A NOVEL ANGLE-RESTRICTED TEST ZONE SEARCH ALGORITHM FOR PERFORMANCE IMPROVEMENT OF HEVC

Niras C Vayalil[⋆] *Manoranjan Paul*[†] *Yinan Kong*[⋆]

[⋆] Department of Engineering, Macquarie University, Sydney, Australia
[†] School of Computing and Mathematics, Charles Sturt University, Bathurst, Australia
niras.cheeckottu-vayalil@mq.edu.au, mpaul@csu.edu.au, yinan.kong@mq.edu.au

ABSTRACT

High Efficiency Video Coding (HEVC) is the latest video encoding standard and has approximately 50% bit-rate saving compared to its predecessor. However, the motion estimation (ME) is considerably complicated by the incorporation of varieties of partitioning modes and a quad-tree based coding structure, and also by increasing the basic coding unit size by a factor of 16. Motion estimation is the most complex task in the video encoding process, consuming 60-80% of overall encoding time. This paper proposes a new algorithm, angle-restricted test zone (ARTZ) for motion estimation which is based on a test zone (TZ) search, exploiting directional probabilities of motion vector search. In our experiments, this proposal achieves a time saving in motion estimation of about 20% to 50% compared to a TZ search in the HEVC test model (HM) implementation for UHD videos without significant degradation of PSNR.

***Index Terms*—** Motion Estimation, Test Zone Search, High Efficiency Video Coding, Fast-search Algorithms.

1. INTRODUCTION

High efficiency video coding (HEVC/H.265) [1] is a video encoding standard proposed by the Joint Collaborative Team on Video Coding (JCT-VC), a collaboration between ISO/IEC MPEG and ITU-T VCEG. HEVC can achieve an equivalent subjective reproduction quality with approximately 50% less bit rate on average than its predecessor [2], advanced video coding (AVC/H.264) [3]. A primary driving factor to improve video coding efficiency is the recent high demand for ultra high definition (UHD) video content in consumer devices. It is expected that video traffic will be 82% of all consumer Internet traffic by 2020, a 70% increase from 2015 [4]. The coding efficiency along with the computational complexity increases with every new proposal as in the case of MPEG2 and AVC [5]. In HEVC the size of the basic coding unit (CU) block is 64×64 whereas it is 16×16 for its predecessor, and also introduced quad-tree structuring as well as a variety of partitioning types. Larger block sizes are useful for compression as it could efficiently compress a larger area, in particular for high-resolution videos, where smoother or similar regions may be found in larger blocks.

Motion estimation is the most computationally intensive task in video compression, consuming 60% to 80% of the total encoding time [6]. Due to this high complexity in ME, numerous algorithms are proposed to tackle its complexity [7–9]. The test zone (TZ) algorithm is one of the best fast-search algorithms, providing a good rate distortion (RD) and a 60% improvement in encoding time compared to full-search algorithms. There are several proposals for TZ search improvement by reducing the number of search points, by changing the search pattern of its initial grid search to hexagon, pentagon or similar [10–12] instead of a diamond/square pattern. However, it seems that they suffer considerable PSNR (peak signal-to-noise ratio) degradation of from 0.026 dB to 0.137 dB. The proposal in [13] avoids searching after a certain distance in the initial grid search of the TZ algorithm, assuming a monotonic error surface, but generally this assumption is not valid.

This paper proposes a new algorithm; angle-restricted test zone (ARTZ) search, based on the directional properties and probability of finding global minima in a search for motion vectors. The experimental observations reveal that ultimate motion vectors are in proximity of predicted motion vectors and the searching of motion vectors follows certain directions. The proposed method mainly targets ultra-high-definition (UHD) videos as they requires larger computations for motion estimation than high definition (HD) or lower resolution videos. The algorithm is based on the TZ search, a preferred algorithm in HEVC, also implemented in HEVC reference software, HEVC test model (HM) [14]. This proposed method considerably reduces search points by discarding the least-probable locations to find minima, thus achieving a speed-up of computation of ME of from 20% to 50% compared to standard TZ search depending on the nature of the video.

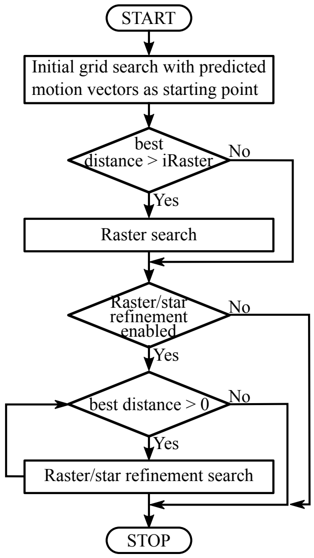

Fig. 1. Flow chart for the Test Zone Search Algorithm.

