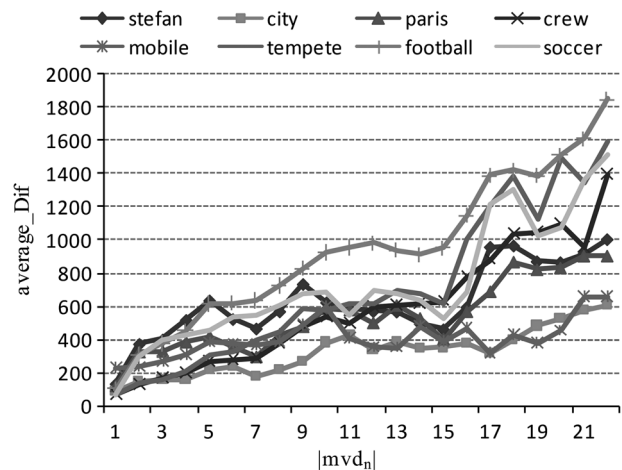


Fig. 2 Relationship between Dif and $|mvd_n|$. (Test conditions: encoded frame = 100; number of reference frames(RFs) = 1; quantization parameter (QP) = 28; SR = 16; group structure is IPP.)

average_Dif is the average Dif value of blocks with the same $|mvd_n|$. A clear positive correlation can be seen, especially for $|mvd_n|$ values up to 11. Although average_Dif does not increase monotonously over the whole $|mvd_n|$ range, there is a high probability that average_Dif will become larger as $|mvd_n|$ increases. Hence, we can develop an adaptive SR selection algorithm, which uses Dif as the criterion for determining the SR. That is, the small Dif means the accurate PMV and the small SR value, whereas the large Dif means the large SR value.

For the current block N in Fig. 1, mv_n is not available until the ME process has been performed across the whole search area, so we must set a predictor for $SAD_N[c, r(mv_n)]$ before it can be used. The influence of this predictor on coding quality and efficiency is briefly analyzed as follows. In Eq. (2), if a larger predictor is selected for $SAD_N[c, r(mv_a)]$, a smaller Dif value will be obtained. According to Fig. 2, this should result in a smaller SR size. Although a higher coding speed can be achieved with a smaller SR, the coding quality may be lost if the true MV falls outside this smaller SR. In contrast, a smaller predictor gives a larger SR and a higher coding quality. Hence, our purpose is not to get an accurate estimate for $SAD_N[c, r(mv_n)]$ but to achieve a tradeoff between the coding quality and the coding efficiency. Therefore, we replace $SAD_N[c, r(mv_n)]$ with a smaller predictor:

$$SAD_N[c, r(mv_n)] = \min\{[SAD_A[c, r(mv_a)], SAD_B[c, r(mv_b)], SAD_C[c, r(mv_c)], SAD_E[c, r(mv_e)]]\} \quad (3)$$

where $SAD_A[c, r(mv_a)]$, $SAD_B[c, r(mv_b)]$, $SAD_C[c, r(mv_c)]$, and $SAD_E[c, r(mv_e)]$ are the SADs computed from the MVs of the spatially neighboring blocks (A is the left block, B is the upper block, C is the upper-right block) and the collocated block (E).

To estimate the SR size of block N , let Dif_N , Dif_A , Dif_B , Dif_C , and Dif_E denote the Dif values of the current block N and its four neighbors (left, upper, upper-right, and collocated block), respectively. Dif_A , Dif_B , Dif_C , and Dif_E are subtracted from Dif_N , and the four differences are collected in the set ΔDif :

$$\Delta Dif = \{Dif_N - Dif_i, i = A, B, C, \text{ and } E\} \quad (4)$$

The SR is then adjusted according to the following conditions.

Condition 1: If any element in ΔDif is smaller than 0, Dif_N is smaller than the Dif value of (at least) one of its four neighbors. Thus, it can be determined that the PMV of block N is closer to its MV than its neighbor, and therefore the following smaller SR should be set for block N :

$$SR = \max[1, \min(SR_A, SR_B, SR_C, SR_E)]. \quad (5)$$

In Eq. (5), SR_A , SR_B , SR_C , and SR_E are the SRs of blocks A, B, C, and E, respectively.

Condition 2: If all elements in ΔDif are greater than 0 and $Dif_N > \alpha \times \max(Dif_A, Dif_B, Dif_C, Dif_E)$, where α is an adjustment parameter, the PMV of the current block N is far from its MV, which is likely to be due to complex motion. In

such cases, in order to guarantee the coding quality, a larger SR should be set as follows:

$$SR = \min\left[R, \max(SR_A, SR_B, SR_C, SR_E) + \frac{R}{8}\right], \quad (6)$$

where R is a fixed SR given in the configuration file of the encoder. Based on extensive experiments, we set the value of α to 2.

