Fast Motion Estimation Methods for HEVC

Xufeng Li, Ronggang Wang, Wenmin Wang, Zhenyu Wang, Shengfu Dong

***Abstract*—Compared with H.264, the High Efficiency Video Coding standard has higher compression efficiency and better transmission performance, and will be used in wider field. TZ search method is adopted to perform integer pixel search in HEVC. Its performance is very perfect but very time-consuming. In this paper, we use hexagon search method in place of TZ search method to speed up motion estimation process. Simulation results show that this scheme accelerates 2-6 times and the performance loss is negligible in sequences with slow movements.**

***Index Terms*—HEVC, TZ, hexagon search, integer pixel search**

I. Introduction

HEVC, the High Efficiency Video Coding standard, is recently issued jointly by ISO/IEC MPEG and ITU-T VCEG to satisfy the demanding for higher and higher quality of video images. It doubles the video compression ratio compared to H.264/MPEG-4 AVC at the same level of video quality[1]. On the other hand, compared with H.264, the encoding complexity of HEVC is increased up to 2-10 times. Therefore, to reduce the encoding complexity with little loss of coding performance is critical for HEVCs application.

Motion estimation which is the most important part of video compression, is the major factor in the compression efficiency and occupies a large part of the whole coding arithmetic. Because block matching algorithm and compensation technique is simple and easy to be realized, it is widely used in video processing. In block matching algorithm, the current frame is firstly divided into a series of macro-blocks, which are fixed-size square blocks, and the motion vector for each macro-block will be estimated by finding the closest block of pixels within its search range in a reference frame according to some matching criteria[2]. The full search block matching algorithm is widely used because of its simplicity and accuracy in which all the candidate blocks within a pre-defined search area in a reference frame to obtain the optimal performance. To overcome the huge complexity burden of FS, fast estimation algorithms are suggested[3].

HM is the reference software being made available to provide a reference implementation of HEVC. It adopts TZ search method as the fast integer pixel motion estimation method. TZ search method performs well by combining diamond search, square search and raster search methods. However, it is still time-consuming and needs some improvements. As a result, other faster search algorithm is carried out to replace the TZ search algorithm.

The present paper proposes using hexagon search algorithm to replace TZ search method in HM to decrease the encoding time without degrading the video quality. The TZ search method in HM is introduced in Section 2. Section 3 details the hexagon search method process. Section 4 presents the simulation results and discussion. Finally, Section 5 draws some concluding remarks.

This work was partly supported by the grant of National Science Foundation of China 61370115, Shenzhen Basic Research Fund and Shenzhen Peacock Plan.

Xufeng Li, Ronggang Wang, Wenmin Wang, Zhenyu Wang, Shengfu Dong are with the National Engineering Laboratory of Video Technology, School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen 518055, China. (E-mail: lixufeng134@126.com, rgwang@pkusz.edu.cn, wangwenmin@pkusz.edu.cn, wangzhenyu@pkusz.edu.cn, sfdong@jdl.ac.cn)

II. TZ Search Method

Motion Estimation is an essential process in HEVC. It finds the best matched block position in the past (or future frames) for every block in the current video frame. Full search algorithms searching all the blocks in reference frame search window can find the most accurate matching block, but they are also too time-consuming[4]. So fast motion estimation algorithms searching only blocks which are likely to be the best matched block position is widely used. TZ search method is adopted as the fast integer pixel motion estimation method in HM. The flowchart of the method is shown in Fig .1. It has four steps as described in the following[5]:

1. *Start Search Center*: Establish a set of search centers, including the motion vector obtained from median prediction, the motion vector of the left, the up and the upper right position in the corresponding block of the reference frame, the motion vector at(0,0) position. Choose the point which has the smallest matching error as search center of next step.

2. *Diamond or Square Search*: Determine the searchrange and the search pattern. Run the search with different stride lengths from 1 through 64, in multiples of 2, if the searchrange=64, as shown in Fig .2. Then performing 2 point search to check only the 2 untested points if the optimal point is around the search center with stride length being 1. Find out the smallest matching error point as search center of step 3.

3. *Raster Search*: If the distance between the optimal point obtained from step 2 and current search center called best distance is 0, stop the search. Otherwise, if it is greater than the value of iRaster which is set appropriately, raster search is performed and the value of iRaster is used as the stride length of raster search.

4. *Raster/Star Refinement*: Set the optimal point from step 3 as the starting point. Raster refinement performs 8 point diamond or square search with the best distance decreasing according to the exponential of 2 and updating every step of the current starting point location until the best distance is 0. The star refinement is similar to step 2 except for the optimal point being the starting point every round. The refinement process will only start if the best distance is greater than zero.

