

Cebrián-Márquez, G., Chi, C. C., Martínez, J. L., Cuenca, P., Álvarez-Mesa, M., Sanz-Rodríguez, S., & Juurlink, B.

Reducing HEVC encoding complexity using two-stage motion estimation

Conference paper | Accepted manuscript (Postprint)

This version is available at <https://doi.org/10.14279/depositonce-6792>



Cebrián-Márquez, G., Chi, C. C., Martínez, J. L., Cuenca, P., Álvarez-Mesa, M., Sanz-Rodríguez, S., & Juurlink, B. (2015). Reducing HEVC encoding complexity using two-stage motion estimation. In 2015 Visual Communications and Image Processing (VCIP). IEEE. <https://doi.org/10.1109/vcip.2015.7457851>

Terms of Use

© © 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

REDUCING HEVC ENCODING COMPLEXITY USING DECOUPLED MOTION ESTIMATION

G. Cebrián-Márquez¹, C. C. Chi², J.L. Martínez¹, P. Cuenca¹,
M. Álvarez-Mesa², S. Sanz-Rodríguez², B. Juurlink²

¹High-Performance Networks and Architectures (RAAP). Albacete Research Institute of Informatics (I3A). University of Castilla-La Mancha, Albacete, Spain.

²Embedded Systems Architecture (AES). Technische Universität Berlin, Berlin, Germany.

ABSTRACT

We propose a technique for optimizing the High Efficiency Video Coding (HEVC) encoder by reducing the number of operations performed in the motion estimation stage. The technique is based on the fact that a significant number of motion estimation operations are performed repetitively for the same image samples, but for different block partition sizes. By decoupling the initial motion estimation and the block partitioning into different stages it is possible to remove a considerable number of redundant motion estimation operations, that in turn translates into a complete application time reduction. An implementation of the proposed technique on a SIMD optimized version of the HEVC reference encoder shows that, on average, a reduction of 79.02% SAD operations can be achieved, that results in an average speed-up of $1.17\times$, with negligible impact on the compression efficiency (BD-rate losses of less than 1%).

Index Terms— HEVC, H.265, Motion Estimation, Fast Video Encoding, Video Compression

1. INTRODUCTION

The continuous market demands for a better quality of experience in video applications have motivated the development of the High Efficiency Video Coding (HEVC) [1] standard. Compared to its predecessor H.264/Advanced Video Coding (AVC), it is able to reduce the bitrate up to 50% while keeping the same quality level [2]. These rate-distortion (R-D) improvements have come, however, at the expense of an increase in the encoding computational complexity, which can be impractical in some application scenarios.

In order to reduce the encoding time, HEVC provides some high-level tools, such as tiles and wavefronts, aiming at allowing the parallel execution of the encoder on multicore computer architectures without sacrificing the compression efficiency significantly [3]. These techniques reduce the execution time by dividing the total work into multiple threads that operate in parallel. An orthogonal way of speeding up the encoder consists of reducing the total amount of work performed *per thread*, but in such a way that the effect on the compression efficiency is low. Among different encoding stages, motion estimation (ME) takes a significant part of the execution time and therefore is a good candidate for this type of optimization.

Figure 1 shows the profiling results obtained from a single instruction multiple data (SIMD)-optimized baseline HEVC encoder

This work has been jointly supported by the Spanish Ministry of Economy and Competitiveness and the European Commission (FEDER funds) under the project TIN2012-38341-C04-04, and by the Spanish Ministry of Education, Culture and Sports under the grant FPU 13/04601.

for the *Kimono* 1080p sequence, using the Random Access configuration and setting the quantization parameter (QP) to 32. As can be seen, most of the encoding time is devoted to the ME operation, whereas only 15.17% of the total execution time corresponds to the remaining encoder modules (labelled as “others”), which include intra prediction, transform, quantization, entropy coding, and in-loop filtering. A significant part of the ME time is consumed by operations that are performed repetitively on the same image samples but in different block partitions; e.g. different Prediction Block (PB) sizes for the same Coding Tree Block (CTB). Due to the considerable increase in block partitioning options that the HEVC standard provides compared to H.264/AVC, there is potentially a large number of those redundant operations. As they are performed many times unnecessarily it would be possible to remove them, and then reduce the total execution time without sacrificing the compression efficiency.