Condition 3: If conditions 1 and 2 are not satisfied, the two minimum elements in ΔDif (corresponding to the blocks that have the closest Dif values to the current block N) are selected. These blocks are marked b_1 and b_2 , and the SR of block N is set according to

$$SR = \min[R, \max(SR_{b_1}, SR_{b_2}) + 1], b_1, b_2 \in \{A, B, C, E\} \quad (7)$$

After determining the new SR according to Eqs. (5)–(7), the ME is performed within this new SR.

2.2 Proposed Algorithm

For clarity and completeness, we summarize the detailed description of the proposed algorithm as follows:

Dif_N is calculated for current block N according to Eqs. (2) and (3), and ΔDif is obtained according to Eq. (4). The following conditions are then used to adjust the size of SR.

- If condition 1 [$\min(\Delta Dif) \leq 0$] is satisfied, SR is set according to Eq. (5).
- If condition 2 [$\min(\Delta Dif) \leq 0$ and $Dif_N > \alpha \times \max(Dif_A, Dif_B, Dif_C, Dif_E)$] is satisfied, SR is set according to Eq. (6).
- If neither condition 1 nor condition 2 is satisfied, SR is set according to Eq. (7).

After finding the MV within the new SR defined by our algorithm, the Dif and SR values of the block N are saved for the next coding block.

In order to verify the correctness of the proposed algorithm, we analyzed the hit ratio to see whether our new SR included the result of ME. The hit ratios for different quantization parameters (QPs) are shown in Table 1. It can be seen that our algorithm gives a high accuracy of 93.28% to 99.61%. The Bus, Mobile, and Stefan sequences contain a high level of motion and thus obtain relatively low hit ratios. For the Akiyo, Paris, and Flower sequences, which contain slower motion, the proposed algorithm yields a hit ratio of around 99%.

3 Proposed Multiframe Selection Algorithm

3.1 Motivation and Analysis

Figure 3 describes the relationship between multiframe and variable block size ME. As shown in Fig. 3, the multiframe ME process is performed on each block size of the MB. These blocks are different partitions of the same MB, so their best RFs are highly correlated. According to this feature, the best RFs of the upper divisions can be used as CRFs for the lower divisions. However, using this up-to-down method to predict the CRFs of the lower divisions has

Table 1 Hit ratio of the motion vector (MV) within the new search range (SR) (test conditions: encoded frame = 100; number of reference frames (RFs) = 1; QP = 28, 32, 36, 40; SR = 16; group structure is IPP).

| Sequence | Hit rate (%) | | | | Average |
|-------------|--------------|---------|---------|---------|---------|
| | QP = 28 | QP = 32 | QP = 36 | QP = 40 | |
| Stefan_Cif | 93.60 | 94.34 | 95.16 | 95.90 | 94.75 |
| ForemanCif | 94.62 | 96.42 | 96.67 | 96.87 | 96.14 |
| ParisCif | 97.74 | 98.79 | 98.92 | 99.00 | 98.61 |
| Mobile_Cif | 94.91 | 95.62 | 96.21 | 96.73 | 95.86 |
| Akiyo_Cif | 99.16 | 99.56 | 99.87 | 99.88 | 99.61 |
| Flower_Cif | 98.18 | 98.98 | 99.24 | 99.99 | 99.09 |
| Soccer_4Cif | 93.34 | 94.19 | 94.83 | 95.13 | 94.37 |
| City_4Cif | 94.46 | 95.37 | 98.01 | 95.20 | 95.76 |
| Tempete_Cif | 94.34 | 95.73 | 96.33 | 96.91 | 95.82 |
| Bus_Cif | 92.85 | 92.19 | 93.51 | 94.59 | 93.28 |

two shortcomings. First, if the MB contains the object boundary or the complex content, then the probability that the lower and upper divisions will share the same best RF will be small. Second, the lower partitions have more sub-blocks than the upper partitions. Therefore, this up-to-down prediction method is not very efficient. For example, the $P8 \times 8$ division contains 36 sub-blocks in total. If the number of CRFs is 2, then the 36 sub-blocks will require 72 ME procedures, whereas for the 16×16 division, only two ME procedures are required. Therefore, reducing the number of CRFs in the lower divisions is critical for improving the efficiency of ME.