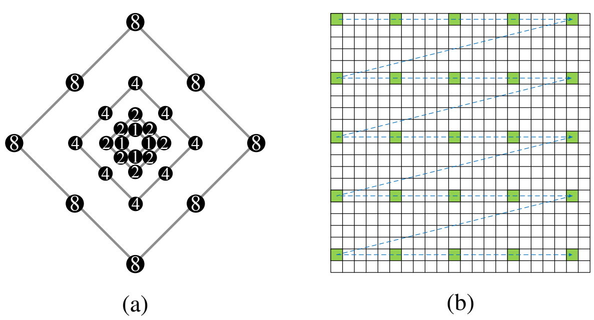

Fig. 2. Search pattern for initial grid search and raster search in test zone (TZ) search algorithm: (a) diamond pattern for initial grid search, (b) raster scan pattern.

2. OVERVIEW OF TEST ZONE SEARCH ALGORITHM

A flow chart of the TZ search algorithm is given in Fig. 1. The TZ search starts with an initial grid search, either using a diamond or square search pattern with different stride lengths ranging from 1 to 'search length' in multiples of two. An 8-point diamond grid pattern used for an initial grid search is shown in Fig. 2 (a). The search starts with a predicted motion vector as its search center. Initial grid searches find a minimum distortion point, usually using a sum of absolute differences (SAD) [15] as a block matching criterion, because SAD is the simplest of the various matching criteria. If the distance from the search center to this minimum distortion point (best distance) is greater than a predefined value, called 'iRaster' in HM, then the algorithm does a raster search; otherwise, skip this step. The raster search is a kind of full-search algorithm where the search window is sub-sampled by a factor equal to 'iRaster', which is set at compilation time. A raster-search pattern with an 'iRaster' equal to 5 is shown in Fig. 2 (b).

Table 1. Percentage of raster searches in a TZ search

Video Name	Frame No.	No. of Raster searches (%)
ToddlerFountain [a]	20	14.99
DrivingPOV [a]	29	0.87
TunnelFlag [a]	27	7.92
Crosswalk [a]	22	5.22

[a] 4K UHD video, encoded as P-frames using 4 reference frames

The next step in this TZ search algorithm is a raster/star refinement if enabled. Generally, only one of them is enabled to speed-up computation. In this step, the algorithm makes a fine adjustment to the motion vector obtained from the previous steps. Both refinements use a diamond or square pattern, and they differ in their search operation. In the raster scan, the stride length of the search pattern is halved in every step and it also changes the search center, whereas the star refinement is similar to the initial grid search except changing its starting point to the minimum distortion point in every round. The refinement search stops when the best distance becomes zero.

3. PROPOSED ALGORITHM

Table 1 shows the number of raster searches involved in the TZ search algorithm for different 4K UHD resolution videos when encoded as P-frames using 4 reference frames. Although the error surface for the motion search is not monotonic, from observing various videos it seems that most of the motion vectors are not too far from the predicted location. This is evident from the table, as in some cases only 0.87% of motion searches involve a raster search, where the criterion for doing a raster search is that the distance to the minimum error location from the search center is larger than 5 pixels. In other words, the majority of motion vectors are found within a 5-pixel radius of the predicted location. Another important property from our observations is that the minimum error location in each step of the TZ motion search algorithm moves in a single direction or does not change its movement direction drastically. In this experiment, we traced the movement of the minimum point in each step of the TZ search using HM reference software. The first 30 frames of the 4K UHD videos were encoded using configuration file 'encoder_lowdelay_P_main.cfg', which encodes all frames except the initial one (I frame) as P frames with 4 reference frames. The motion vectors which involve a raster scan have a particular interest because others (motion vectors found without raster scan) are in the close vicinity of the predicted motion vector, and thus we do not expect any directional properties for their minima movements. In the TZ search algorithm, if the minimum is found outside of the predefined 'iRaster', the next step involves a raster search, which is equivalent to a full search where the search window

Table 2. Percentage of first minimum error locations occurs after the 'iRaster' (5 pixels) distance

	% of first minima at the distances					
Video Name	8	16	32	64	128	256
ToddlerFountain, frame no. 25	25.1	44.5	16.0	7.9	4.3	2.2
DrivingPOV, frame no. 27	51.8	35.9	7.9	2.8	1.1	0.5
TunnelFlag, frame no. 19	20.2	45.4	14.9	9.4	6.3	3.8
Crosswalk, frame no. 25	27.5	48.6	11.8	6.6	3.6	1.9

