Pattern-based integer sample motion search strategies in the context of HEVC

Georg Maier[a], Benjamin Bross[b], Dan Grois[b], Detlev Marpe[b], Heiko Schwarz[b], Remco C. Veltkamp[c], Thomas Wiegand[b]

[a]Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (Fraunhofer IOSB), Karlsruhe, Germany
[b]Fraunhofer Institute for Telecommunications - Heinrich Hertz Institute (Fraunhofer HHI), Berlin, Germany
[c]Department of Information and Computing Sciences, Utrecht University, The Netherlands

ABSTRACT

The H.265/MPEG-H High Efficiency Video Coding (HEVC) standard provides a significant increase in coding efficiency compared to its predecessor, the H.264/MPEG-4 Advanced Video Coding (AVC) standard, which however comes at the cost of a high computational burden for a compliant encoder. Motion estimation (ME), which is a part of the inter-picture prediction process, typically consumes a high amount of computational resources, while significantly increasing the coding efficiency. In spite of the fact that both H.265/MPEG-H HEVC and H.264/MPEG-4 AVC standards allow processing motion information on a fractional sample level, the motion search algorithms based on the integer sample level remain to be an integral part of ME. In this paper, a flexible integer sample ME framework is proposed, thereby allowing to trade off significant reduction of ME computation time versus coding efficiency penalty in terms of bit rate overhead. As a result, through extensive experimentation, an integer sample ME algorithm that provides a good trade-off is derived, incorporating a combination and optimization of known predictive, pattern-based and early termination techniques. The proposed ME framework is implemented on a basis of the HEVC Test Model (HM) reference software, further being compared to the state-of-the-art fast search algorithm, which is a native part of HM. It is observed that for high resolution sequences, the integer sample ME process can be speed-up by factors varying from 3.2 to 7.6, resulting in the bit-rate overhead of 1.5% and 0.6% for Random Access (RA) and Low Delay P (LDP) configurations, respectively. In addition, the similar speed-up is observed for sequences with mainly Computer-Generated Imagery (CGI) content while trading off the bit rate overhead of up to 5.2%.

Keywords: Video compression, motion estimation, inter-picture prediction, H.265, MPEG-H, HEVC, integer sample motion search.

1. INTRODUCTION

The hybrid video coding approach was first adopted in the H.261 standard, thereby incorporating the synergy of inter-picture prediction and linear transform coding of the prediction residual. Then, a similar hybrid design was also provided for all following video coding standards, including H.265/MPEG-H HEVC. According to the hybrid video coding approach, each frame is split into blocks, which are then encoded by using either the aforementioned inter-picture prediction or intra-picture prediction. While the intra-picture prediction exploits spatial redundancies within a frame, the inter-picture prediction exploits temporal redundancies, as typically exist in consecutive frames. Particularly, during the HEVC encoding process, the mode decision tool determines an appropriate prediction technique for each coding unit (CU) as well as the quadtree block partitioning into CUs for each coding tree unit (CTU).[1,2] In turn, the rate-distortion optimization (RDO) minimizes a cost function which is shown in Equation (1).[3] The overall cost J is calculated by using a weighted combination of the rate cost R and the distortion cost D. Obviously, determining the most efficient coding mode requires each CU to be encoded in every possible mode, thereby resulting in a high computational burden for an encoder.

$$J = D + \lambda R \tag{1}$$

In case of the inter-picture prediction, luma and chroma coding blocks of the aforementioned CU are encoded by means of displacements relative to matching blocks in prior encoded reference pictures. The displacement between the actual block and the reference block is denoted by the *motion vector* (MV) $\vec{m} = (m_x, m_y)^\top$, which typically is determined as a result of the search algorithm. It is important to note that the block matching search algorithm does not yield a vector representing *true* motion, but rather tries to find an optimum with respect to an implemented cost function such as Equation (2a). Given a set of reference pictures T and a search window M limiting the motion vectors, the motion parameters can be determined at the encoder by selecting the reference picture Δt^* and the motion vector $\vec{m}^*$ that minimize the cost function according to Equation (2).[4]

$$J_{\text{motion}}(\Delta t, \vec{m}) = D(\Delta t, \vec{m}) + \lambda_{\text{motion}} R(\Delta t, \vec{m}) \tag{2a}$$

$$(\Delta t^*, \vec{m}^*) = \underset{\Delta t \in T, \vec{m} \in M}{\arg\min} \; J_{\text{motion}}(\Delta t, \vec{m}) \tag{2b}$$