In summary, the 8×8 block is used as the basic unit in this study to determine the best RFs for the other coding modes. We use the 8×8 block as it better reflects the motion and texture characteristics of the MB and improves the efficiency as much as possible. For convenience, the RFs are hereafter marked as $ref_0, ref_1, ref_2, ref_3, \dots$ in accordance with their distance from the current frame.

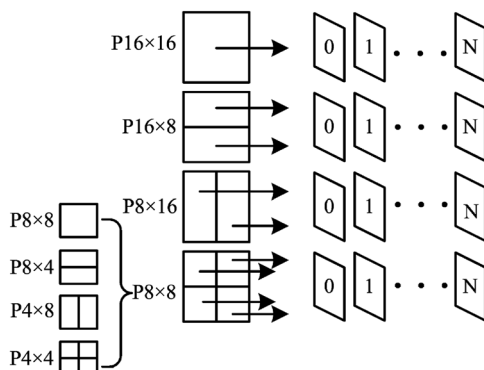


Fig. 3 Multiple RFs motion estimation (ME) with various block size.

Due to the temporal consistency of a video sequence, it is highly probable that blocks in static or slow-moving regions will select ref_0 as the optimal RF. In this work, the movement complexity (MC) of the MB is defined as

$$MC = \max(|MV_0|, |MV_1|, |MV_2|, |MV_3|), \quad (8)$$

where $MV_0, MV_1, MV_2,$ and MV_3 denote the MVs of four 8×8 blocks at ref_0 .

To inspect the relationship between movement complexity and ref_0 , various video sequences with different motion activities are used as the test material. Table 2 documents the average experimental results for different QPs. T_B is the number of blocks with the same movement complexity, and RF_ratio represents the percentage of blocks whose ultimate RF is ref_0 . For the static blocks (which MC is equal to 0), an average of 97.87% of them select ref_0 as the best RF. For the News, Container, and Akiyo sequences, the percentage of blocks choosing ref_0 among T_B is greater than 99%. Table 2 also shows that the RF_ratio decreases as the value of MC increases. For the Stefan sequence, in particular, when the MC is 3, only 68.32% of T_B select ref_0 as their ultimate RF. Therefore, we can conclude that when the value of MC is less than some threshold Th (set to 2 in this paper), there is a very high probability that only ref_0 will be used as the best RF. From this result, we conclude that the search across other RFs can be skipped.

If MC is larger than Th, it indicates that the motion is complex. Thus, we perform ME on the remaining RFs for each 8×8 block. The reference maps are then generated according to the correlation between the 8×8 block and the other blocks. Figure 4 shows the reference maps for other blocks, where $ref_a, ref_b, ref_c,$ and ref_d denote the optimal RFs of the first, second, third, and fourth 8×8 block, respectively. The CRFs for each partition are described as follows:

- The $P16 \times 16$ mode is used to encode regions containing slow motion. Thus, if more than two RFs are contained in Fig. 4(a), the motion of the current block is complex and the $P16 \times 16$ mode cannot be selected. In this case, only ref_0 is selected as the best RF; otherwise all RFs contained in Fig. 4(a) are selected as CRFs for the 16×16 block.
- For the 16×8 block, ref_a and ref_b are regarded as CRFs for the upper 16×8 division, whereas ref_c and ref_d are CRFs for the lower 16×8 division in Fig. 4(b).
- For the 8×16 block, ref_a and ref_c are regarded as CRFs for the left 16×8 division, whereas ref_b and ref_d are CRFs for the right 16×8 division in Fig. 4(c).
- For the three 8×8 subdivisions, the best RF for the 8×8 block is directly used as the best RF for its subdivisions in Fig. 4(d).

3.2 Proposed Algorithm

Based on the above description, the pseudocode for our multiframe selection algorithm can be written as shown in Fig. 5. The initial number of CRFs is usually set to 5.

Table 2 Relationship between the movement complexity and ref_0 (test conditions: encoded frame = 60; number of RFs = 5; QP = 28, 32, 36, 40; SR = 16; group structure is IPP).