In this paper we propose a technique for removing these redundant operations by decomposing the ME stage into two sub-stages called pre-analysis and encoding. In the pre-analysis stage, ME is performed for a small and single partition block size in order to calculate initial motion estimation information. In the encoding stage, motion vector (MV) refinement and block partitioning are conducted based on the initial motion information, to end up with the final encoding.

A similar technique to the one proposed in this paper is employed in the lookahead module of x264 H.264/AVC encoder [4]. However, neither a deep analysis of its parameters nor its benefits is available. Moreover, this technique is targeted for H.264/AVC, which involves less block partitions than in HEVC. Some other works proposed algorithmic solutions for inter mode complexity reduction in HEVC. These approaches aim to perform a reduced search to decide the block partitioning by using techniques such as early termination of Coding Unit (CU) splitting [5], reduction of tested partitions based on motion activity [6], reduction of inter modes and CU sizes based on statistically derived thresholds [7], and fast inter CU selection based on correlations between CU splitting and motion divergence [8]. In most of the cases average time savings of around 50% can be obtained with low R-D losses [9]. As most of these approaches still evaluate multiple partitioning schemes, they can be combined with the technique described in this paper.

The rest of this paper is organized as follows. In Section 2 the proposed decoupled ME is described in detail. In Section 3 the experimental setup is described, followed by the analysis of the experimental results in Section 4. Finally, in Section 5 conclusions are drawn.

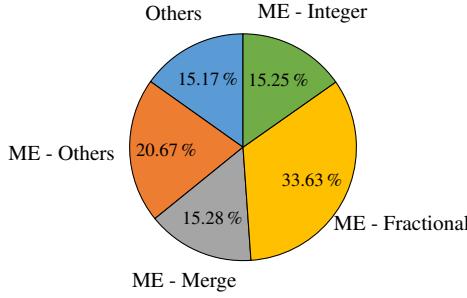


Fig. 1: Time profiling of the SIMD-optimized HEVC encoder. Sequence: Kimono_1920x1080_24fps, Random Access, QP32

2. DECOUPLED MOTION ESTIMATION ALGORITHM

The goal of the proposed approach is to reduce the number of ME search operations by decomposing the encoder into two stages: pre-analysis and encoding. A block diagram of the proposed scheme is presented in Figure 2. As can be seen, both stages perform a motion estimation of the current frame. First, the pre-analysis performs a faster and more coarse-grained estimation over a large search area, and later, the encoding stage carries out a refinement of the MVs estimated in the pre-analysis for every Prediction Unit (PU). The reduction of ME operations comes from the reduced search range in the encoding stage, and the sharing of initial ME information resulting from the pre-analysis phase among several partition block sizes. These two stages are described in detail in the next sections.

2.1. Pre-Analysis Stage

In this phase, the motion estimation is performed on the input picture using a fixed block size and taking the original pictures as reference. The motion estimation is performed with integer sample precision and fixed search area for all the reference frames. As a result, the output of this phase is an intermediate buffer containing the best MV associated to each block for each reference picture. The following features are included in this stage with the aim of obtaining a reliable MV map:

- The neighbour MVs of the current block are considered as predictors similarly to the Advanced Motion Vector Prediction (AMVP) feature of the HEVC standard [1].
- The ME algorithm is the same as the one used in the encoding stage.
- The well-known Lagrangian multiplier R-D optimization model is used:

$$J_{MV} = SAD(MV) + \lambda_{MOTION} R(MV), \quad (1)$$

where J_{MV} stands for the cost function, $SAD(MV)$ the distortion function in terms of the sum of absolute differences (SAD), λ_{MOTION} is the Lagrangian multiplier for motion estimation, and $R(MV)$ represents the bits required to code the MV.