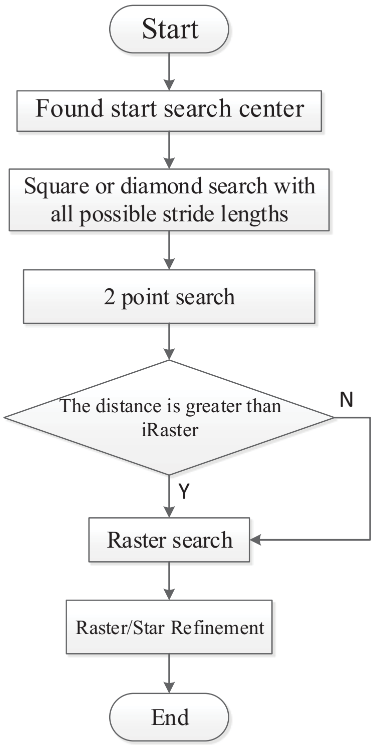
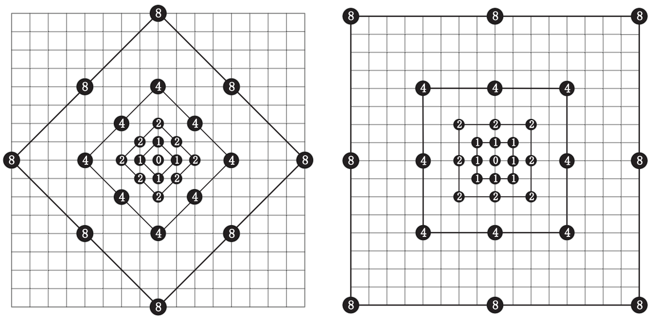


Fig. 1. Flowchart of the TZ search method



(a) Diamond search pattern (b) Square search pattern

Fig. 2. Diamond search pattern in (a) and Square search pattern in (b)

When the best distance equals to 0, the search stops.

However, time performance of TZ search method is still poor, although the search result turns out to be very good. Some other fast search methods should be considered to replace it.

III. HEXAGON SEARCH METHOD

To improve the search efficiency, the number of search points and the computation time should be reduced with little performance loss. Hexagon search method is known for its good performance and high efficiency. Traditional hexagon-based search pattern is depicted in [6]. In the first step, a start search center should be chosen as the TZ search method does. Then large hexagonal pattern with six checking points around the search center is used for the first search round. If the optimum is found at the center, we switch to use small hexagonal pattern depicted in fig.3, including four checking points for the focused inner search. Otherwise, the search continues around the point with minimum block distortion (MBD) using the same large hexagonal pattern. Note that only three nonoverlapped checking points will be evaluated as candidates each time after the first round as shown in Fig.4. Small hexagonal pattern cant find the possible MBD point at four corners, so square refinement for hexagon search method is adopted to replace it to make up for the defects. Fig.5 shows how the refinement process works.

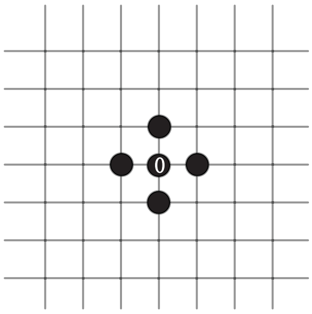


Fig. 3. Small hexagonal pattern

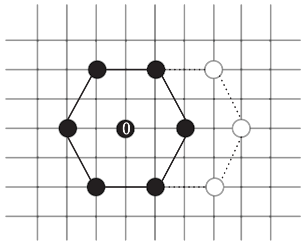


Fig. 4. Large hexagonal pattern

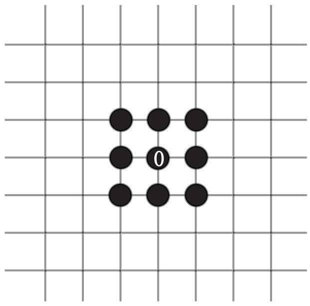


Fig. 5. Hexagon refinement

The search with different stride lengths in TZ search method uses square or diamond search pattern, and 8 points will be evaluated for every round (the first round of diamond search pattern is 4). The number of search points is at least 52 if the stride lengths are 1 to 64. However, statistical result indicates that large hexagonal pattern is implemented no more than nine times, 30 (=6+38) candidate points. Besides, raster search step is more time-consuming and takes approximately 75% of the TZ search time, while square refinement of hexagon search method evaluates only 8 points.