| Sequence | MC = 0 | MC = 1 | MC = 2 | MC = 3 |
|---------------|-----------------|-----------------|-----------------|-----------------|
| | T.B/RFratio (%) | T.B/RFratio (%) | T.B/RFratio (%) | T.B/RFratio (%) |
| Akiyo_Cif | 264,189/99.58 | 86,128/97.30 | 29,605/91.85 | 25,107/91.52 |
| Foreman_Cif | 43,255/95.07 | 19,532/90.63 | 18,055/86.83 | 25,834/84.51 |
| News_Cif | 225,201/99.41 | 98,559/96.78 | 29,105/91.80 | 24,615/92.47 |
| Container_Cif | 235,774/99.01 | 118,000/93.43 | 45,186/84.95 | 19,903/87.60 |
| Stefan_Cif | 49,608/95.92 | 39,998/86.18 | 27,724/73.45 | 29,338/68.32 |
| Highway_Cif | 213,324/96.31 | 34,260/90.09 | 12,538/85.63 | 25,929/84.60 |
| Flower_Cif | 128,911/98.90 | 3,350/91.39 | 2,658/90.26 | 14,392/87.08 |
| Silent_Cif | 170,445/99.00 | 114,479/97.77 | 34,214/93.76 | 26,529/90.71 |
| Hall_Cif | 242,229/99.61 | 112,543/92.41 | 20,033/89.84 | 12,179/86.87 |
| Average | /97.87 | /92.87 | /87.60 | /86.00 |

As shown in Fig. 5, ME is first performed on ref_0 for the four 8×8 blocks, and the MVs of the four blocks are used to calculate the motion complexity according to Eq. (8). If MC is less than the threshold Th (set as 2), then ref_0 is set as the CRF. Otherwise, ME is conducted on from ref_1 to ref_4 for the four 8×8 blocks, respectively. The next step is to determine a candidate reference frames set (CRFS) for other partitions based on the reference maps in Fig. 4. ME is then performed on the RFs in the CRFS for the 8×8 sub-block, 16×16 block, 16×8 block, and the 8×16 block. In the ME process for H.264/AVC, the coding order is $P16 \times 16$, $P16 \times 8$, $P8 \times 16$, and then $P8 \times 8$. As the proposed algorithm changes the coding order, i.e., the 8×8 block is processed first, the MVs of the second, third, and fourth 8×8 blocks must be repredicted after finishing the whole ME process, otherwise there will be some inconsistency between the decoder and encoder.

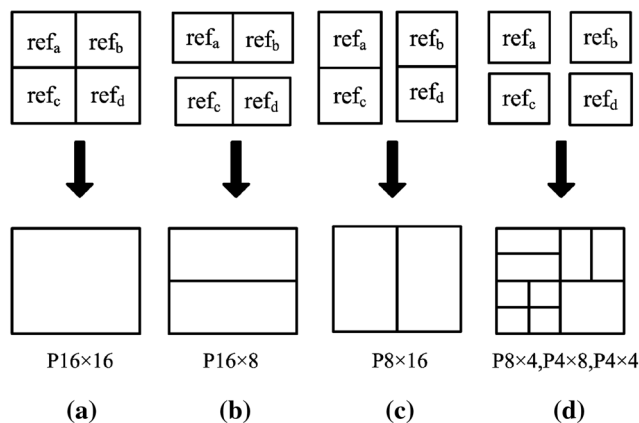


Fig. 4 Reference maps of different partitions.

Using the pseudocode above, we investigate the number of RFs under the best- and worst-case scenarios. In the best case, only one RF (ref_0) is required for all block divisions. In the worst case, all five RFs are required for the 8×8 blocks; each of the 16×16 block, 16×8 block, and 8×16 block have two CRFs. However, subpartitions such as 8×4 , 4×8 , and 4×4 will always have one RF. Therefore, the average number of RFs required for the entire coding of the MB is $[(4 \times 5) + (1 \times 2) + (2 \times 2) + (2 \times 2) + (16 \times 1) + (8 \times 1) + (8 \times 1)]/41 \approx 1.51$.

To verify the effectiveness of the above method, the average hit ratio for different QPs at each block size was

```

for (j=0; j<4; j++) {
    Perform the motion estimation for the jth 8x8 block on ref_0. }
if (MC < Th)
    CRFS = {ref_0};
else{
    for (j=0; j<4; j++)
        for (i=1; i<5; i++)
            Perform the motion estimation for the jth 8x8 block on ref_i.}
    Set CRFS for each partition according to the Figure 4.

for (i=0; i<5; i++) {
    If ref_i is in the CRFS :
        Perform the motion estimation for the sub-partition of 8x8
        block size on ref_i. }
    :
    :
    :
for (i=0; i<5; i++){
    If ref_i is in the CRFS :
        Perform the motion estimation for the 8x16 block size on ref_i.}

Perform motion vector prediction for 8x8 block size .
    
```

Fig. 5 Pseudocode for our algorithm.

