

An Efficient Fast Full Search Motion Estimation Algorithm for Multiple Reference Frames

G. Komala Yadav, V. Mounika Reddy and T. Krishna Moorthy

Department of Electronics and Communication Engineering,
Sree Vidhyanikethan Engineering College, India

Abstract: The H.264/AVC coding standard allows more than one decoded frame as reference frames to make full use of reducing temporal redundancy in a video sequence. However, motion estimation in multiple frames introduces tremendous computational complexity. This paper presents a fast full search motion estimation algorithm for multiple reference frame motion estimation. The proposed method can be applied to successive elimination algorithm (SEA) or its modified version, multi level successive elimination algorithm (MSEA). We have derived an additional inequality to eliminate the highly impossible candidate blocks in reference frames that are temporally preceding the first reference frame. The experimental results show that the proposed method reduces computational complexity while maintaining same quality as the full search algorithm.

Key words: Successive elimination algorithm • Multi reference frame • Fast full search • Block matching algorithm

INTRODUCTION

The H.264 standard is the video codec developed by the *Joint Video Team* (JVT) [1]. It has achieved a significant improvement in rate-distortion performance relative to past coding standards. Some highlighted features of H.264 are, Variable block-size motion compensation with small block sizes, Quarter-sample-accurate motion compensation and multiple reference picture motion compensation [2]. Multiple reference frame motion compensation allows the encoder to predict a better picture, using several pre-coded and stored pictures. Many conditions demonstrate that multiple reference frames generate better predictions than a system using just one, like repetitive motion, uncovered background, non integer pixel displacement and lighting change [3]. However, the computational complexity of motion estimation increases linearly compared with that of a single reference frame and could be even worse if full search block matching algorithm is applied. Since motion estimation is very effective to exploit temporal redundancy of video signals, the methods that can find out optimal motion vectors speedily are needed. The full

search algorithm finds the optimal motion vectors but requires many computations. Over the last three decades many fast search algorithms have been proposed to reduce the computational complexity. Of these algorithms, some algorithms use only a subset of all the search locations in the search window. The three-step search (TSS) [4], new three step search (NTSS) algorithm [5], the diamond search (DS) [6] and Hexagon-Based Search algorithm (HEXBS) [7] belong to this category. The algorithms in this category do not search all possible search positions, which results suboptimal motion vectors. However, algorithms in the second category reduce the computational load while preserving the minimum matching error as in the FS algorithm. The most popular method to achieve this goal is to use the lower bounds (also called boundary values) of the complete matching error, SAD. This idea was first employed in the successive elimination algorithm (SEA) [8] and has been improved by adopting more lower bounds in the multilevel successive elimination algorithm (MSEA) [9] and the fine granularity successive elimination (FGSE) [10]. The performance of the algorithms in this category depends heavily on the lower bounds. In this paper, we

present an efficient fast full search motion estimation algorithm which is effectively applicable to multiple reference frames. We show that the computational complexity for motion estimation process for the reference frames following the first reference frame can be reduced using the relation between the first reference frame and the other reference frames.

This correspondence is organized as follows: In section II, the SEA with single reference frame is reviewed briefly. In section III, the straight forward extension of SEA for multiple reference frames is reviewed briefly. In section IV, proposed algorithm is presented. The section V shows the experimental results. Finally, we conclude this correspondence in section VI.

SEA with Single Reference Frames: The main idea of successive elimination algorithm is stated as follows. In the early stage, a simple check is done to detect whether a candidate block is better matching candidate block. Then, only the potential candidate blocks are further processed for detailed distortion calculation. Thus, unnecessary computations for impossible candidate blocks can be avoided. Let the size of a block is $N \times N$, the size of search window is $(2M+1) \times (2M+1)$ pixels and $f(p, q, t)$ represents intensity value of the pixel at position (p, q) in frame t . The matching criteria function is the sum of absolute difference (SAD) which is the distortion between two blocks. The SAD between the current block in current frame t and candidate blocks in first reference frame, $t-1$ with motion vector (x, y) is given by eq (1).

$$SAD(x, y, t, t-1) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left| f(p+i, q+j, t) - f(p+i-x, q+j-y, t-1) \right| \quad (1)$$

where (p, q) represents the coordinates of the left-top most corner of the current block. Based on the Minkowski's inequality $\|A\| - \|B\| \leq \|A - B\|$ W. Li and E. Salari derived lower boundary of partial SAD, $P_SAD(x, y, t, t-1)$.

$$P_SAD(x, y, t, t-1) = \left| \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(p+i, q+j, t) - \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(p+i-x, q+j-y, t-1) \right| \leq SAD(x, y, t, t-1), \quad (2)$$

The $P_SAD(x, y, t, t-1)$ is the absolute difference between the sum norms of current block in current frame t and candidate blocks in first reference frame, $t-1$ with motion vector (x, y) .