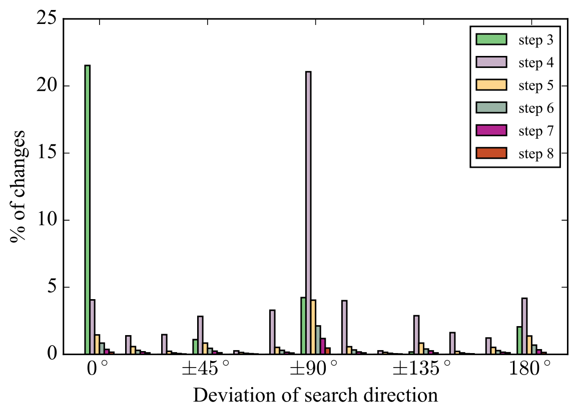

Fig. 3. Change of search direction in each step of initial search in the TZ Search Algorithm. This image shows the location of minima changes in the 22nd frame of the Netflix_Crosswalk video sequence.

is sub-sampled by a factor equal to 'iRaster', i.e. searching blocks in increment of 'iRaster' pixels in both horizontal and vertical directions. The initial searches, involving raster scan searches, are monitored for various video sequences and the point where the first minimum occurs after an 'iRaster' distance is noted, and the percentages of such incidents are tabulated in Table 2. The raster search modifies or finds a better minimum if the initial search lies outside of 'iRaster' because it covers all the blocks within the 'iRaster' distance. For this reason the proposal in [13] entirely avoids a search after the 'iRaster' distance in the initial grid search of the TZ search algorithm. This assumption is only valid for a monotonic error surface; that is not the case generally, thus there is a good probability of finding a minimum after the 'iRaster' distance even though the error is increasing till the 'iRaster' distance (or even more than 'iRaster distance'), which is evident from Table 2. In some video sequences, approximately more than 80% of minimum locations first arise after a distance of 8 pixels; if these points are skipped 80% of raster scans would not happen in the TZ search algorithm for some cases, which affects finding global minimum error locations and predictions of further blocks. In these observations, it is also found that in several cases the minimum location updates follow the direction of previous updates. Fig. 3 shows search directional changes in the initial search of the TZ algorithm for the 22nd frame of the Netflix_Crosswalk video sequence. Only searches having minima outside of the 'iRaster' distance are used for the plot. From the figure, it is clear that large number of searches (28.41% in total) follow the previous direction (0°) especially at the 3rd step of the initial diamond search. In the 4th and 5th steps, 21.1% and 4% respectively are at ±90°. After this step (i.e. distance larger than 32 pixels) only a few minima are found, approximately equal to 10% in total. In a similar way, the videos in Table 2 also analyzed; even though the results vary, they all follow a similar pattern for the probability of finding minima in the search steps. These videos are selected because of having high motion content in 4K UHD resolution. It is obvious that higher resolution and high motion demands more computations, also the motion estimation characteristics of 4K UHD videos differ from the lower resolutions.

The algorithm was developed based on the above results; the search points are restricted to certain locations after analyzing the probability of each point in the diamond-grid search. Search all points at step 4 (16 pixel distance), and search restricted to 0°, ±90° and 180° points for step 3 (8 pixel distance), and 0° and ±90° points for step 5 (32 pixel distance). No other points in the initial diamond search seem to have a higher probability of finding a first minimum; thus they are omitted from the initial grid search, hence saving considerable computational effort for motion estimation. These changes reduce the computational effort for the initial grid search by approximately 56% (HM uses a 16-point diamond grid if the distance is more than 8 pixels) if the search distance is set at 64 pixels, but the TZ search has a raster search in certain cases, which is the most time-consuming part in this algorithm. Hence the time saving for ME depends on how many times the algorithm does this raster search, which in turn depends on the motion characteristics of the video.