Table 3 Average hit ratios of the proposed algorithm for each block size (test conditions are the same as Table 2).

| Sequence | P16 × 16 | P16 × 8 | P8 × 16 | P8 × 8 | Sub-P8 × 8 |
|----------------|----------|---------|---------|--------|------------|
| Akiyo_Cif | 98.31 | 97.91 | 97.82 | 99.58 | 96.60 |
| News_Cif | 98.23 | 97.43 | 97.22 | 99.12 | 95.76 |
| Stefan_Cif | 92.17 | 91.21 | 90.76 | 97.68 | 88.06 |
| Forman_Cif | 92.17 | 90.00 | 89.37 | 97.94 | 87.32 |
| Hall_Cif | 97.35 | 97.20 | 96.96 | 98.63 | 96.34 |
| Mobile_Cif | 95.52 | 90.65 | 89.48 | 98.07 | 84.75 |
| Container_Cif | 97.21 | 96.67 | 96.79 | 99.09 | 95.35 |
| Paris_Cif | 97.24 | 95.76 | 95.87 | 98.92 | 93.16 |
| Coastguard_Cif | 96.18 | 92.37 | 93.14 | 99.08 | 90.13 |
| Average | 95.76 | 93.91 | 93.70 | 98.57 | 91.36 |

compared to the full RF search method. The results are shown in Table 3. As can be seen, our algorithm achieves a high average accuracy of 91% to 98% for all block partitions. For the 16 × 16 block size, the hit ratio averages above 95%. Even the 8 × 8 sub-block averages over 91%, with a worst-case performance of 84.75% with the Mobile sequence.

4 Simulation Results and Analysis

To evaluate the performance of the proposed approach, it was implemented on the H.264/AVC reference software JM15.1. Various standard sequences were used, including CIF, 4CIF, and HD sizes, at 30 fps for 100 frames. The simulations were performed with rate distortion (RD) optimization and Hadamard transform enabled, QP = 28, 32, 36, and 40, IPPP sequence types, CAVLC entropy coding, and one or five RFs. The default SR was set to 16 for CIF, 32 for 4CIF, and 64 for HD sequences.

For a numerical comparison, the Bjontegaard delta peak signal-to-noise ratio (BDPSNR) and Bjontegaard delta bit rate (BDBR) were used to evaluate the coding quality. These quantities are recommended for comparing the difference between two RD curves.²⁵ A negative BDPSNR (dB) or positive BDBR (%) indicates a coding loss over the full-search algorithm in JM15.1. For an objective comparison, the average used reference frame (URF) and average computational complexity (CPX) are used to measure the complexity reduction. CPX (%)⁴ is the speed-up ratio of the SR adjustment algorithm, which is defined as the ratio of the number of search points in the proposed method to the number of search points in the full-search motion estimation (FME) algorithm. For example, a CPX value of 21% implies that the number of search points in the proposed method is 21% of the number in the FME algorithm. The average URF represents the average number of RFs used by each MB, with a smaller URF signifying fewer RFs.

For an objective comparison of coding performance, the proposed approach was compared with corresponding methods. Our SR adjustment method was compared to two recent schemes.^{4,11} It can be seen from Table 4 that the proposed

Table 4 Performance comparison of the proposed algorithm on JM 15.1 (with one RF).

| Sequence | CPX | | | BDBR (%) | | | BDPSNR (dB) | | |
|----------------|--------------------|------|-------|--------------------|-------|-------|--------------------|-------|-------|
| | Ours (Refs. 4, 11) | | | Ours (Refs. 4, 11) | | | Ours (Refs. 4, 11) | | |
| Akiyo_Cif | 2.85 | 4.13 | 12.60 | +0.01 | +0.18 | +0.01 | -0.00 | -0.02 | -0.00 |
| Mobile_Cif | 3.51 | 7.19 | 14.68 | +0.01 | +0.77 | -0.01 | -0.01 | -0.04 | +0.01 |
| Hall_Cif | 2.95 | 5.81 | 13.08 | +0.17 | +0.91 | +0.14 | -0.06 | -0.04 | -0.05 |
| Coastguard_Cif | 3.35 | 6.94 | 13.86 | +0.11 | +0.32 | +0.08 | -0.05 | -0.01 | -0.05 |
| Forman_Cif | 3.31 | 7.02 | 14.14 | +0.35 | +0.95 | +0.28 | -0.10 | -0.07 | -0.04 |
| Stefan_Cif | 3.39 | 7.14 | 15.71 | +0.15 | +1.57 | +0.05 | -0.06 | -0.09 | -0.03 |
| Crew_4Cif | 2.49 | 4.09 | 16.70 | +0.11 | +0.89 | +0.03 | -0.01 | -0.05 | -0.02 |
| City_4Cif | 1.67 | 2.89 | 13.62 | +0.17 | +0.65 | +0.01 | -0.08 | -0.04 | -0.01 |
| Soccer_4Cif | 1.29 | 3.11 | 13.15 | +0.31 | +1.73 | +0.04 | -0.13 | -0.12 | -0.07 |
| Rush_hour_Hd | 1.70 | 3.43 | 10.36 | +0.20 | +0.56 | +0.05 | -0.10 | -0.14 | -0.05 |
| Station2_Hd | 1.69 | 2.55 | 9.67 | +0.04 | +1.45 | +0.03 | -0.08 | -0.06 | -0.04 |
| Average | 2.56 | 4.94 | 13.42 | +0.14 | +0.91 | +0.06 | -0.06 | -0.06 | -0.03 |

