2015 2nd International Conference on Information Science and Control Engineering

A Modified Octagon-based Search Algorithm for Fast Block Motion Estimation

Yuming Wu
Department of Energy and Electrical Engineering
Hunan University of Humanities, Science and Technology
Loudi, P.R. China
chy8370002@gmail.com

Yun Cheng*, Siwen Deng, Wenwen Chen
Department of Computer Science and Engineering
Hunan University of Humanities, Science and Technology
Loudi, P.R. China
xml6677@163.com

Abstract—Based on zero-block detection method in H.264/AVC, the directional characteristic of the SAD(Sum of Absolute Differences Distortion), the center-biased characteristic of motion vectors, and by analyzing the characteristics of OCTBS(Octagon-based Search), MOCTBS (Modified Octagon-based Search) algorithm is proposed in this paper. Firstly, MOCTBS employs small octagon pattern to judge if the MBD (Minimum Block Distortion) point located in the circle of one pixel around the initial search center. If the MBD point does not locate in the one pixel circle around the initial search center or no zero-block was detected, large octagon pattern will be used in the next search step. In order to speed up the motion estimation, zero-block detection method is used in the whole processing of block-matching motion estimation to terminate the search process as early as possible. Experimental results show that MOCTBS can speed up the motion estimation process greatly under the same encoding efficiency compared to OCTBS, especially for those video sequences with simple and slow motion vectors.
.

Keywords-motion estimation; zero-block; detection; octagon; pattern

I. INTRODUCTION

Block-matching motion estimation and compensation can improve the encoding efficiency greatly by eliminating the temporal redundancy information between successive frames and it was adopted by many video-coding standards such as MPEG-1/2/4, H.261, H.263 and H.264/AVC etc. [1] [2]. The most basic block-matching motion estimation algorithm is the full search (FS). Although FS can find the best matching block by exhaustively testing all the candidate blocks within the search window, its computation is too heavy: experimental results demonstrate that the time of the block-matching motion estimation consumed by FS in H.264/AVC is about 60%~80% of the total encoding time. In order to speed up the block-matching motion estimation, many researchers have been working hard for many years and have proposed many kinds of fast block-matching motion estimation algorithms [3]-[15]. Among all those fast motion estimation algorithms, Octagon-based Search Algorithm (OCTBS) is a famous and typical one. So it is very important to study deeply on the OCTBS algorithm.

Based on the analysis of the OCTBS, we proposed a modified octagon-based search algorithm (MOCTBS) for fast block matching motion estimation, in this paper.

The rest of this paper is organized as follows. Section 2 analyzes the OCTBS algorithm. Section 3 describes the MOCTBS. Experimental results are presented in Section 4. Finally, conclusion is given in Section 5.

II. ANALYSIS OF THE OCTBS ALGORITHM

OCTBS is one of the most famous fast block matching motion estimation algorithm. S. Zhu et al. pointed out in [8] that over 50% of those motion vectors are enclosed in a circular area with a radius of two pixels and centered on the position of zero motion as illustrated in Figure 1.

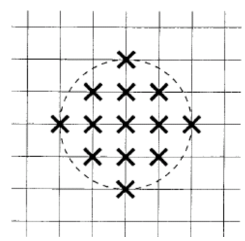


Figure 1. Motion vectors distribution.

The OCTBS algorithm uses two search patterns as illustrated in Figure.2. The first pattern, called large octagon pattern, comprises nine checking points from which eight points surround the center one to form a octagon shape. The second pattern consisting of five checking points composes a smaller octagon shape, called small octagon pattern. In the processing of the OCTBS algorithm, large octagon pattern is repeatedly used until the step in which the MBD (Minimum Block Distortion) point is just located in the center of the large octagon search pattern. Then the search pattern is switched from large octagon pattern to small octagon pattern so as to find the MBD point. Among the five checking points in SOP, the position of the MBD point will provide the motion vector of the best matching block.

