MODIFIED WINNER UPDATE WITH ADAPTIVE BLOCK PARTITION FOR FAST MOTION ESTIMATION

Shou-Der Wei, Shao-Wei Liu, Shang-Hong Lai

Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
{greco, phenixjon, lai}@cs.nthu.edu.tw

ABSTRACT

Motion estimation (ME) plays an important role in video compression. Block-based ME has been adopted in most video compression standards due to its efficiency. In this paper, we propose a novel and fast block-based ME algorithm based on applying the modified winner-update scheme in conjunction with the adaptive partition order of macroblock. The partition order is determined from the block gradient distribution. Experimental results show the proposed algorithm achieves the optimal motion estimation very efficiently.

1. INTRODUCTION

Motion estimation (ME) has been widely used in many video applications, such as video tracking, video segmentation, and video compression. Especially, block-based motion estimation is very popular for motion-compensated video compression. Block-based motion estimation is an essential part of a common video coding system, since it basically finds the temporal correlation to reduce the redundancy between frames, thus achieving high compression ratio. Many block-based ME algorithms have been proposed in the past. Most of these previous ME methods are based on using a predefined search pattern to reduce the total number of search points, such as diamond search [6], three steps search [7], new three steps search [8], and four steps search algorithm [9]. Even though these techniques can speed up the motion estimation process because they only consider a reduced number of search points in the block matching, the resulting video quality is normally further degraded.

Besides the approximate methods, another approach checked the accumulated SAD to eliminate impossible candidates as early as possible to reduce the computational cost. Li and Salari proposed the successive elimination algorithm (SEA) [1] that provides the optimal ME solution which is the same as that of full search (FS) but with less operation by using the early termination in the SAD computation. The SEA used the difference of the block sum as the elimination criterion to successively reject impossible candidates and reduce the computational cost. Lee and Chen proposed a block sum pyramid algorithm (BSPA) [10] that uses the block sum pyramid to efficiently calculate the block sums and they solved the ME problem in a coarse to fine strategy. Gao et al. extended SEA to multilevel successive elimination algorithm (MSEA) [2] to further speed up the optimal ME algorithm. Chen et al. [3] proposed a fast block matching algorithm based on the winner-update strategy, which can reduce a large portion of calculations and guarantee the search result is still globally optimal. In the winner update technique, only the current winner location with the minimal accumulated distortion is considered for updating the accumulated distortion. This updating process is repeated until the winner has gone through all the pixels in the macroblock for calculating the distortion.

In this work we divided the marcoblock by the gradient distribution for more efficient winner-update [2][4]. The region with larger gradient contains more details. Then we applied a modified winner update method in conjunction with the adaptive partition order of macroblock to achieve efficient optimal motion estimation.

The rest of this paper is organized as follow: In the next two sections, we briefly review the winner-update scheme and the MSEA. Then, we present the proposed method that performs the modified winner-update scheme with the adaptive partition order of macroblock in section 4. Some experimental results with comparisons to previous methods are given in section 5. Finally, we conclude this paper in the last section.

2. THE WINNER-UPDATE SCHEME

In this section, we use a simple poker game to describe the concept of the winner-update algorithm [3]. As illustrated in Figure 1, suppose there are four players in the poker game and each player has four cards. The player with the minimum total points of his cards is the winner of the poker game. At the first stage, every players show one of their cards, and the card's point is added into the accumulated sum. Subsequently, the current winner, i.e. the player with the minimum accumulated sum, show the next card and add

this card's point to update his accumulated sum. The above procedure is repeated iteratively until one player has showed all of his cards and his accumulated sum is the smallest among all, and then this player is the winner of the game. An example of the winner update scheme is illustrated in Figure 1. After the first stage, player 3 is the temporary winner and he shows the next card. After updating the accumulated sum of player 3, player 1 becomes the temporary winner. Following the winner update procedure described above, each player alternatively becomes the temporary winner and finally player 3 is the winner that shows all his cards with the minimal accumulated sum.