Table 5 Performance comparison with UMHexagonS algorithm (test conditions are the same as Table 4).

| Sequence | BDBR | BDPSNR | Δ MET |
|----------------|-------|--------|--------------|
| | (%) | (dB) | (%) |
| Akiyo_Cif | -0.08 | +0.11 | +14.12 |
| Mobile_Cif | -0.06 | +0.03 | -16.61 |
| Hall_Cif | +0.07 | -0.01 | +12.38 |
| Coastguard_Cif | +0.08 | -0.04 | -11.11 |
| Forman_Cif | +0.21 | +0.05 | +3.77 |
| Stefan_Cif | +0.17 | +0.02 | -9.96 |
| Crew_4Cif | +0.02 | -0.01 | -22.87 |
| City_4Cif | -0.22 | +0.11 | -21.35 |
| Soccer_4Cif | +0.25 | -0.13 | -19.72 |
| Rush_hour_Hd | +0.11 | -0.09 | -14.94 |
| Station2_Hd | +0.19 | -0.04 | -14.32 |
| Average | +0.07 | -0.01 | -9.15 |

algorithm reduced the number of search points by an average of 2.56%, albeit at the cost of a 0.14% increase in bit rate and a 0.06 dB reduction in peak signal-to-noise ratio (PSNR). When the hitting probability of MVDs was set to 0.7, the method of Ko et al.⁴ exhibited a good performance, with an average CPX of 4.94%. However, the implementation cost of this method is much higher than that of the proposed scheme due to the overheads inherent in the integral operation and likelihood estimation. For several sequences, such as Stefan, Station2_Hd, and Soccer_4Cif, the method of Ref. 11 exhibited better coding quality than our method in terms of BDPSNR and BDBR, but our approach can further deduct an average of 11% search points with only a slight bit rate increase and small PSNR decrease.

For further comparison, Table 5 depicts the ME time of the proposed method compared to the UMHexagonS algorithm in JM15.1. This shows that the proposed method reduces the ME time by 9.15% on average, with a marginal increase in BDBR (+0.07%) and a small loss of BDPSNR (-0.01 dB). For sequences with static and slow motion activity, such as Akiyo_Cif and Hall_Cif, UMHexagonS achieves a high speed-up factor, because the ME process is terminated after finishing the small and middle diamond pattern search. In contrast, our algorithm works well for CIF sequences with high motion levels and 4CIF and HD sequences with larger SR values.

We compared our multiframe selection algorithm with two well-known efficient algorithms.^{19,23} The comparison results are listed in Table 6. For most sequences, the method of Lee et al.¹⁹ yielded a high bit rate increment and PSNR loss (average of 0.09 dB BDPSNR decrease and 0.61%

Table 6 Performance comparison of the proposed algorithm on JM 15.1 (with five RFs).