2.2. Encoding Stage

In this stage, the final encoding process is conducted using an integer precision MV map from the motion information of the previous

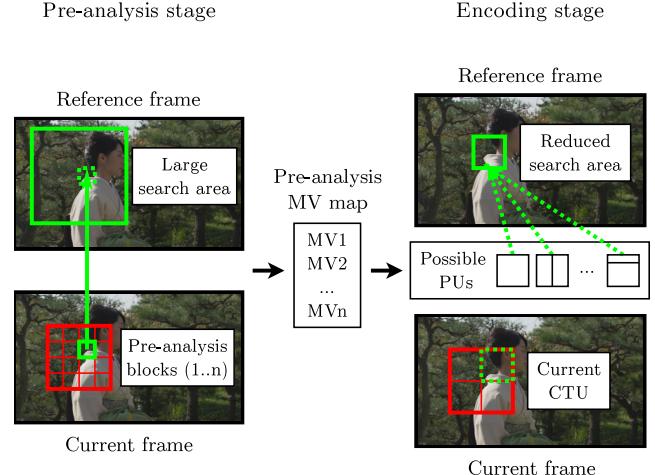


Fig. 2: Proposed decoupling of ME into pre-analysis and encoding phases.

stage. This allows the ME process in the block encoding to be accelerated mainly in the following ways:

- Instead of testing the two AMVP candidates in a PU to select the best ME starting point, we directly use the MV from the pre-analysis phase that was estimated for that position. If the PU is larger than a pre-analysis block, the median of the corresponding estimated MVs is calculated and set as starting point. The nearest AMVP candidate to the estimated MV is selected as the predictor.
- Because the pre-analysis MV is in most cases similar to the one that the baseline encoder would find, the search range for this stage can be reduced significantly with only a very small impact on the coding efficiency.

3. EXPERIMENTAL SETUP AND ENCODING CONDITIONS

The experiments have been executed using an optimized version of the reference software: HEVC test Model (HM-14.2) [10]. In the optimized HM we have disabled several tools and simplified the RD-model to improve the feasibility of the experiments and to better approach the impact of the proposed technique on real-world HEVC encoders. The tools that are not considered are the Residual Quad Tree (RQT) splitting, transform skip, and differing chroma intra modes. Furthermore, the merge candidates are not considered in a full-RD scheme where the residual is transformed and entropy coding is performed to estimate the rate (R). Similarly the number of candidates considered in the full-RD scheme in intra prediction is reduced to 3.

In the ME search algorithm, simplification has also been introduced by removing the raster search refinement phrase from the original 3-step HM algorithm [11]. Finally, many operations in the encoder have been SIMD optimized using Intel AVX2 instructions, providing noticeable speed-ups with no coding efficiency penalties, specially for the motion estimation and compensation modules. On average, these changes provide a $4.02 \times$ speed-up at the cost of only a 3.5% increment in BD-rate compared to the reference HM.

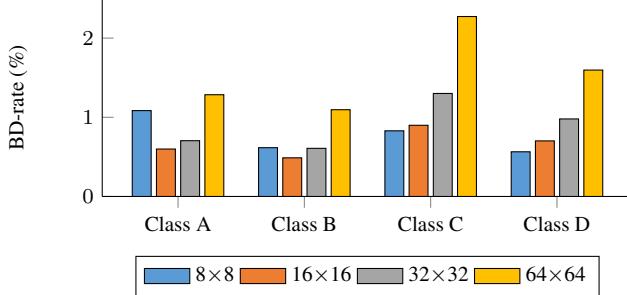


Fig. 3: BD-rate results for different pre-analysis block sizes

The Random Access configuration has been selected for the experiments and, except for the optimization detailed before, the JCT-VC common test conditions [12] have been used for the experiments. In this way, among other parameters, the QP values that have been used are 22, 27, 32 and 37. The encodings have been carried out using 4:2:0 chroma format and 8-bit bit depth. The JCT-VC class A to D test sequences have been used for the experiments, which include resolutions 2560×1600 (A), 1920×1080 (B), 832×480 (C), and 416×240 (D).

The hardware platform and software stack used in these experiments is composed of an Intel Core i7-4770T processor running at 2.50 GHz, and 12 GB of system memory. The encoder has been compiled with GCC 4.8.2-19 and executed on Ubuntu 14.04 (Linux 3.13.0-39). Turbo boost and Hyperthreading have been disabled to improve the reproducibility of the results.