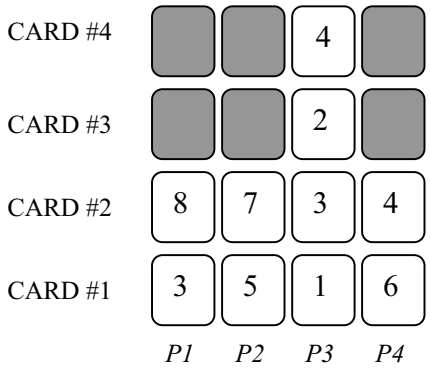

Figure 1: An illustration of winner-update strategy with a simple poker game.

3. REVIEW OF MSEA

The Successive Elimination Algorithm (SEA) [1] used an upper bound for a block sum difference as the criterion to eliminate the impossible candidate blocks to reduce the computation of motion estimation. SEA first used the predicted motion vector (PMV) as the initial motion vector and then calculated the SAD of the current macroblock and the block corresponding to the PMV as the upper bound to eliminate the impossible candidates. If the difference of block sum between current macroblock and candidate is greater than the upper bound, then this candidate block is impossible to be the block of the best match. If the difference of the block sum is less than the upper bound, then we calculate the SAD between the current block and the candidate block and then check if the SAD is less than the upper bound. When a smaller SAD is found, it is used to update the upper bound.

To obtain a tighter upper bound, the Multi-levl SEA (MSEA) was proposed to divide the block into four sub-blocks and accumulate the difference of each block sum with the candidate block as the elimination criterion. As Figure 2 shows, the macroblock size is 16x16 and there are 5 levels in MSEA. Using only level zero to eliminate impossible candidates is the same as the SEA .

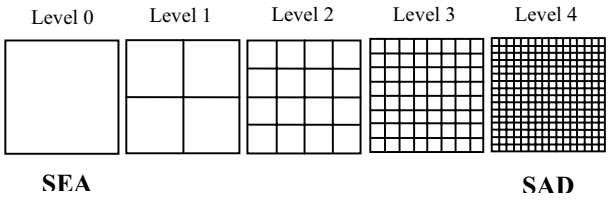

Fig 2: The levels of elimination order in MSEA. Using only level 0 as the elimination criterion is the same as SEA.

4. PROPOSED FAST ME ALGORITHM

In this section, we give the details of the proposed motion estimation algorithm. Our algorithm applies the modified winner update technique in the novel partition order of macrkbolck. The purpose of dividing subblock into 4 smaller subblock is to obtain the tighter boundary. The partition order of macroblock is based on the gradient distribution of macroblock. The flowchart of our ME algorithm is shown in Figure 3.

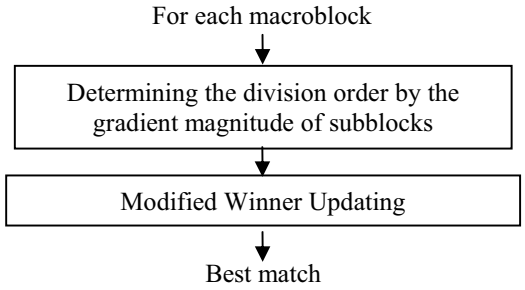

Figure 3: Flowchart of our proposed algorithm.

4.1. Determine the elimination order by the gradient magnitude

The drawback of SEA is that the difference of block sum is not close enough to the SAD. Even we can find the best match as the predicted motion vector, many candidates still can pass the criterion. Dividing a block into subblocks can have a tighter bound [2]. The block sum cannot present the details of block. The block with large variance normally contains more details. To obtain a tighter bound in the early stage, it is reasonable to check the blocks with large variances first. For simplicity we determine the block elimination order by the gradient magnitudes not the variances. For each macroblock, we determine the partition order by the distribution of gradient magnitude of the current macroblock. The block with largest gradient magnitude is divided for checking first.