By analyzing the OCTBS algorithm, we found that it has the following limitations:

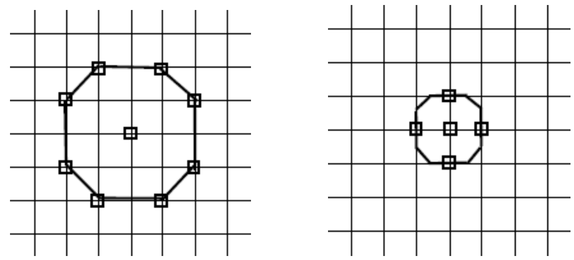
- The OCTBS algorithm doesn't utilize the correlative motion information of the adjacent

*Corresponding author

978-1-4673-6850-6/15 $31.00 © 2015 IEEE
DOI 10.1109/ICISCE.2015.13

blocks, so sometimes it will take a long time to find the MBD point with large motion vector.

- The OCTBS algorithm doesn't make the use of the center-biased characteristic of motion vectors. Even if the motion vector is zero, it will still search thirteen points to find the MBD point, and in this case, the idea number of search points is only five.



(a) Large octagon pattern (b) Small octagon pattern

Figure 2. Two search patterns used in the OCTBS algorithm.

In short, the efficiency of the OCTBS algorithm can be improved. So we proposed a modified octagon-based search algorithm, MOCTBS, which is based on the octagon-based search pattern.

III. THE MODIFIED OCTAGON-BASED SEARCH ALGORITHM

A. *MOCTBS Patterns*

It is well know that the form and the size of the search pattern used in the block-matching motion estimation algorithm will affect directly the speed and the performance of the video encoder. Good search patterns can reduce the searching points remarkably without being trapped into the local minimum point. Considering the block displacement of the real video sequences may be mainly in horizontal and vertical directions and most of the best matching blocks' motion vectors are small or even zero, small octagon pattern (as shown Figure 2 (b)) is used in the proposed MOCTBS algorithm at the beginning of the block matching motion estimation, and if the best matching point is located in the search center, the process of block-matching motion estimation will terminate immediately. Then in order to find the best matching block perhaps with large motion vector, another search pattern consisting of eight checking points forms a octagon shape, called large octagon pattern , as shown in Figure 2(a), is adopted.

In the searching process of block-matching motion estimation, SOP is used to refine the motion vectors and it is necessary no matter how the motion vector being small or big, while LOP is used to locate the best matching block with large motion approximately and it can be discarded if the motion vector is zero.

B. *Using zero-block detection method to speed up the Proposed MOCTBS Algorithm*

If the being searched block is a zero one(i.e. all those coefficients for the block are zero.), the process of motion search can be terminated early, and this will speed up the video encoding greatly. In the inter-frame encoding mode, the motion compensatory data can be prejudged during the process of motion search, which decides whether or not to stop the process of motion search. If the block being searched is a zero-block, the motion search will stop, and the transform and quantization of this zero-block can be skipped. The maximum SAD (Sum of Absolute Differences) of a 4X4 zero-block for H.264/AVC is defined as follows[15].

$$\sum_{u=0}^{3}\sum_{v=0}^{3}\left|x_{uv}\right| < 5\sqrt{2} * Q_{step} / 3 \approx 2.357 Q_{step} \tag{1}$$

where Q_{step} is the quantization step. A total of 52 values of Q_{step} re supported by the H.264/AVC standard and these are indexed by a Quantization Parameter (QP). The values of Q_{step} corresponding to each QP are shown in Table I. Note that Q_{step} doubles in size for every increment of 6 in QP; Q_{step} increases by 12.5% for each increment of 1 in QP. The wide range of quantizer step sizes makes it possible for an encoder to accurately and flexibly control the trade-off between bit-rate and quality.

TABLE I. QUANTIZATION STEP SIZES IN H.264 CODEC

QP	**0**	**1**	**2**	**3**	**4**	**5**
Q_{step}	0.625	0.6875	0.8125	0.875	1	1.125
QP	**6**	**7**	**8**	**9**	**10**	**…**
Q_{step}	1.25	1.375	1.625	1.75	2.25	…
QP	**24**	**…**	**…**	**30**	**…**	**36**
Q_{step}	10	…	…	20	…	40
QP	**42**	**…**	**48**	**…**	**…**	**51**
Q_{step}	80	…	160	…	…	224