| Sequence | URF | | | BDBR (%) | | | BDPSNR (dB) | | |
|---------------|---------------------|------|------|---------------------|-------|-------|---------------------|-------|-------|
| | Ours (Refs. 19, 23) | | | Ours (Refs. 19, 23) | | | Ours (Refs. 19, 23) | | |
| | | | | | | | | | |
| Akiyo_Cif | 1.09 | 1.44 | 1.19 | +0.03 | +0.89 | +0.01 | -0.01 | -0.09 | -0.01 |
| Mobile_Cif | 1.43 | 2.39 | 2.03 | +0.26 | +0.73 | +0.55 | -0.08 | -0.11 | -0.11 |
| Hall_Cif | 1.11 | 1.73 | 1.56 | +0.47 | +0.42 | +0.25 | -0.13 | -0.07 | -0.06 |
| Bridge_Cif | 1.09 | 1.76 | 1.38 | +0.07 | +0.27 | +0.05 | -0.02 | -0.06 | -0.01 |
| Waterfall_Cif | 1.32 | 1.87 | 1.54 | +0.01 | +0.56 | +0.49 | -0.04 | -0.08 | -0.04 |
| Paris_Cif | 1.19 | 1.72 | 1.43 | +0.11 | +0.30 | +0.48 | -0.06 | -0.10 | -0.05 |
| Silent_Cif | 1.16 | 1.69 | 1.18 | +0.05 | +0.78 | +0.73 | -0.02 | -0.06 | -0.12 |
| Flower_Cif | 1.32 | 1.87 | 1.52 | +0.15 | +0.54 | +0.51 | -0.08 | -0.08 | -0.04 |
| Foreman_Cif | 1.39 | 2.01 | 1.64 | +0.05 | +0.77 | +0.87 | -0.02 | -0.17 | -0.05 |
| Stefan_Cif | 1.40 | 2.18 | 1.93 | +0.34 | +0.90 | +0.49 | -0.11 | -0.10 | -0.07 |
| Tempete_Cif | 1.29 | 2.02 | 1.81 | +0.23 | +0.46 | +0.55 | -0.09 | -0.11 | -0.07 |
| Average | 1.25 | 1.88 | 1.56 | +0.16 | +0.61 | +0.45 | -0.06 | -0.09 | -0.06 |

BDBR increase), with 1.88 frames searched compared to the JM15.1 reference encoding. The method of Ref. 23 achieved good PSNR and bit rate performance (average 0.06 dB BDPSNR decrease and 0.45% BDBR increase) with a higher speed-up factor (1.52 RFs used). The proposed algorithm gave superior results to both of these techniques in terms of coding complexity reduction. It outperformed the method of Lee et al.¹⁹ in all aspects, with a 0.63-frame reduction in the number of searched RFs, which implies a reduction of 12% in the number of ME operations. At the same time, the proposed method improved the URF value by 0.31 frames compared to the method of Huang et al.,²³ with a better bit rate and the same PSNR.

It should be mentioned that the URF and CPX parameters are virtually independent of the programming quality or hardware platform. Therefore, it is clear that the performance of the proposed methods is superior to the corresponding methods evaluated above. It must also be noted that the proposed methods can be combined with any fast ME method, such as Fast Full search, EPZS, and UMHExagonS, in the JM reference encoder.

5 Conclusions

In this paper, we presented an efficient adaptive SR adjustment and multiframe selection algorithm for ME. The proposed SR adjustment algorithm adjusts the SR for ME based on the accuracy of PMV. We also designed an RF selection scheme with the aim of reducing the complexity of multiframe ME based on the relationship between the different block sizes of the MB. The described technique significantly reduces the CPX of ME in H.264/AVC. Through a comparative analysis, it was found that the proposed approach gave an average RF saving of 3.75 frames (1.25 URFs) with a setting of five RFs and a reduction of 97% in the number of search points with a setting of one RF, with a negligible bit rate increment and a minimal loss of image quality. These features will be important in the development of real-time coding applications for video conferencing and telephony. In future work, we will combine our two algorithms to further speed up the ME process.