As shown in Figure 4, each time we select the block with largest average gradient magnitude and then divide it into four subblocks to obtain tighter boundary. If one block

that is flat and has less details, the block sum can represent this block, it is not necessary to divided it. If the gradient magnitude of one block is less then a given threshold, then it will not be divided further. In our implementation, we use a queue to record the elimination order. For each block, we divide it into 4 subblocks and calculate the average gradient magnitudes of these subblocks. The subblocks with their average gradient values greater than a given threshold are pushed into the queue.

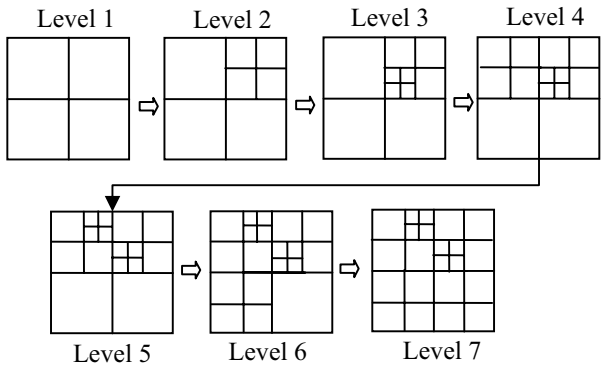

Figure 4: An example of the elimination order at 1st frame of the sequence Coastgaurd. The order is determined by the average gradient magnitude of the subblocks.

The algorithm of determining the elimination order is given as follows:

For each marcorblock, push the largest block into the queue
Repeat
1. *Select the block with max gradient magnitude from the queue.*
2. *Divide the current block into 4 subblocks and calculate their gradient magnitude.*
3. *Check the 4 divided blocks and push each subblock into the queue if its average gradient magnitude is greater than a given threshold.*

Until *the queue is empty*.

Figure 5: Algorithm of determining the block comparison order

4.2. The Modified Winner Update Scheme

In this work, we apply the winner update scheme on the adaptive partition order of macroblock as the previous section described. In this section we used the example shown in Figure 1 to describe how to apply the winner update scheme on the partition order of macroblock. The block sum differences with different candidates are the number on the poker cards and the partition order of macroblock is the flipping order of the poker card. The winner is the one with minimum accumulated sum and the best match of current macroblock is the one who has the minimum distortion with current macroblock. The block sum difference is the distortion and flipping card k means to calculate the block sum difference at the partition order k.

In [5], Zhou and Yu proved that the card showing order can significantly affect the performance of the winner-update algorithm. If the cards of each player are sorted by its number and the card showing order is from the maximum to the minimum, then the efficiency of the winner-update strategy can be further improved. The gradient magnitude is proportional with the distortion [5], so to prevent too many unused computations that caused by unnecessary partition, we used a threshold to restrict the levels of partition. Zhu et al. [4] used 85 levels to early reject impossible candidates by the successive elimination method and in MSEA [2] there are only 4 levels. The partition order and levels in [2] and [4] are two extremely cases. In the proposed method, the partition order of macroblock and level number are determined by the gradient distribution of current macroblock. If one candidate flips the final card, the maximal level of macroblock partition, we skip the unnecessary levels and directly jump to calculate the value of SAD, and then push it back to the candidate pool of winner update to wait the next time selection. The algorithm stop when the selected candidate with SAD as it's low bound. For fast find the candidate with minimum low bound, we used a hash structure to store the low bound value. In the following modified winner update scheme we used the same terminology in [3] named the accumulated distortion as low bound (LB) [3].

1. *Calculate the block sum difference of level 0 of all candidates as the LBs.*
2. *Select one candidate with minimal LB as current_best_candidate from the candidate pool.*
3. *If the level of the current_best_candidate is 1000, goto step6.*
4. *If current_best_candidate reaches the maximal level of macroblock partition, calculate the SAD as LB, set level 1000, and then push current_best_candidate into the candidate pool.*
5. *Update the LB and level, go to step2*
6. *Search the minimum in the bucket that current_best_candidate belongs to.*

Figure 6: Algorithm of the modified winner update scheme.