The SEA algorithm calculates the P_SAD in eq (2) at each search point before calculating the SAD at that search point. This partial SAD is compared with the up-to-date minimum SAD, denoted as SAD_{min} .

If

$$P_SAD(x, y, t, t-1) = |M| - |R| = SAD_{min} \quad (3)$$

Then this search point might be better matching block and processed further for calculating SAD by eq (1). Otherwise that search point will be eliminated from the further calculation. The R denotes $N \times N$ dimensional current block in current frame with left topmost corner (p, q) and M denotes the $N \times N$ dimensional candidate block in reference frame with displacement (x, y) . $|\cdot|$ is the sum norm of a block, for example

$$|R| = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |f(p+i, q+j)| \quad (4)$$

The headings and sub-headings, first with "I. Introduction", appear in upper and lower case letters and be supposed to be set in bold and aligned flush left. All headings from the Introduction to Acknowledgements are numbered sequentially using I, II, III, etc. Subheadings is numbered 1.1, 1.2, etc. If a subsection should be further divided, the numbers 1.1.1, 1.1.2, etc. The font size for heading is 10 points capital letter and bold face and subsections with 10 points and bold. Do not underline any of the headings, or add dashes, colons, etc. (10 Font size).

SEA with Multiple Reference Frames: Now we look at straight forward implementation of SEA for more than one reference frame. Apply SEA to frame $t-1$ and obtain optimal motion vector $(m^*, n^*, t-1)$ for the frame $t-1$. Apply SEA to frame $t-2$, using $SAD(m^*, n^*, t, t-1)$ as an initial minimum SAD. The same procedure is to be applied to all the remaining frames. The constraint on the search process for frames $t-2, t-3$ and so on... will be same as that of constraint on the search process for frame $t-1$. Since the motion vector has to be searched in more than one reference frames the computations significantly increases. Thus the methods have to be found to decreases the

huge computations. We proposed a method that eliminates more blocks in a search window than straight forward SEA without excluding the optimum point.

Proposed Method: Our proposed method can be applied to second reference frame onwards but not for the immediately previous frame of a current frame. In the proposed method we derived an additional inequality as follows

From Minkowski's inequality, $\|A\| - \|B\| \leq \|A - B\|$, we have

$$| |C - M| - |M - R| | \leq |C - R|, \quad (5)$$

where C denotes $N \times N$ dimensional candidate block in reference frame $t-2$ with displacement (x, y) and $|C - R|$ corresponds to the SAD between a current block in frame t and a reference block with displacement of (x, y) in frame $t-2$.

For a current position (x, y) to be potential candidate block, the following has to be satisfied,

$$|C - R| \leq SAD_{min} \quad (6)$$

$$\text{Then, } | |C - M| - |M - R| | \leq SAD_{min} \quad (7)$$

$|M - R|$ corresponds to the SAD between the current block, R and the reference block, M, indicated by the motion vector (x, y) . $|C - M|$ is the SAD between the blocks, C and M, located at the same position $(p+i-x, q+j-y)$ in two reference frames at $t-1$ and $t-2$.

The procedure of our proposed algorithm is as follows. While performing motion estimation in first reference frame, if the inequality in eq (3) satisfied, $|M - R|$ will be computed because the block M is a candidate of the optimal one. In this case, if $|C - M|$ is available by some means, the details of which will be given later, the inequality in eq (3) for frame $t-2$ as well as the inequality

in eq (7) can be used to constrain the search process for C. This is our key idea where the SAD between the blocks in frame t and frame $t-1$ is used to eliminate the search process for the next previous frame $t-2$, providing with another additional inequality in eq (7). On the other hand, if the inequality in eq (3) is not satisfied for frame t , our method will not be applied because $|M - R|$ should be additionally calculated which is not on the ordinary process of SEA.

Fast Computation Of $|C - M|$: C and M are associated to blocks located at a same position but in different frames $t-1$ and $t-2$. Thus, $|C - M|$ is the SAD between those blocks. To develop its fast computation, suppose that the size of an image is $W \times H$. The computation takes following two steps.

