Computational complexity tradeoffs in HEVC motion estimation

Jakub Stankowski, Tomasz Grajek, Damian Karwowski, Krzysztof Klimaszewski, Olgierd Stankiewicz, Krzysztof Wegner
Chair of Multimedia Telecommunications and Microelectronics
Poznan University of Technology
Poznan, Poland
jstankowski@multimedia.edu.pl

***Abstract*—The paper reports the results of the experiments conducted on the HEVC encoder, regarding the complexity and performance of the motion estimation algorithm. The research focuses on the search range of motion vectors and its influence on the encoding time of the video with the use of a HEVC encoder, as well as on the method for choosing the starting point of the motion estimation algorithm. Performance is measured as the time reduction and Bjontegaard metric bitrate reduction.**

Keywords—HEVC compression, motion estimation, motion vectors

I. Introduction

The most recent development in video coding, standardized as High Efficiency Video Coder (HEVC) by Moving Pictures Experts Group (MPEG) and as H.265 standard by International Standardization Organization (ISO), gives a significant advantage in compression efficiency compared to previously developed standards, like AVC. With about 50% reduction of bitrate compared to AVC with the same quality of reconstructed video [1], HEVC is sure to replace older standards in the coming years.

The high performance of the new codec, however, comes at a cost of greatly increased computational complexity of the encoding process. Entirely revised concept of macroblocks, replaced by a much larger tree structure called Coding Tree Unit (CTU) and introduction of complex prediction algorithms poses a great challenge for hardware and software that encodes the video using the new standard.

In order to make it easier to use the HEVC, many different ways of simplifying the process of encoding are being proposed and evaluated.

One of the most time consuming tasks during the encoding process is motion estimation. The HEVC performs motion estimation using a sophisticated iterative method called TZsearch [2]. The time that is required for this method to find the motion vector depends on the contents of the currently estimated block and on the accuracy of the motion vector prediction. The time required for the motion estimation depends also on the search area.

The works reported in this paper investigate the complexity of the motion estimation process and the potential speedup of the encoding based on the modifications in motion vector estimation process. The issue of motion estimation has already been the subject of the authors' research that were described in the work [4]. This work is a continuation of those studies, focused on different aspects of the process, and deeply explores to what extent the motion vector search range and the motion estimation starting point affect the performance of HEVC motion estimation process.

II. Motion estimation in HEVC and experiments methodology

The motion estimation in HEVC works on the tree like structure. In this structure, every CTU can be further divided into smaller coding units (CU). Each CU is, in turn, encoded using prediction performed in prediction units (PU). Each CU can be coded using a single square PU, two square or rectangular PUs or 4 square PUs of half the size of CU. For each PU, the motion vector or plurality of motion vectors need to be estimated.

In order to study the issues connected with time complexity of the motion estimation process as used in HEVC, a set of tests was performed. Two phases of the experiment can be clearly distinguished in the research. In the first phase, the influence of the search range and position of the starting point on performance (complexity and coding efficiency) of HEVC motion estimation were deeply studied. Experience taken from this stage of the research was used in the second phase to elaborate modifications of HEVC motion estimation process. The efficiency of the modified encoder was thoroughly investigated with the use of the framework of the HM 16.9 reference software.

All the tests were performed using the widely accepted dataset, also used by MPEG in its works on video compression standardization and development [3]. The set includes 24 sequences with different properties and different types of motion. The tests were conducted in accordance to the "common test conditions", as defined by MPEG for the works on the video compression technologies [3].

Research project was supported by The National Centre for Research and Development, POLAND, Grant no. LIDER/023/541/L-4/12/NCBR/2013.

For the purposes of the first stage of the research, a standalone software was prepared from scratch, which enabled an accurate estimation of the time that is spent in the encoder on the process of motion estimation. In this place, special attention was paid to take into account all nuances of the motion estimation used in the HEVC video encoder. Prepared software gave the possibility of testing the complexity of the algorithm with many different settings outlined below.