5. EXPERIMENTAL RESULTS

In the proposed algorithm, we divide a block into 4 subblocks when the gradient energy in this block is large enough in determining the partition order. We use the block sum pyramid [2] technique in our implementation. In our experiments, we used macroblock of size 16x16 pixels, and the search range is within +/- 16 pixels in both horizontal and vertical directions.

The proposed method is compared with the FS and DS algorithms in terms of PSNR values for experiments on three different video sequences as the results summarized in Table 1. Because we need the gradient magnitude to determine the partition order, we need not only the block sum pyramid of the current image but also the block sum pyramid of the gradient map. Table II shows the operation counts of the FS, DS and the proposed method for different sequences. The operation counts of our method contain the operations of building the current frame pyramid, the gradient map and the gradient pyramid. It is obvious that the proposed fast ME algorithm outperforms the DS algorithm and equal to the FS in terms of PSNR for the three test video sequences. In addition, the computational cost of the proposed algorithm is less than the FS and some sequence is near to DS methods. In our experiments, we used 100 frames of the tree test video with QCIF format.

6. CONCLUSIONS

In this paper, we proposed a fast motion estimation algorithm that combines the winner update scheme with the adaptive macroblock partition. The partitioning order is based on the block gradient distribution. Experimental results showed that the proposed algorithm is optimal in terms of PSNR equal to the FS with much better efficiency. In the future, we will extend this technique to the multi-frame motion estimation.

Table 1. Average PSNR values of different algorithms for three different sequences.

	News	stefan	coastguard
FS	38.54	28.18	32.14
DS	38.53	27.86	32.08
FGSE	38.54	28.18	32.14
Proposed	38.54	28.18	32.14

Table 2. Average operation counts of different algorithms for three different sequences.

	news	stefan	coastguard
FS	453637.17	453637.17	498535.95
DS	6796.57	10230.23	8212.95
FGSE	8773.76	18478.12	16434.23
Proposed	7176.03	15207.25	15200.54

ACKNOWLEDGEMENT

This work was supported by MOEA research project under the grant 94-EC-17-A-01-S1-034

REFERENCES

[1] W. Li and E. Salari, “Successive elimination algorithm for motion estimation,” IEEE Trans. Image Process., vol. 4, no. 1, pp. 105–107, Jan. 1995.

[2] X. Q. Gao, C. J. Duanmu, and C. R. Zou, “A multilevel successive elimination algorithm for block matching motion estimation,” IEEE Trans. Image Process., vol. 9, no. 3, pp. 501–504, Mar. 2000.

[3] Y.S. Chen, Y.P. Huang, and C.S. Fuh, “Fast Block Matching Algorithm Based on the Winner-Update Strategy,” IEEE Trans. Image Processing, vol. 10, 2000.

[4] C. Zhu, W. S. Qi, and W. Ser, "Predictive Fine Granularity Successive Elimination for Fast Optimal Block-Matching Motion Estimation," IEEE Transactions on Image Processing, Vol14, Feb, 2005

[5] J.Zhou, J.Li, and S. Yu, “Modified winner-update search algorithm for fast block matching,” Pattern Recognition Letters 25, 2004, p807-p816.

[6] S. Zhu, and K.K. Ma, “A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation,” IEEE Trans. Image Processing, vol. 9, Feb. 2000.

[7] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishinguro, “Motion Compensated Interframe Coding for Video Conferencing,” Proc. Nat. Telecommun. Conf., 1981.

[8] R. Li, B. Zeng, and M.L. Liou, “A New Three-Step Search Algorithm for Block Motion Estimation,” IEEE Trans. Circuits Syst. Video Technol., vol 4, Aug. 1994.

[9] L.M. Po and W.C. Ma, “A Novel Four-Step Search Algorithm for Fast Block Motion Estimation,” IEEE Trans. Circuits Syst. Video Technol., vol. 6, June, 1996.

[10] Chang-Hsing Lee and Ling-Hwei Chen, “A fast motion estimation algorithm based on the block sum pyramid,” IEEE Transactions on Image Processing, vol. 6, no. 11, pp. 1587–1591, 1997.