4. SIMULATION AND RESULTS

Computation time for the motion-estimation function is estimated by reading CPU cycles from the CPU's hardware cycle counter according to the benchmarking instructions provided in [16]. The simulations are run in the CentOS 6.8 platform on an Intel® Xeon® CPU @ 2.67 GHz, having disabled hyper-threading and frequency scaling for getting accurate timing results. The first 30 frames of videos with 4K UHD are encoded with and without modification in the HM software for various test videos, and the obtained results are tabulated in Table 3. All video frames except the initial I-frame are encoded as P-frames using one reference frame, the quantization parameter (QP) set as 32, and with the three search ranges of 64, 128, and 256 pixels in both horizontal and vertical directions. From the results it is clear that PSNR degradation with

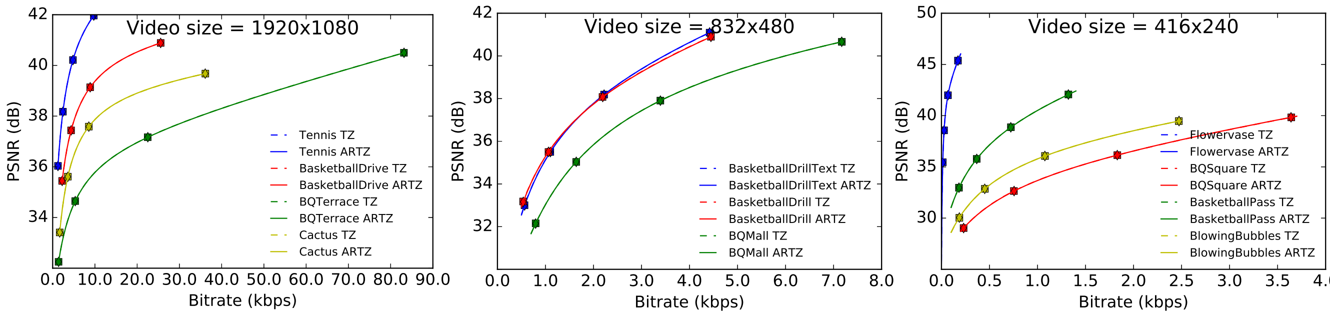

Fig. 4. RD curves for different video sequences with QPs 22, 27, 32, 37, with search length 64.

Table 3. ARTZ search algorithm results (negative values are improvements) with different search lengths for 4K UHD resolution

Video Sequence	Search range	Bitrate increase (%)	PSNR reduction (dB)	ME time saving (%)
TunnelFlag	64	0.291	0.013	39.9
TunnelFlag	128	0.2693	0.0035	44.2
TunnelFlag	256	0.459	0.0048	39.2
ToddlerFountain	64	−0.0016	−0.0001	25.5
ToddlerFountain	128	0.0057	−0.0008	24.0
ToddlerFountain	256	−0.0105	0.0005	20.7
DrivingPOV	64	−0.0485	−0.0009	42.9
DrivingPOV	128	−0.3278	−0.0011	50.3
DrivingPOV	256	−0.0675	0.003	49.5

the standard TZ search algorithm is negligible; for the worst case, it is 0.013 dB with a search range of 64 pixels in 4K UHD video. Bit-rate increases are also insignificant in the results, which have a worst-case value of 0.459%. It seems that in some cases the proposed ARTZ algorithm is better able to track global minima instead of being trapped by local minima due to its directional properties, thus improving PSNR and bit rate (negative values indicates improvements by ARTZ) in several cases. Although the algorithm is developed for 4K UHD videos, the experiments with standard test videos of different sizes reveal that the proposed algorithm performance closely follows that of an HM software implementation of a TZ search as seen in the RD curves [17] plotted in Fig. 4, and from Table 4.

5. CONCLUSION

The algorithm proposed for computation saving in motion estimation is based on the probability of obtaining minima in certain locations of an initial diamond grid-search. The proposal is mainly intended for UHD-resolution video encoding in the HEVC standard. The proposed ARTZ algorithm improves the computation time for motion estimation considerably, with a maximum of 50.3% time saving compared to the standard TZ search algorithm. The worst-case bit-rate degradation and PSNR degradation are 0.459% and 0.013 dB respectively, which are negligible, and for the best-case bit rate and PSNR are improved by 0.52% and 0.0239 dB respectively. Hence the modified algorithm can be used for encoding in HEVC and may also be suitable for other encoding standards such as AVC.