Only when the data in a 4x4 block satisfy Eq. (1), the block can be judged as a zero-block, whose transform and quantization can be skipped consequently. It should be noticed that Eq.(1) can only be the threshold of a 4x4 zero-block. But for the encoding of a macro-block, it is allowed to use different block modes. In H.264, the size of a block may be 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 or 4x4. So the threshold should be adjusted to the block mode's changing during the motion search as follows.

$$T_{SAD} = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} | x_{uv} | < M \times N \times 5\sqrt{2} \times Q_{step} / 48 \approx 0.146 \times M \times N \quad (M、N为16,8,4) \tag{2}$$

The motion search method based on the zero-block detection can be presented as follows. The detection is performed by the SAD comparing with Eq.(2) during the motion search, and the motion search will stop when a zero-block is detected, consequently the residual search points, transform and quantization for this zero-block can be skipped. This zero-block detection method only need the SAD comparison in the motion search, and has no effect on the video decoder, and it can be applied to all sorts of the motion searches such as the full search, the three step search, the other fast search and multi-resolution searches such as integer pixel resolution, 1/2, 1/4 pixel resolution etc. So the zero-block detection method is used in the MOCTBS algorithm to speed up the process of motion search.

C. Description of the Proposed MOCTBS Algorithm

The proposed MOCTBS algorithm mainly comprises two stages. In the first stage, in order to reduce the search points for the best matching block with large motion, we use the median motion value of the adjacent blocks (as shown in Figure 3) to predict the motion vector of the current block. The median prediction is expressed as Eq. (3).

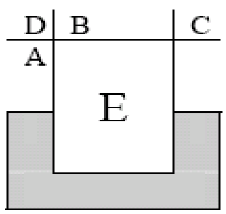


Figure 3. Reference block location for predicting motion vector

$$pred\_mv = median(mv\_A, mv\_B, mv\_C) \qquad (3)$$

In the second stage, the proposed MOCTBS algorithm uses small octagon pattern and large octagon pattern adaptively according to video series with different motion vectors. Based on the assumption that most of the macro blocks in a video of real world would be quasi-stationary or stationary, the MOCTBS algorithm adopts small octagon pattern at the first search step. If the best matching point locates at the search center, the MOCTBS algorithm needs only five search points to find it out. Otherwise, in order to judge whether the best matching motion vector would be large, the MOCTBS algorithm utilizes large octagon pattern at the next search step. If the new MBD(Minimum Block Distortion) point located in one vertex of the small octagon pattern, which indicates that the best matching motion vector would be small, small octagon pattern is repeatedly used until the step in which the MBD locates at the center of the small octagon pattern.

The MOCTBS algorithm is summarized as follows.

Preparation: Use Eq.(3) to predict the initial motion vector of the current block, and set the initial search center point according to the predicted value.

Step1: Test the four or five checking points of small octagon pattern. If the MBD point located at the center of the small octagon pattern, go to **step5**; otherwise, go to **step2**.

Step2: The new MBD point found in the previous search step is re-positioned as the search center. Test the rest checking points of small octagon pattern. If the MBD point located at the center of the small octagon pattern or a zero-block is being detected, go to **step5**; otherwise, go to **step3**.

Step3: The new MBD point found in the previous search step is re-positioned as the search center. Test the checking points of large octagon pattern. If a zero-block is being detected, goto **step5**; If the new MBD point still located in one vertex of small octagon pattern, go to **step4**; otherwise, go to **step2**.

Step4: The new MBD point found in the previous search step is re-positioned as the search center point. Test the two or three checking points of small octagon pattern. If the MBD point located just in the center of small octagon pattern, go to **step5**; otherwise, recursively repeat this step.

Step5: Stop searching. The MBD point is the final solution of the motion vector which points to the best matching block.

D. Analysis of the Proposed MOCTBS Algorithm

For block-matching motion estimation, computational complexity could be measured by average number of search points required for motion vector estimation. According to the statistical distribution law of motion vectors in different video sequences, assume that the best matching point locates just in a circle area with a radius of one pixel around the initial search point, the least search points needed for OCTBS, DOS and MOCTBS are listed in Table II.