As already highlighted above, some parts of the HEVC motion estimation algorithm have also been changed in the HM 16.9 software in order to test the possibility of improving the algorithm used in the HM software.

All of the results were obtained on computers of the following configuration: Intel core i7 – 5820K (3.6 GHz – 1 core, 3,4 GHZ – 2 cores, 3.3 GHz – more than 2 cores) and 64 GB of RAM (68 GB/s memory bandwidth).

III. Performance Analysis of Motion Estimation – Results and Discussion

A. *Encoding time versus search range*

The first experiment was focused on assessing the time required for motion estimation process when the search range is different. The results of this experiment are shown on Fig. 1. The reference time is the time used by the motion estimation with search range (SR) set to ±64 pixels. The starting points (SP) for the motion estimation considered during the search are the ones coming from the prediction mechanism used in HEVC as well as zero prediction vector (in this case the starting point of the search is situated in the position of the block itself, i.e. motion of this block is predicted to be zero). The results are averaged over all of the test sequences.

The first observation is that reduction of the search range does influence the estimation time, but the time is not proportional to the area of the search. The time used as a reference of motion estimation process, for the area of ±64 x ±64 pixels is only five times smaller than the time for search range of ±256 x ±256 pixels, which range has 16 times greater area. Similarly, for smaller search ranges, the decrease of the time for search range ±4 x ±4 pixels is only by a factor of little less than 3, while the search area is 256 times smaller. The justification of the results lies in the nature of the TZsearch algorithm. It operates partially as a logarithmic search, therefore limiting the influence of the search area.

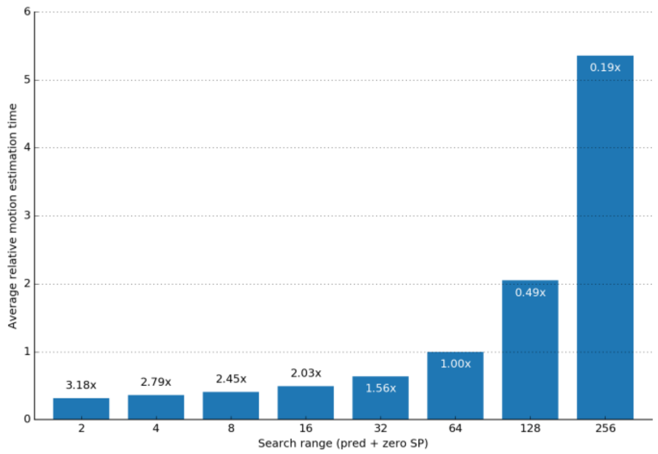

Fig. 1. Dependency of the time consumed by motion estimation algorithm on the search range.

According to those results, while not drastic, the noticeable reduction of the encoding time can be achieved by reducing the search range for motion vectors. Presented results show precisely what is the factor of the time reduction.

However, the reduction of the search range inevitably leads to the loss of the quality of encoded data, especially so for the sequences with complex and fast motion. This kind of motion does not lend itself to an accurate prediction, and reduction of the search range for the motion vectors causes the encoder to select the motion vector that can be far from being optimal.

To verify the quality impact of the reduced search range, another set of data was also gathered during the experiment. This time, the quality of the prediction signal is calculated for different block sizes (only one size of the block is allowed for each of the block size set of results). This quality is measured as a PSNR metric calculated with the original frame undergoing prediction used as reference. The results can be seen on Fig. 2. Additionally to the search range, also the block size is recorded. As previously, the results show an average over all of the sequences used in the test.

As can be seen, the highest PSNR of the prediction signal is achieved when using small blocks of the size 4x4 pixels (size not allowed for inter coded units in HEVC). In this case, the difference of quality between the search range equal to ±256 pixels and ± 2 pixels is 2dB. This difference decreases with the growing size of the search range, and is equal to 1.4dB for the largest possible blocks (64x64 pixels). For the larger block the quality of prediction signal significantly decreases, this is, however, not a surprising fact, since smaller blocks can better adjust to the motion field of the sequence. It must be noted however, that in the case of the use of smaller bocks the total bit cost of sending motion vectors increases (since there is more of them), which quantitatively has not been presented in this work. However, it may be an interesting subject of future research.