4. ALGORITHM TUNING AND RESULTS

To find a reasonable configuration for the decoupled motion estimation, experiments have been performed for different fixed block sizes of the pre-analysis phase, and different search ranges of the encoding phase. The pre-analysis search range has been set to 64 for all the experiments, because we found that reducing this parameter affected the compression performance noticeably. For brevity, only a subset of the complete results of different pre-analysis search ranges are presented.

The results are shown in terms of speed-up in the case of timing values, while the coding efficiency is calculated with the Bjøntegaard Delta Rate (BD-rate) metric [13]. The BD-rate indicates the percentage of bitrate reduction between two sequences with the same objective quality.

4.1. Pre-analysis Block Size Results

The size of the pre-analysis blocks determines the granularity of the motion estimation. A smaller block size results in a higher number of estimated MVs per frame, thus increases the complexity of the pre-analysis phase. A too large block size, however, results in sub-optimal predictors for the ME of the smaller PUs. To determine the best block size, experiments have been performed with block sizes from 8×8 to 64×64. Figure 3 shows the BD-rate results of different pre-analysis block sizes. The baseline is the decoupled ME encoder with the search range of the encoding phase set to 4. This configuration represents the practical scenario of the decoupled ME encoder. Results with other encoding search ranges have been performed and show a similar trend.

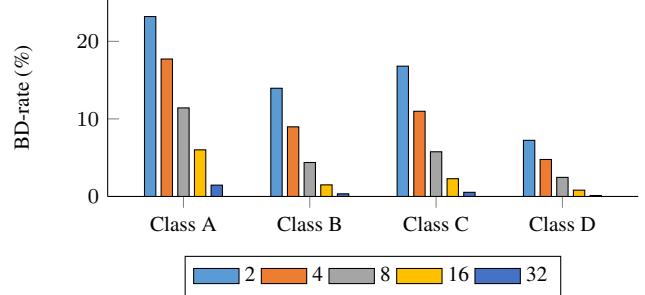


Fig. 4: BD-rate results for different search ranges (baseline encoder)

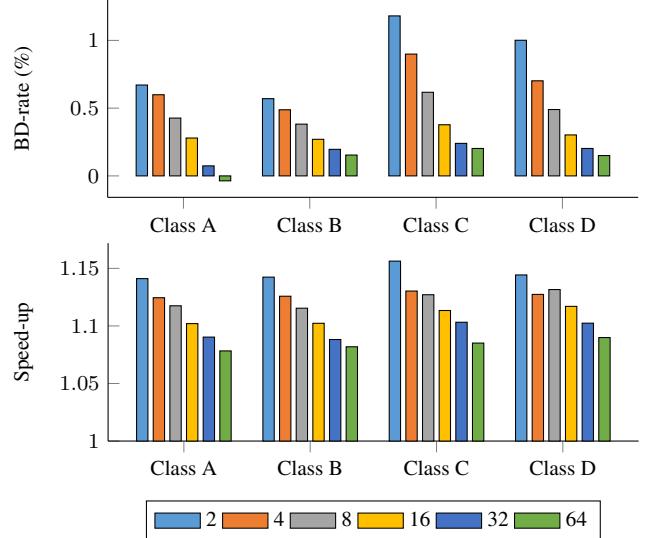


Fig. 5: BD-rate and speed-up results for different encoder search ranges (proposed encoder)

As the figure shows, block sizes 8×8 and 16×16 have smaller BD-rate losses compared to the 32×32 and 64×64 block sizes. When analysing the results in more detail, a correlation is found between the QP, the resolution, and the optimal pre-analysis block size. In general, the BD-rate results improve when the pre-analysis block size better matches the final encoding block size. For the remainder of the experiments a 16×16 block size is used because it provides overall the best compression, especially for higher resolution videos. As we will see in more detail in Section 4.3, the time spent in the pre-analysis phase is small and, therefore, the contribution to the total encoding time is negligible independent of the block size.

4.2. Encoder Search Range Results

Because the pre-analysis provides an estimated version of the MV, it is possible to assume that it is a good starting point for the ME search. Therefore, it is not necessary to use a large search range. To find out how much the encoding search range can be reduced without compromising the coding efficiency too much, experiments have been performed with encoding search ranges from 2 to 64.