TABLE II. COMPARISON OF LEAST SEARCH POINTS NEAR THE INITIAL SEARCH CENTER FOR OCTBS, DOS AND MOCTBS

	the best matching point is located in the	
	center	***Circular area with a Radius of 1 pixel***
OCTBS	13	13
DOS	5	8
MOCTBS	⩽5	⩽8

From Table II we observe that the least search points needed for MOCTBS is always less than that of OCTBS, the reduced search points is always five to eight.

If the best matching point located outside the circular area with a radius of one pixel, the search points needed for MOCTBS will be great different between the video series with different motion extents.

IV. EXPERIMENTAL RESULTS

Our proposed MOCTBS algorithm was integrated within version 18.5 of the H.264 software [16], and it is compared versus FS, OCTBS [9] and DOS [14]. Even though many image sequences are tested in the experiment, only four of them are selected out to be compared. The CABAC (Context-Adaptive Binary Arithmetic Coding) entropy coder [17] was used for all of our tests, with quantization parameter (QP) values of 28, 32, 36, and 40, a search range of ± 16, and 1 reference frame.

The four selected sequences are container (Common Intermediate Format, CIF), news(CIF), carphone (CIF), and mobile(CIF). The former 21 frames of every sequence are tested, and only the first frame was encoded as I-frame, while the remainders are encoded as P-frames. Although H.264/AVC provides seven different block-sizes for inter-frame coding, we have only used the 16×16 mode so as to compare the speed of motion search algorithm accurately. To simplify our comparison, we have used ASP(Average number of Search Points per block) and RD(Rate Distortion) performance plot as shown in Table III and Figure 4 respectively.

TABLE III. THE AVERAGE NUMBER OF SEARCH POINTS PER BLOCK

		FS	OCTBS	DOS	MOCTBS
container	QP=28	1089	13.010	5.201	4.179
	QP=32	1089	13.017	5.146	4.133
	QP=36	1089	13.004	5.077	4.065
	QP=40	1089	13.000	5.045	4.036
news	QP=28	1089	13.131	5.680	4.533
	QP=32	1089	13.139	5.667	4.534
	QP=36	1089	13.131	5.587	4.459
	QP=40	1089	13.132	5.544	4.451
carphone	QP=28	1089	13.795	9.253	8.433
	QP=32	1089	13.339	8.863	7.939
	QP=36	1089	13.775	8.481	7.516
	QP=40	1089	13.659	8.006	7.107
mobile	QP=28	1089	13.347	8.043	7.265
	QP=32	1089	13.339	8.091	7.267
	QP=36	1089	13.368	8.158	7.348
	QP=40	1089	13.365	8.238	7.410

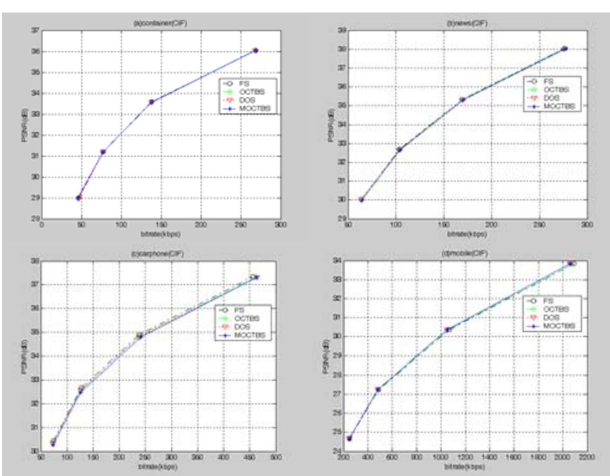


Figure 4. RD performance plot for sequences (a) container, (b) news, (c) carphone, (d) mobile

From Table III we can observe that the average number of search points per macro-block needed for OCTBS, DOS and MOCTBS are 13.000~13.795, 5.045~9.253 and 4.036~8.433 respectively. It's obvious that MOCTBS is faster than FS and OCTBS. From Figure 4 we can observe that the encoding efficiency of MOCTBS is almost the same as that of OCTBS.

V. CONCLUSION

Based on the directional characteristic of SAD distribution and the center-biased characteristic of motion vectors, a fast block-matching motion estimation algorithm, MOCTBS, is proposed in this paper. MOCTBS employs small octagon pattern to refine the motion vector, and large octagon pattern to locate the best matching block with large motion vector approximately. Although the proposed MOCTBS may also be trapped in local minima, the experimental results show that it is faster than OCTBS and DOS, while its encoding efficiency is almost the same as that of OCTBS.