B. *Encoding time versus starting point*

As already stated, the performance of the search algorithm also depends on the point where the search starts. Choosing the starting point close to the final motion vector coordinates greatly reduces the time spent on the process of finding the motion vector. In the next experiment, the influence of the

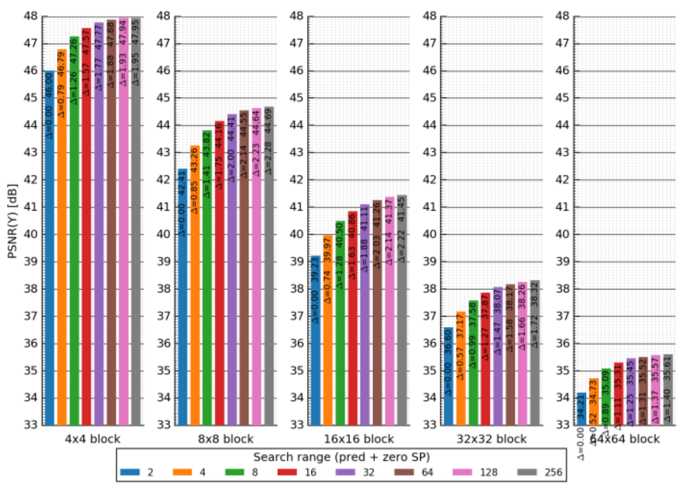

Fig. 2. Dependency of the prediction signal PSNR on the block size and the search range.

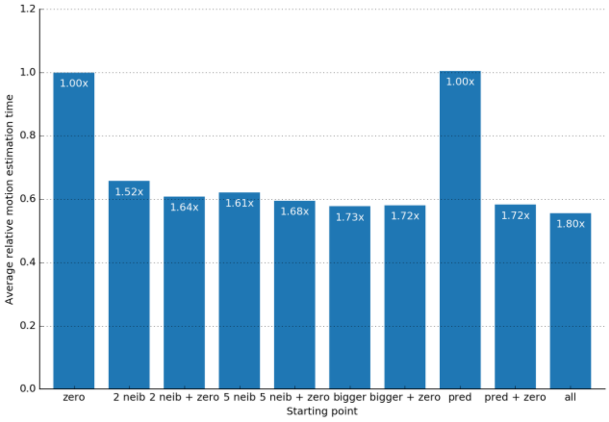

Fig. 3. Dependency of the time consumed by motion estimation algorithm on the starting point.

starting point on the time of the process of motion estimation was evaluated. The results are shown on Fig. 3. The following starting points were investigated:

- *zero*: the motion estimation algorithm starts from the zero position (i.e. the corner of the collocated block),
- *2 neib*: two spatially neighboring blocks motion vectors are checked, and the one with smaller difference to the currently encoded block is used as starting point,
- *2 neib + zero*: a combination of two above cases,
- *5 neib*: five spatially neighboring blocks motion vectors are checked, and the one with the smallest difference to the currently encoded block is used as starting point,
- *bigger*: the motion vector of the parent block (i.e. the bigger one that is one level higher in the CTU division tree) is used as the starting point,
- *pred*: the method used natively in HEVC, selected from the set of candidates as described by HEVC,
- *all*: all of the starting points mentioned above are verified, and the one with the smallest difference to the currently encoded block is used as a starting point.

It can be seen that the process of the motion estimation greatly depends on the starting point, since the closer the starting point is to the ultimately chosen motion vector, the faster the TZsearch algorithm is able to terminate. A reduction of about 40% of the motion estimation time can be expected when all of the mentioned starting points are considered, comparing to the case where only zero starting point is used or the points taken from the algorithm used in HEVC. This shows that the choice of the potential starting points in HEVC is not effective and can be improved by selecting different prediction candidates.