Table 4. ARTZ results with search length 64 and different QPs

Video sequence	QP	Bitrate increase (%)	PSNR reduction (dB)	ME time saving (%)
Tennis	37	0.0342	0.0022	33.6
Tennis	32	0.0167	0.0022	31.6
Tennis	27	−0.027	0.0019	30.1
Tennis	22	0.0596	0.0004	28.5
BasketballDrive	37	−0.0284	0.0007	36.9
BasketballDrive	32	0.0384	0.0002	35.6
BasketballDrive	27	−0.0132	0.0006	34.2
BasketballDrive	22	−0.0242	0.0007	31.9
Cactus	37	0.026	−0.0021	39.4
Cactus	32	0.0425	0.0044	39.1
Cactus	27	0.0391	−0.0004	38.5
Cactus	22	−0.0818	0.0004	38.2
BasketballDrillText	37	0.3566	0.0095	39.5
BasketballDrillText	32	0.1318	0.0056	38.0
BasketballDrillText	27	−0.0446	0.0027	37.8
BasketballDrillText	22	−0.0139	−0.0062	37.8
BasketballDrill	37	0.5209	0.0181	39.6
BasketballDrill	32	−0.0224	−0.0056	38.8
BasketballDrill	27	0.2136	0.0001	37.8
BasketballDrill	22	−0.0052	0.0055	37.6
BQMall	37	−0.137	0.0119	39.6
BQMall	32	0.0402	−0.0047	39.6
BQMall	27	0.0247	0.0023	39.4
BQMall	22	−0.0269	−0.0031	39.8
BQSquare	37	0.0294	−0.0012	41.6
BQSquare	32	0.2745	0.0023	41.4
BQSquare	27	−0.0336	−0.0024	40.7
BQSquare	22	−0.0792	−0.0053	41.2
BasketballPass	37	0.0	0.0017	40.0
BasketballPass	32	0.0936	0.0054	39.0
BasketballPass	27	−0.0962	0.0087	38.9
BasketballPass	22	−0.0107	0.0078	40.4
BlowingBubbles	37	−0.0759	−0.0239	37.6
BlowingBubbles	32	−0.2183	−0.0015	37.6
BlowingBubbles	27	−0.0933	0.0032	37.1
BlowingBubbles	22	0.0911	−0.0059	37.5

6. REFERENCES

[1] G. J. Sullivan, J. Ohm, W. Han, 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, Dec. 2012.

[2] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards – including high efficiency video coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[3] ITU-T and ISO/IEC JTC 1, *Advanced video coding for generic audiovisual services*, 2003.

[4] "Cisco visual networking index: Forecast and methodology, 2015–2020," [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html.

[5] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7–28, 2004.

[6] Z. Chen, J. Xu, Y. He, and J. Zheng, "Fast integer-pel and fractional-pel motion estimation for H.264/AVC," *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 264–290, 2006, Introduction: Special Issue on emerging H.264/AVC video coding standard.

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

[8] X. Li, R. Wang, W. Wang, Z. Wang, and S. Dong, "Fast motion estimation methods for HEVC," in *2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, June 2014, pp. 1–4.

[9] S. H. Yang, J. Z. Jiang, and H. J. Yang, "Fast motion estimation for HEVC with directional search," *Electronics Letters*, vol. 50, no. 9, pp. 673–675, April 2014.

[10] H. Kibeya, F. Belghith, H. Loukil, M. A. Ben Ayed, and N. Masmoudi, "TZ search pattern search improvement for HEVC motion estimation modules," in *Advanced Technologies for Signal and Image Processing (ATSIP), 2014 1st International Conference on*, March 2014, pp. 95–99.

[11] J. H. Jeong, N. Parmar, and M. H. Sunwoo, "Enhanced test zone search algorithm with rotating pentagon search," in *2015 International SoC Design Conference (ISOCC)*, Nov 2015, pp. 275–276.

[12] T. Nguyen, P. Nguyen, P. Nguyen, and C. Dinh, "A novel search pattern for motion estimation in high efficiency video coding," in *2016 International Conference on Computer Communication and Informatics (ICCCI)*, Jan 2016, pp. 1–6.

[13] N. Purnachand, L. N. Alves, and A. Navarro, "Improvements to TZ search motion estimation algorithm for multiview video coding," in *2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, April 2012, pp. 388–391.

[14] "HEVC reference software 16.3," [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/.

[15] N. C. Vayalil, A. Safari, and Y. Kong, "ASIC design in residue number system for calculating minimum sum of absolute differences," in *2015 Tenth International Conference on Computer Engineering Systems (ICCES)*, Dec. 2015, pp. 129–132.

[16] G. Paoloni, "How to benchmark code execution times on Intel® IA-32 and IA-64 instruction set architectures," [Online]. Available: http://www.intel.com/content/www/us/en/embedded/training/ia-32-ia-64-benchmark-code-execution-paper.html, Sept. 2010.

[17] G. Bjøntegaard, *Calculation of average PSNR differences between RD-curves*, Joint Collaborative Team on Video Coding (JCTVC), Apr. 2001.