- For the whole frames, obtain the absolute difference frame, $d(p, q) = |f(p, q, t-1) - f(p, q, t-2)|$. This requires $W \times H$ sum operations and $W \times H$ absolute operations.
- In this step, compute $|C_{p+1,q} - M_{p+1,q}|$, using $|C_{p,q} - M_{p,q}|$ that have been calculated for the immediate previous search point. $|C_{p+1,q} - M_{p+1,q}|$ requires only 2N sum operations for 2N pixels newly come in $C_{p+1,q}$ and $M_{p+1,q}$ and 2N subtraction operations for 2N pixels gone out from $C_{p,q}$ and $M_{p,q}$.

Experimental Results: In our experiment, the standard video sequences with. avi (176x144) such as "akiyo.avi", "suzie.avi" and. avi (320x240) such as "cricket.avi", were used. We tested 100 frames of the sequences. The akiyo sequence shows a small range of motion, the Suzie sequence shows a medium range of motion and cricket sequence shows large range of motion. The block size was 16x16 pixels ($N=16$). The size of the search window was 31x31 pixels ($M=15$) and only integer values for the motion vectors were considered. The number of reference frames chosen for motion estimation is 2. Experimental results are shown in Table 1. Table 1 shows the number of

Table 1:

Sequence	Akiyo	Suzie	Cricket
Total number of Candidate blocks tested in frame $t-2$	77,43,388	77,43,900	2,56,58,856
Number of candidate blocks rejected in frame $t-2$ with eq (3)	77,41,997	77,36,512	2,56,26,111
Number of candidate blocks passed in frame $t-2$ with eq (3)	1,391	7,388	32,745
Number of candidate blocks rejected in frame $t-2$ with eq (7)	1,272	7,014	8,151
Total Number of operations per block required for motion estimation in frame $t-2$ with proposed method	14,220	50,009	50,432
Total Number of operations required for motion estimation in frame $t-2$ with conventional SEA	71,450	79,950	76,450

candidate blocks rejected in frame $t-2$ at eq (3) and eq (7) for 100 frames. The results show that 95% of candidate blocks which are passed in eq (3) are rejected by eq (7). The total computations required for motion estimation in second reference frame using both the eq (3) and eq (7) are less than the total computations required for motion estimation in first reference frame using eq (3) only.

CONCLUSION

An efficient fast full search algorithm is proposed for motion estimation in multiple reference frames. This method reduces computational complexity by maintaining same quality as the full search algorithm. The experimental results show the efficiency in terms of computations as compared with straight forward SEA algorithm. The algorithm reduces the search space than conventional SEA algorithm in reference frames which are following first reference frame.

REFERENCES

1. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification, ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, 2003.
2. Ostermann, J., J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer and T. Wedi, 2004. Video coding with H.264/AVC: Tools, performance and complexity, IEEE Circuits Syst. Mag., 4(1): 7-28.
3. Su, Y. and M.T. Sun, 2006. Fast multiple reference frame motion estimation for H.264/AVC, IEEE Trans. Circuits Syst. Video Technol., 16(3): 447-452.
4. Koga, T., K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, 1981. Motion compensated interframe coding for video conferencing, Pro. Nat. Telecommun. Conf., New Orleans, pp: G5.3.1-5.3.5, Nov. 1981
5. Li, R., B. Zeng and M.L. Liou, 1994. A new three step search algorithm for block motion estimation, IEEE Trans. on Circuits and Systems for Video Technology, 4(4): 438-442.
6. Zhu, S. and K.K. Ma, 2000. A new diamond search algorithm for fast block matching motion estimation, IEEE Trans. Image Processing, 9(2): 287-290.
7. Ce Zhu, Xiao Lin and Lap-Pui Chau, 2002. Hexagon-Based Search Pattern for Fast Block Motion Estimation, IEEE Trans. on Circuits and Systems for Video Technology, 12(5): 349-355.
8. Li, W. and E. Salari, 1995. Successive elimination algorithm for motion estimation, IEEE Trans. Image Processing, 4: 105-107.
9. Gao, X.Q., C.J. Duanmu and C.R. Zou, 2000. A multilevel successive elimination algorithm for block matching motion estimation, IEEE Trans. Image Processing, 9: 501-504.
10. Zhu, C., W.S. Qi and W. Ser, XXXX. Predictive fine granularity successive elimination for fast optimal block matching motion estimation.