The choice of the starting point for the motion estimation also does influence the prediction signal PSNR, measured with the original encoding frame taken as reference. The corresponding results are presented on the Fig. 4. It can be seen that the significant influence on the PSNR of the prediction signal is visible only for block sizes 4x4 pixels, 8x8 pixels and 16x16 pixels. For large blocks of 64x64 pixels there is practically no difference between different starting points of motion estimation algorithm. For large blocks the quality is already so low, that it seems it cannot be degraded any further by a bad choice of the starting point for motion estimation. This comes from the fact that large blocks do not reflect the motion of objects in the scene with enough precision. What is very prominent is the fact that the selection of the starting point using the method from HEVC performs bad for the small block sizes.

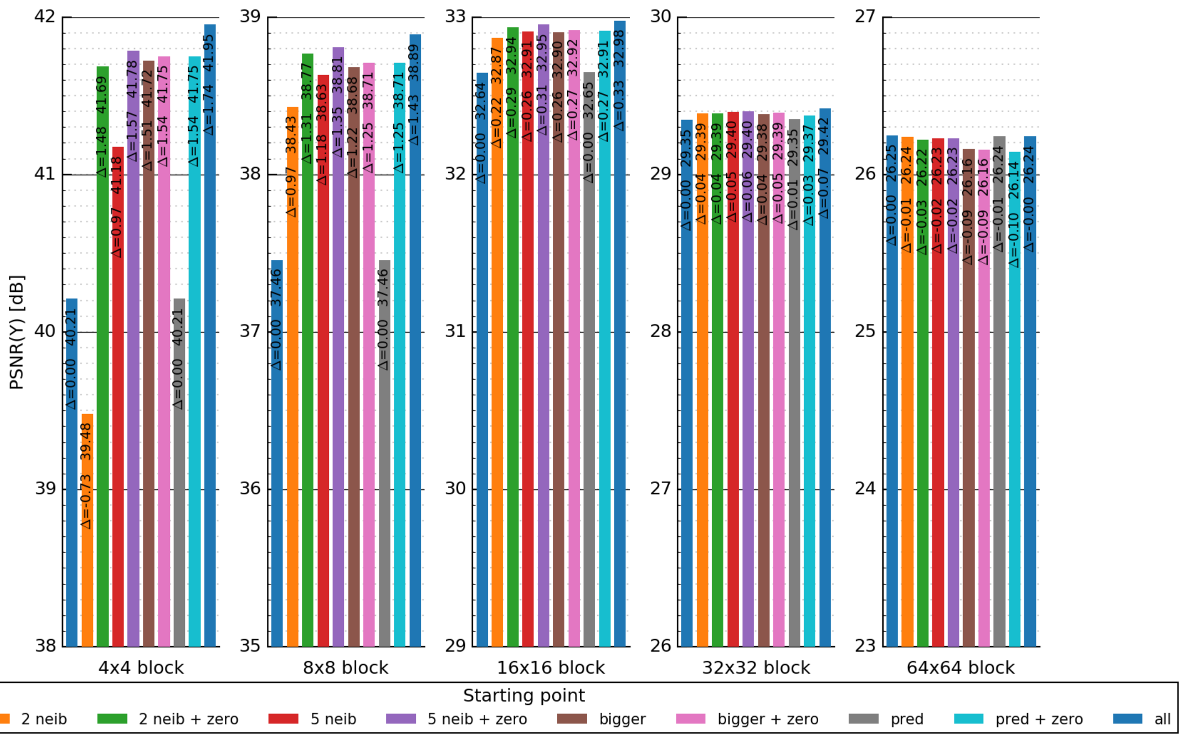

Fig. 4. Dependency of the time consumed by motion estimation algorithm on the starting point.