However, since the MVs in the resulting MV field are likely to be highly correlated,[5] the motion vector is signaled in the bit stream by using predictive coding as *motion vector difference* (MVD) to a certain *motion vector predictor* (MVP) which is known to both the encoder and decoder in terms of a shared rule.[6] Modern search algorithms adopt this predictive characteristic, and therefore, the MVP also serves as the starting point of the subsequent search. In other words, the search window M is spanned using the prior selected MVP as the center. Then, the integer sample positions inside this search window are considered as possible displacements, whereas the distortion, which is typically described by means of the sum of absolute differences (SAD), serves as a quality metric for the considered displacements. In the context of the HM reference encoder used for this paper, the distortion D is calculated by means of SAD while the rate R is an estimate of the bits used to code the MVD. Furthermore, the motion-based Lagrange multiplier λ_{motion} is derived from the Lagrange multiplier used for the mode decision in terms of $\lambda_{\text{motion}} = \sqrt{\lambda}$, as described in the literature.[7]

Instead of expressing the displacements in units of integer sample positions, MVs can also be defined as displacements with fractional sample precision requiring interpolation filters to obtain the sample values at fractional positions. The H.264/MPEG-4 AVC standard was the first widely used video coding standard that introduced MVs in units of one quarter of the distance between samples (i.e. the quarter sample distance), and its successor, the H.265/MPEG-H HEVC continued employing this approach. Therefore, upon determining the integer sample position, a so called fractional sample refinement process considers fractional sample positions, which are adjacent to the prior determined integer sample position, up to the desired precision.

As generally discussed above, in the hybrid video coding scheme, the error resulting from the prediction is encoded in the bitstream, and therefore, the prediction accuracy is highly essential to the coding efficiency. The RD-based optimal integer sample search is carried out by the so-called *full search*, which tests every possible displacement of the block to be encoded inside the search window or search range by calculating the implemented cost function. Obviously, the *full search* method is very costly in terms of computational load, and consequently, it cannot be employed for low-delay or real-time/live encoding. As a result, the integer sample motion estimation evolved to a research topic of a high interest, yielding numerous approaches attempting to sub-sample the search space in order to achieve near optimal results with respect to the RD performance, while lowering the computational complexity. However, when trying to solve the optimization problem, there is a risk of getting stuck in local minima, which do not correspond to the global minimum.

In Figure 1, two examples of error surfaces of different integer sample displacements within a 16 × 16 search windows are plotted. It can be seen, several of the aforementioned local minima exist, where fast searches can get trapped in. As new standards evolve, the type of the video signals which are fed to new encoders also evolves. On the one hand, focusing on higher video resolutions, integer sample search algorithms are operating on a more fine grained raster, which changes the size of motion measured in samples, since one sample covers a smaller area of the picture compared to lower resolution sequences. On the other hand, the higher frame rates typically reduce the amount or size of motion between consecutive frames. However, in case of the HEVC encoding, other changes in the coding scheme, such as the variable block sizes, should be considered as well for the design of a motion search algorithm.