TABLE I
COMPARE RESULTS OF AVERAGE SEARCH POINTS

Sequence	APTZ	APHEX	HEX/TZ	Sequence	APTZ	APHEX	HEX/TZ
Kimono	36.25	18.75	52%	PartyScene	75	17.25	30%
ParkScene	24.5	16.75	68%	RaceHorses (832x480)	146.75	19.5	23%
FourPeople	34.5	16	46%	BlowingBubbles	75.75	17.5	52%
KristenAndSara	32.75	16	49%	RaceHorses(416x240)	150	19.5	15%

IV. SIMULATION RESULTS

In order to evaluate the efficiency of hexagon search method, HM10.0 is used as the software platform. The simulations are carried on Window Server 2012 R2 OS platform with Intel E5-2670 0@2.60 GHz CPU and 4GB RAM. Four kinds of resolution sequences are tested in standard low delay configure and each resolution choose two sequences(1920x1080: Kimono, ParkScene, 1280x720: FourPeople, KristenAndSara, 832x480: PartyScene, RaceHorses, 416x240: BlowingBubbles, RaceHorses). Table I shows the statistical result of the average search points of each block. The data is gotten from the mean of 100 frames of each sequence under four QPs. APTZ means the average search points of each block using TZ search method and APHEX the HEX search method. HEX/TZ is the proportion of APHEX relative to APTZ. BD-rate and time cost between hexagon and TZ search method is depicted in II. AR means the acceleration rate of hexagon search algorithm related to TZ search algorithm. BD-rate(Y) is the performance loss of Y component. The result of each sequence is the average value under four QPs. Fig.6 shows RD performance for eight sequences at the last page.

We can observe that candidate search points are saved up to 85% at most. TZ search method can be accelerated about 1.7-6.5 times by our proposed method and BD-rate loss is under 2% when the movement is slow. For Sequence RaceHorses with large motion intensity, the BD-rate loss grows a lot up to 3.1% and 2.2% respectively. Its obvious that hexagon search method is faster than TZ search method, and the encoding efficiency is almost the same in the case of slow movements.

TABLE II
COMPARE RESULTS OF TWO FAST MOTION ESTIMATION METHODS

Sequence	AR	BD-rate(Y)
Kimono(1920x1080)	3.34	0.6%
ParkScene(1920x1080)	3.25	0.6%
PartyScene(832x480)	3.35	1.2%
RaceHorses(832x480)	4.31	3.1%
BlowingBubbles(416x240)	1.94	0.4%
RaceHorses(416x240)	6.56	2.2%
FourPeople(1280x720)	2.22	0.7%
KristenAndSara(1280x720)	1.72	0.5%
average	3.34	1.2%

V. CONCLUSION

Hexagon search method has done very well in slow motion intensity situation with 2 to 6 times acceleration rate and negligible performance loss. For sequences with large motions, the performance loss is larger (2%-3%). Therefore, the search method can be adopted by adding judgment conditions which is used to judge the motion intensity.

REFERENCES

[1] G. J. Sullivan, J. Ohm, H. Woo-Jin, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, 2012, circuits and Systems for Video Technology, IEEE Transactions on.

[2] W. Xuzhi, W. Wanggen, and M. Yanru, "A novel hexagon diamond search algorithm for motion estimation," in *Audio Language and Image Processing (ICALIP), 2010 International Conference on*, ser. Audio Language and Image Processing (ICALIP), 2010 International Conference on, Shanghai, 2010, pp. 1489–1493.

[3] H. Kyuseo and C. Byungtae, "Adaptive hexagon search pattern for block motion estimation," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, ser. Systems, Man and Cybernetics, 2003. IEEE International Conference on, vol. 2, 2003, pp. 1406–1409 vol.2.

[4] N. Purnachand, L. N. Alves, and A. Navarro, "Fast motion estimation algorithm for hevc," in *Consumer Electronics - Berlin (ICCE-Berlin), 2012 IEEE International Conference on*, ser. Consumer Electronics - Berlin (ICCE-Berlin), 2012 IEEE International Conference on. Berlin: IEEE, 2012, pp. 34–37.

[5] X.-l. Tang, S.-k. Dai, and C.-h. Cai, "An analysis of tzsearch algorithm in jmvc," in *Green Circuits and Systems (ICGCS), 2010 International Conference on*, ser. Green Circuits and Systems (ICGCS), 2010 International Conference on. Shanghai: IEEE, 2010, pp. 516–520.

[6] Z. Ce, L. Xiao, and L.-P. Chau, "Hexagon-based search pattern for fast block motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 12, no. 5, pp. 349–355, 2002, circuits and Systems for Video Technology, IEEE Transactions on.

[7] S. Wang, S. Sun, W. Shen, W. Xu, and Y. Zheng, "Improved strategy for tzsearch algorithm."