Figure 4 shows the BD-rate results obtained from the reduction of the search range in the baseline encoder. As can be seen, the

Table 1: Speed-up and coding efficiency of the decoupled motion estimation algorithm

Class	Video sequence	BD-rate (%)	Encoding speed-up	Uni-IME speed-up	Uni-IME SAD savings (%)
A	Traffic	0.87	1.15	2.43	65.84
	PeopleOnStreet	1.82	1.18	4.19	83.50
	NebutaFestival	-0.03	1.17	4.09	80.28
	SteamLocomotive	0.26	1.18	3.79	81.81
B	Kimono	0.45	1.17	3.63	80.93
	ParkScene	0.60	1.16	2.82	72.08
	Cactus	0.73	1.16	3.18	75.99
	BasketballDrive	0.71	1.19	4.15	83.76
C	BQTerrace	0.55	1.15	2.61	69.34
	BasketballDrill	0.88	1.16	3.64	79.72
	BQMall	0.77	1.18	3.37	76.31
	PartyScene	0.90	1.16	3.10	73.15
D	RaceHorses	2.22	1.22	5.21	86.69
	BasketballPass	1.01	1.21	4.44	83.35
	BQSquare	0.40	1.14	2.37	64.58
	BlowingBubbles	0.86	1.15	2.73	68.32
	RaceHorses	1.88	1.21	4.75	84.95
	Mean values	0.87	1.17	3.55	79.02

BD-rate losses caused by the reduction of this parameter are impractical, reaching values of more than 20%. Moreover, these coding inefficiencies incur additional work for other modules, so that the provided speed-up is limited to $1.06\times$ on average. Figure 5 shows, in turn, both the BD-rate and speed-up results for different search ranges when using the proposed decoupled motion estimation. It can be seen that the BD-rate losses are generally lower than 1%, which constitutes a major improvement in comparison. The BD-rate increases with the encoder search range, while the obtained speed-up reaches values between $1.08\times$ and $1.15\times$. It can also be seen that there are some losses in the case of search range 64 as well as some speed-up. This speed-up is a consequence of the complexity reduction of the AMVP part, as the MV calculated in the pre-analysis is used as starting point for the ME. The slight losses in BD-rate, in turn, are caused by small drifts produced in the calculation of the MVs in the pre-analysis phase.

As a conclusion, the results show that the encoding search range can be significantly reduced without a notable effect in coding efficiency. Moreover, the performance keeps improving with a higher reduction of the search range. A search range of 4 will be chosen for the experiments instead of 2 due to the existence of some outliers in classes C and D.

4.3. Overall Proposal Results

Up to this point, the proposed decoupled ME has been tuned to provide a good trade-off between speed-up and coding efficiency. According to the results, we propose to use of a pre-analysis block size of 16×16 , a pre-analysis search range of 64, and an encoding search range of 4 as the optimal configuration for the proposed algorithm. Furthermore, the reduced encoding search range allows the ME search pattern to be adapted to perform a local search in the reference area. With this regard, a hexagon plus refinement pattern has been integrated in the encoder ME for the remainder of the experiments. Table 1 shows the detailed results for each of the tested video sequences in terms of BD-rate and total encoding speed-up. In addition, the speed-up of the uni-predictive integer ME and the reduction of SAD operations is provided to better show the effect of

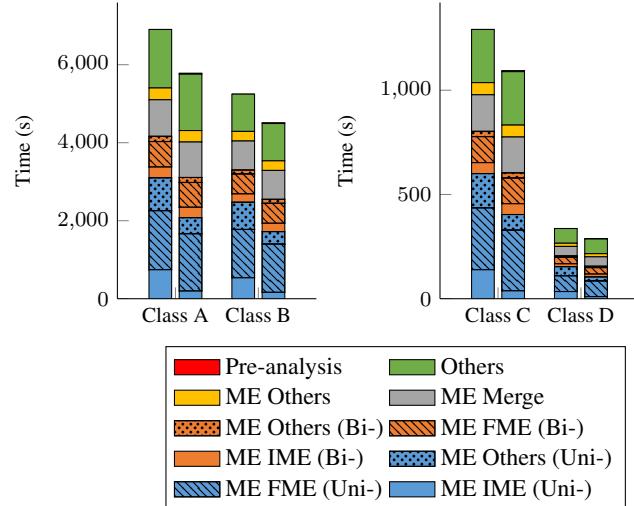


Fig. 6: Profiling of the baseline (left side bar) compared to the proposal (right side bar)

the reduced encoding search range.