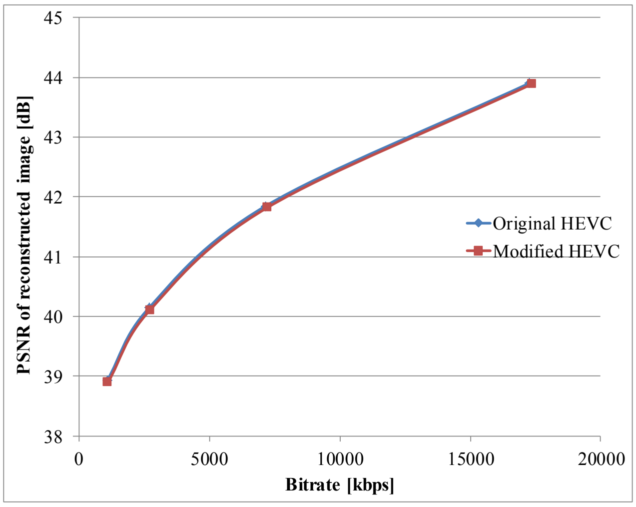

Fig. 5. The quality versus bitrate for scenario A1.

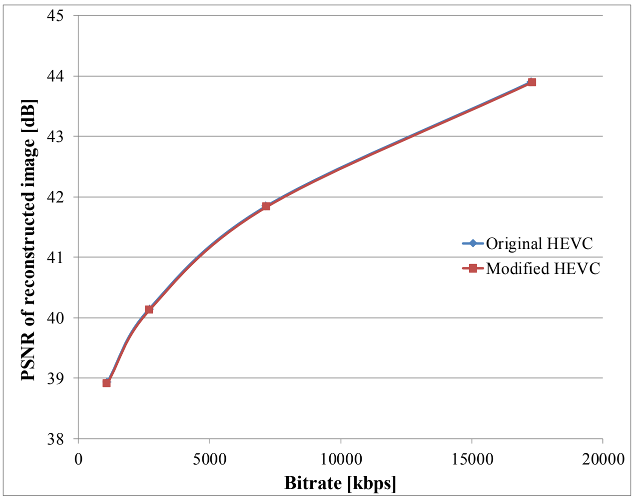

Fig. 6. The quality versus bitrate for scenario A2.

IV. Modifications of Motion Estimation in the HEVC – Results and Disscussion

In order to exploit the described dependency of the motion estimation time on the search range, the following method in two variants has been developed and implemented in HEVC reference codec version HM 16.9. It was decided that the smaller the PU block, the smaller the search range should be, since the search for their motion vectors consumes the most time for the unmodified HEVC encoder In the test, the search range was therefore set to correspond with the size of the PU.

A. *Algorithm description*

For each prediction unit (PU), for which the motion estimation is performed during the encoding process, the search range is selected in the following way.

1. In variant A1, the search range was set to be equal to the maximum from the dimensions of the PU. This means that the PU of the size of 16x16 pixels had search range set to ±16 points, and the PU of the size of the 32x16 pixels had the search range set to ±32 points.
2. In variant A2, the search range was set to be equal to the minimum from the dimensions of the PU. This means that the PU of the size of 16x16 pixels had search range set to ±16 points, and the PU of the size of the 32x16 pixels had the search range set to ±16 points.

B. *Results*

The results obtained using those two strategies are presented on Fig. 5 and 6, averaged over all of the tested sequences. Time reduction factor is presented in Tables 1 and 2. For the Fig. 5 and 6, four encoding runs were performed for the QP parameter equal to 22, 27, 32, 37.

The aggregated results are presented on Fig. 7. Those results are expressed in terms of the bitrate increase in the Bjontegaard metric, as described in [5], versus the encoding time change. For comparison purposes, also the results that were obtained using a constant search range smaller than the default ±64 x ±64 are presented. In this case, every block, regardless of its size, had the motion vector estimated within the same constant search range. There are a few interesting observations concerning those results.