[8] C. Yun, Y. Lin, F. Zhiwen, H. Hailiang, and C. Ganxin, "A fast motion estimation algorithm based on diamond and hexagon search patterns," in *Pervasive Computing (JCPC), 2009 Joint Conferences on*, ser. Pervasive Computing (JCPC), 2009 Joint Conferences on, Tamsui, Taipei, 2009, pp. 595–598.

[9] H. Wenwei and Z. Yuling, "Improved hexagon-based searching algorithm for fast motion estimation," in *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, ser. Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on, Chengdu, 2010, pp. 1–3.

[10] L. Yufeng, X. Jufei, and W. Wei, "Motion estimation based on h.264 video coding," in *Image and Signal Processing (CISP), 2012 5th International Congress on*, ser. Image and Signal Processing (CISP), 2012 5th International Congress on, Chongqing, 2012, pp. 104–108.

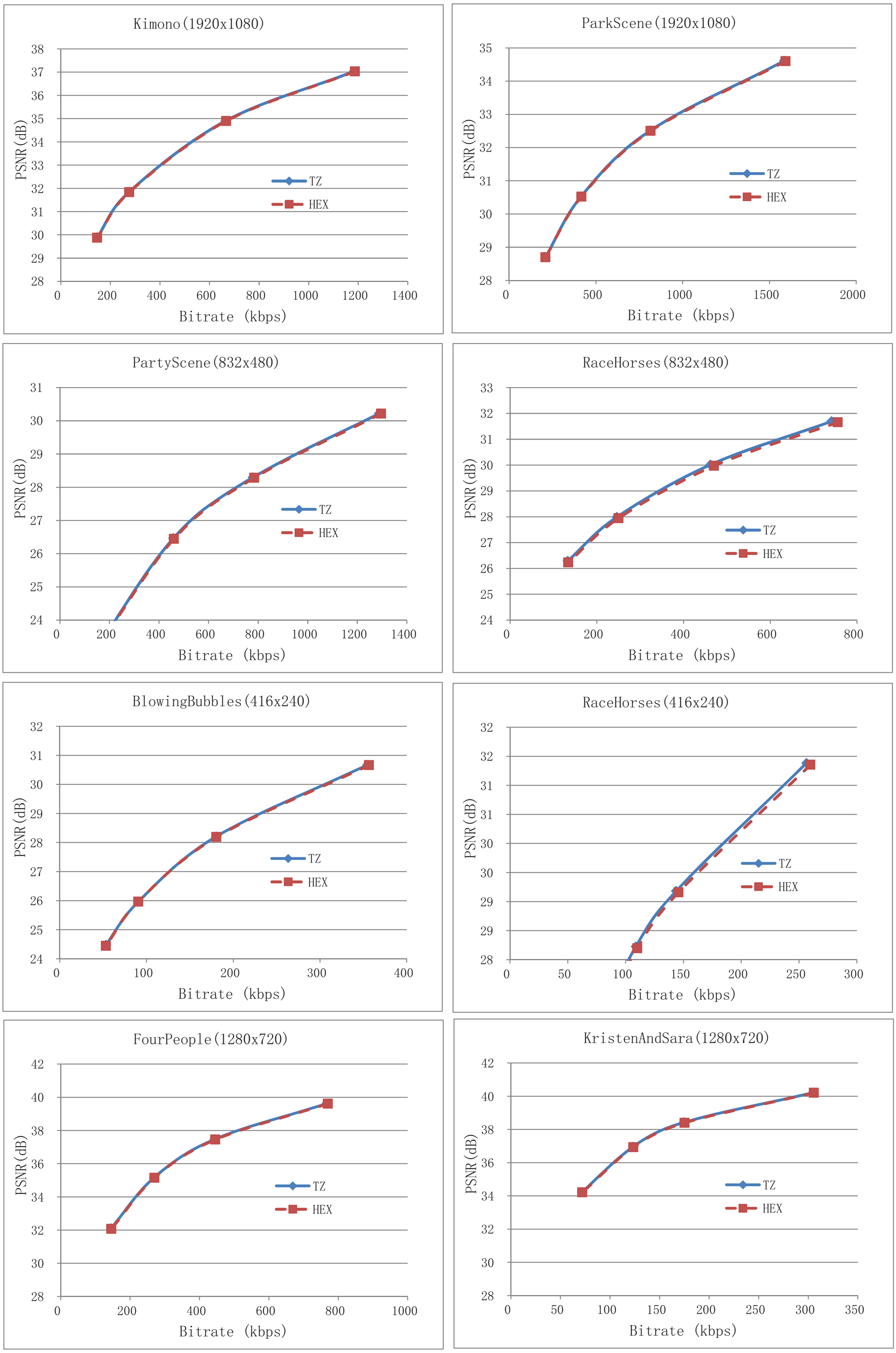


Fig. 6. RD performance for eight sequences