References

- Z. He et al., "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Trans. Circuits Syst. Video Technol.* **15**(5), 645–658 (2005).
- K. Denolf et al., "Initial memory complexity analysis of the AVC codec," in *IEEE Workshop on Signal Processing Systems*, pp. 222–227, IEEE Signal Processing Society, San Diego (2002).
- C. A. Gonzales, H. Yeo, and C. J. Kuo, "Requirements for motion-estimation search range in MPEG-2 coded video," *IBM J. Res. Dev.* **43**(4), 453–470 (1999).
- Y.-H. Ko, H.-S. Kang, and S.-W. Lee, "Adaptive search range motion estimation using neighboring motion vector differences," *IEEE Trans. Consum. Electron.*, **57**(2), 726–730 (2011).
- H.-S. Kang, S.-W. Lee, and H. G. Hosseini, "Probability constrained search range determination for fast motion estimation," *J. Electron. Telecommun. Res. Inst.* **34**(3), 369–378 (2012).
- I. Kim and J. Kim, "Low-complexity block-based motion estimation algorithm using adaptive search range adjustment," *Opt. Eng.* **51**(6), 067010 (2012).
- S. Lee, "Fast motion estimation based on adaptive search range adjustment and matching error prediction," *IET Image Process* **4**(1), 1–10 (2010).
- Y. Ismail et al., "Fast variable padding motion estimation using smart zero motion prejudgment technique for pixel and frequency domains," *IEEE Trans. Circuits Syst. Video Technol.* **19**(5), 609–626 (2009).
- Y. Ismail et al., "Fast motion estimation system using dynamic models for H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.* **22**(1), 28–42 (2012).
- S.-W. Lee, S.-M. Park, and H.-S. Kang, "Fast motion estimation with adaptive search range adjustment," *Opt. Eng.* **46**(4), 040504 (2007).
- S. Lee, K. Choi, and E. S. Jang, "Adaptive search range adjustment scheme for fast motion estimation in AVC/H.264," *Opt. Eng.* **50**(6), 067402 (2011).
- C.-C. Lou, S.-W. Lee, and C.-C. J. Kuo, "Adaptive motion search range prediction for video encoding," *IEEE Trans. Circuits Syst. Video Technol.* **20**(12), 1903–1908 (2010).
- J. Jung, J. Kim, and C.-M. Kyung, "A dynamic search range algorithm for stabilized reduction of memory traffic in video encoder," *IEEE Trans. Circuits Syst. Video Technol.* **20**(7), 1041–1046 (2010).
- O. Urhan, "Constrained one-bit transform based fast block motion estimation using adaptive search range," *IEEE Trans. Consum. Electron.* **56**(3), 1868–1871 (2010).
- M.-J. Chen et al., "Fast multiframe motion estimation algorithms by motion vector composition for the MPEG-4/AVC/H.264 standard," *IEEE Trans. Multimedia* **8**(3), 478–487 (2006).
- S.-E. Kim, J.-K. Han, and J.-G. Kim, "An efficient scheme for motion estimation using multireference frames in H.264/AVC," *IEEE Trans. Multimedia* **8**(3), 457–466 (2006).
- T.-K. Lee, Y.-L. Chan, and C.-H. Fu, "Reliable tracking algorithm for multiple reference frame motion estimation," *J. Electron. Imaging* **20**(3), 033003 (2011).
- D. S. Jun and H. W. Park, "An efficient priority-based reference frame selection method for fast motion estimation in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.* **20**(8), 1156–1161 (2010).
- K. Lee, G. Jeon, and J. Jeong, "Fast reference frame selection algorithm for H.264/AVC," *IEEE Trans. Consum. Electron.* **55**(2), 773–779 (2009).
- I. Park and D. W. Capson, "Improved motion estimation time using a combination of dynamic reference frame selection and residue-based mode decision," *Signal Image Video Process.* **6**(1), 25–39 (2012).
- L. Shen, Z. Liu, and Z. Zhang, "Fast multiframe selection algorithm based on spatial-temporal characteristics of motion field," *J. Electron. Imaging* **17**(4), 043004 (2008).
- L. Shen et al., "An adaptive and fast multiframe selection algorithm for H.264 video coding," *IEEE Signal Process. Lett.* **14**(11), 836–839 (2007).
- Y.-W. Huang et al., "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.* **16**(4), 507–522 (2006).
- X. Li, E. Q. Li, and Y.-K. Chen, "Fast multi-frame motion estimation algorithm with adaptive search strategies in H.264," in *Proc. Int. Conf. Acoustics Speech Signal Process.*, pp. 369–372, IEEE Signal Processing Society, Montreal, Canada (2004).
- K. Andersson, R. Sjöberg, and A. Norkin, "Reliability Measure for BD Measurements," ITU-T Q.6/SG16 VCEG 33th Meeting, Document VCEG-AL22, London, UK (July 2009).



Yingzhe Liu received BS and MS degrees from YanShan University of China in 2002 and 2004, respectively. He is currently working toward a PhD degree at the Microelectronics Center of Harbin Institute of Technology, China. His current research interests include audio and video coding and digital hardware design for video compression.

Jinxiang Wang received his MS and PhD degrees in semiconductor physics and communication and information systems from Harbin Institute of Technology, China, in 1993 and 1999, respectively. He is now a professor in the School of Astronautics, Harbin Institute of Technology. His current research interests mainly include multimedia data compression, 3G wireless communication, systems on chips (SOC), and networks on chips (NOC).

Fangfa Fu received his MS and PhD degrees in semiconductor physics and communication and information systems from Harbin Institute of Technology, China, in 2006 and 2011, respectively. He is now a lecturer in the School of Astronautics, Harbin Institute of Technology. His research interests include systems on chips (SOC) and networks on chips (NOC).