ACKNOWLEDGMENT

This work was partly supported by the Ministry of Education's Research Based Learning and Innovation Experimental Program of China, the Hunan Provincial Research Based Learning and Innovation Experimental Program of China, the Hunan Provincial Natural Science Foundation of China (12JJ2040), the Construct Program of the Key Discipline in Hunan Province, China, the Aid program for Science and Technology Innovative Research Team in Higher Educational Institute of Hunan Province, China.

REFERENCES

[1] K. R. Rao and J. J Hwang, "Techniques and Standards for Image, Video and Audio Coding", Englewood Cliffs, NJ: Prentice Hall, 1996

[2] T. Wiegand and G. Sullivan, "ITU-T Rec. H.264|ISO/IEC 14496-10 AVC", Final Draft, Document JVT-G050, 7th Meeting: Pattaya, Thailand, March 2003

[3] M. Ghanbari, "The cross-search algorithm for motion estimation", IEEE Transaction on Communication, July 1990, vol. 38, pp. 950–953.

[4] L. W. Lee, J. F. Wang, et al., "Dynamic search-window adjustment and interlaced search for block-matching algorithm", IEEE Transaction on Circuits and Systems for Video Technology, 1993, Vol. 3, pp.85–87.

[5] R.Li, B. Zeng, et al., "A new three-step search algorithm for block motion estimation", IEEE Transaction on Circuits and Systems for Video Technology, Aug. 1994, vol. 4, pp. 438–442.

[6] L. M. Po and W. C. Ma., "A novel four-step search algorithm for fast block motion estimation", IEEE Transaction on Circuits and Systems for Video Technology, June 1996, vol. 6, pp. 313–317.

[7] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding", IEEE Transaction on Circuits and Systems for Video Technology, Aug. 1996, vol. 6, pp. 419–423.

[8] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation", IEEE Transaction on Image Processing, Feb. 2000, vol. 9, pp.287–290.

[9] C. Zhu, X. Lin, and L.P. Chau, "Hexagon-based search pattern for fast block motion estimation", IEEE Transaction on Circuits and Systems for Video Technology, May 2002, vol.12, pp.349-355.

[10] L.P. Chau and C. Zhu, "A fast octagon-based search algorithm for motion estimation", Signal Processing, March 2003, Vol. 83, Issue 3, pp. 671-675.

[11] Y. Cheng, Z.Y. Wang, et al., "A fast motion estimation algorithm based on diamond and triangle search patterns", IbPRIA 2005, LNCS 3522, pp.419-426.

[12] Z. Chen, Y. He, and J. Xu, "Hybrid unsymmetrical-cross multi-hexagon-grid search strategy for integer pel motion estimation in H.264," in Proc. Picture Coding Symp., Saint Malo, Apr. 2003, pp. 17-22.

[13] C. U. Jeong, T.Ikenaga, and S. Goto, "An extended small diamond search algorithm for fast block motion estimation," in Proc. The 23rd International Technical Conference on Circus/Systems, Computaters and Communications, Shimonoseki City, Yamaguchi Pref., Japan, July 6-9, 2008, pp.1037-1040.

[14] M.L.Xiao, M.S.Deng, "A Fast Motion Estimation Algorithm Based on Diamond and Octagon Search Patterns", Engergy Procedia 13 (2011), pp.2931~2937.

[15] A.B.Jambec, A.A.Juri, "Low-energy motion estimation architecture using quadrant-based multi-octagon (QBMO) algorithm", Journal of Real-Time Image Processing, May 2014, at http://link.springer.com/article/10.1007%2Fs11554-014-0426-x#page-1

[16] Joint Video Team (JVT), "Test Model JM18.5", December 2014 at http://iphome.hhi.de/suehring/tml/download/old_jm/

[17] D. Marpe, H. Schwarz and T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in H.264/AVC Video Compression Standard", IEEE Transaction on Circuits and Systems for Video Technology, July 2003, vol.13, pp. 620–636.