First of all, the time for search range reduced to ±32 x ±32 leads to the increase of encoding time. This may be caused by less accurate motion estimation that causes the algorithms to predict and choose worse starting points and to extend the time spent on motion estimation. This increase of time is not compensated by the decrease of the time spent during motion compensation on comparisons for each block and each tested vector value. Another explanation would be that more time is spent on other stages when the estimated motion vector is worse than optimal one obtained for reference case (search range ±64 x ±64).

The same observation holds true for the case with search range set to ±4 x ±4 pixels. For this case a massive 12% bitrate increase (in terms of Bjontegaard metric) can be observed with only a slight decrease of compression time (by about 0,7%).

Another observation is that both methods, A1 and A2, outperform other tested scenarios either in terms of encoding time reduction or bitrate increase.

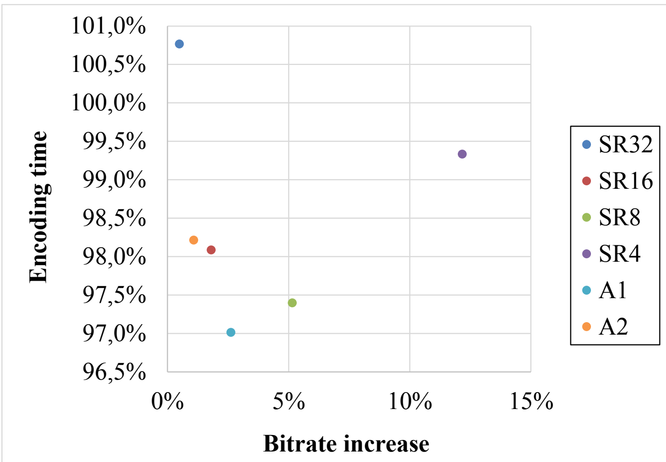

Fig. 7. Encoding time (relative to default HEVC value of ±64 x ±64) and bitrate increse (relative to ±64 x ±64) for different search range establishing algorithm.

TABLE I. TIME REDUCTION FOR SCENARIO A1

QP	Compression time relative to original HEVC
22	94.8%
27	97.2%
32	97.4%
37	96.9%
Average:	96.6%

TABLE II. TIME REDUCTION FOR SCENARIO A2

QP	Compression time relative to original HEVC
22	94,0%
27	96,2%
32	100,2%
37	99,3%
Average:	97.4%

V. CONCLUSIONS

The results presented in the paper show, quite surprisingly, that the reduction of time used for motion estimation does not significantly reduce the overall compression time. This may be caused by the fact that choosing suboptimal motion vectors has a side effect of decreasing the performance of the further stages of video encoding. The stages like choosing the transform block size and estimating the number of bits may perform worse, since they must process more data (since the prediction signal is worse, and, therefore, the error signal undergoing the coding process, is larger and more complex).

Another important conclusion is that the starting point selection for motion estimation in HEVC might be further improved in order to provide faster compression without a significant loss in compression efficiency and reconstructed image quality.

REFERENCES

[1] G. Correa, P. Assuncao, L. Agostini, L. Cruz, Performance and Computational Complexity Assessment of High-Efficiency Video Encoders, IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, December 2012.

[2] Xiu-li T., Univ H., Sheng-kui D., An analysis of TZSearch algorithm in JMVC, International Conference on Green Circuits and Systems (ICGCS), Shanghai, 21-23 June 2010, s. 516 – 520.

[3] Bossen F.. 2012. „Common test conditions and software reference configurations". Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc. JCTVC-J1100.

[4] J. Stankowski, D. Karwowski, K. Klimaszewski, K. Wegner, O. Stankiewicz, T. Grajek, Analysis of the Complexity of the HEVC Motion Estimation, 23rd International Conference on Systems, Signals and Image Processing (IWSSIP), Bratislava, 23-25 May, 2016.

[5] Bjøntegaard G.. 2001. „Calculation of average PSNR differences between RD curves". ITU-T SG16/Q6, Doc. VCEG-M33.