The results show that the proposed algorithm is able to speed up the encoder by $1.17\times$ (14.6% of time reduction) at the cost of less than 0.9% coding efficiency penalties on average. These time savings, in turn, are produced by the fact that the uni-predictive integer motion estimation (IME) is being accelerated by $3.55\times$ average (71.8% of time savings), derived from the reduction of the number of SAD operations performed in the search area, which is significantly smaller with the use of the decoupled ME.

As a consequence, this algorithm involves a redistribution of the time in the encoder as shown in Figure 6. These results have been extracted from the average profiles of all sequences for classes A to D, and for all the tested QP values. As can be seen, the time spent in the uni-predictive IME is reduced due to the reasons previously mentioned, as well as the “ME Others” time, which includes comparing the AMVP predictors for determining the starting point of the ME in the baseline configuration. The rest of the encoder is unaffected by the proposed algorithm. The pre-analysis itself introduces such a small overhead that is not visible in the provided chart, as it is lower than 0.5% of the total time.

5. CONCLUSIONS

This paper has proposed a decoupled motion estimation (ME) algorithm to reduce the complexity of ME in HEVC. In the HEVC test model, ME is performed for each possible block partition, which results in this operation to be performed multiple times for the same samples. The decoupled ME algorithm reduces this redundant work by splitting off the integer ME into a pre-analysis and an encoding phase. Using the motion vectors found in the pre-analysis phase as predictors, the search range of the encoding phase can be reduced notably with negligible effects in the compression efficiency.

The experimental tests have shown that pre-analysis block sizes of 8×8 or 16×16 provide the lowest BD-rate losses. It is also shown that the encoding search range can be reduced from 64 to 4, without affecting the compression performance. Using this configuration, the decoupled ME algorithm is able to speed up the encoder by $1.17\times$ with less than 0.9% coding BD-rate losses.

References

- [1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, 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, pp. 1669–1684, Dec 2012.
- [3] C. C. Chi, M. Alvarez-Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, and T. Schierl, "Parallel Scalability and Efficiency of HEVC Parallelization Approaches," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 1827–1838, Dec 2012.
- [4] J. Garrett-Glaser, "A Novel Macroblock-Tree Algorithm for High-Performance Optimization of Dependent Video Coding in H.264/AVC," tech. rep., Department of Computer Science, Harvey Mudd College, 2009.
- [5] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, "An Effective CU Size Decision Method for HEVC Encoders," *IEEE Transactions on Multimedia*, vol. 15, pp. 465–470, Feb 2013.
- [6] L. Shen, Z. Zhang, and Z. Liu, "Adaptive Inter-Mode Decision for HEVC Jointly Utilizing Inter-Level and Spatiotemporal Correlations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, pp. 1709–1722, Oct 2014.
- [7] J. Lee, S. Kim, K. Lim, and S. Lee, "A Fast CU Size Decision Algorithm for HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, pp. 1–1, Jul 2014.
- [8] J. Xiong, H. Li, Q. Wu, and F. Meng, "A Fast HEVC Inter CU Selection Method Based on Pyramid Motion Divergence," *IEEE Transactions on Multimedia*, vol. 16, pp. 559–564, Feb 2014.
- [9] C. E. Rhee, K. Lee, T. S. Kim, and H.-J. Lee, "A survey of fast mode decision algorithms for inter-prediction and their applications to high efficiency video coding," *IEEE Transactions on Consumer Electronics*, vol. 58, pp. 1375–1383, Nov 2012.
- [10] "HM-16.2 Reference Software." https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.2/.
- [11] N. Purnachand, L. N. Alves, and A. Navarro, "Improvements to TZ search motion estimation algorithm for multiview video coding," in *19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 388–391, Apr 2012.
- [12] F. Bossen, "Common Test Conditions and Software Reference Configurations," Tech. Rep. JCTVC-L1100, Jan 2013.
- [13] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," Tech. Rep. VCEG-M33, ITU-T Video Coding Experts Group (VCEG), 2001.