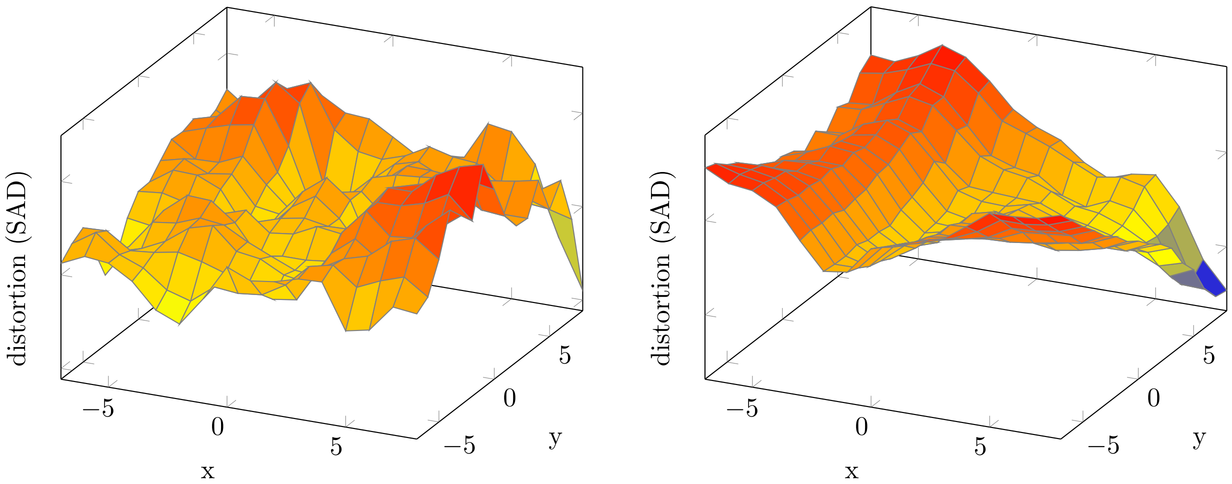


Figure 1: Examples of the matching error surface within a search window of 16×16 luma samples. As it can be seen, there is a large risk for fast search algorithms to get trapped in local minima. These examples represent values recorded for the test sequence "Race Horses".[8]

In this paper, an integer sample search algorithm framework is proposed consisting of multiple consecutive stages. Various options for these different stages are considered and their impact on the computation time and coding efficiency is investigated. Among other insights, experimental results suggest that the application of irregular shaped, rotating patterns appears beneficial in terms of the achieved trade-off between the required computation time and RD performance. Also, various early termination criteria are evaluated within the proposed framework.

The rest of the paper is organized as follows. Section 2 provides a brief overview of related work with regard to the HEVC standard as well as to recent fast integer sample ME algorithms. Then, in Section 3, the proposed integer sample ME framework is presented, further providing strategies how to apply and optimize known techniques (which are reviewed in Section 2) towards a proposed algorithm that considers the RD performance and required computation time. The test methodology and simulation settings, which are applied for experiments conducted in the course of this paper as well as the fast motion search of the HM reference software (that serves as a reference for all experiments) are described in Section 4. After that, the experimental results are presented and discussed in Section 5. Finally, Section 6 concludes this paper and gives an outlook of future work.

2. BACKGROUND AND RELATED WORK

The H.265/MPEG-H High Efficiency Video Coding (HEVC) standard[9] achieves significant bit-rate savings of about 50% compared to its predecessor H.264/MPEG-4 AVC[10] for essentially the same subjective quality. Several detailed coding efficiency comparisons with both H.264/MPEG-4 AVC and H.262/MPEG-2 standards has been conducted upon finalizing HEVC version 1 in January, 2013[7] followed by detailed comparisons with other video coding schemes, such as VP9.[11, 12] In addition, studies regarding the HEVC software decoding performance and complexity for high resolution sequences have been also performed.[13]

As already noted in the introduction, the inter-picture prediction in HEVC relies on MVs with quarter sample accuracy similarly to the inter-picture prediction in H.264/MPEG-4 AVC. However, for the MVP derivation, the median based predictor using spatially neighboring MVs was replaced by a competitive scheme, which is applied on a set of candidates derived from both spatial and temporal neighboring MVs. In HEVC, the *Advanced Motion Vector Prediction* (AMVP)[14] was introduced, which defines a shared rule between encoder and decoder of how to fill a list of two MVP candidates based on spatially and temporally neighboring motion vectors.[15] Then, the encoder needs to signal to the decoder which one of the two MVP candidates is selected to be the MVP.

In the following subsection below, traditional pattern based integer sample ME techniques as well as more advanced predictive frameworks and early termination thresholds for the search algorithm are overviewed, further including an assessment of their potential deployment in HEVC.

2.1 Pattern based fast integer sample motion estimation

Early works on integer sample motion estimation suggest sub-sampling of the search space by repeatedly applying a certain search pattern to navigate through the search space.[16–19] Algorithms of this kind might be appreciated due to their low complexity and their easy implementation in hardware, however they typically suffer from rather high loss in rate-distortion (RD) performance caused by the assumption that the distortion surface is a convex function of the MVs and the distortion decreases monotonically towards the position yielding the minimum distortion. Within that context, irregular shaped patterns such as an irregularly shaped diamond pattern have also been proposed.[20] The irregularly shaped diamond pattern employs two possible rotations and adapts according to whether a better candidate was found on an horizontal or vertical displacement from the origin, which corresponds to the best candidate identified so far. This way, the pattern is supposed to better follow the motion during consecutive phases of application. Similarly, a hexagonal shaped pattern to be used in two possible rotations was introduced.[21]

2.2 Predictive frameworks for fast integer sample motion estimation

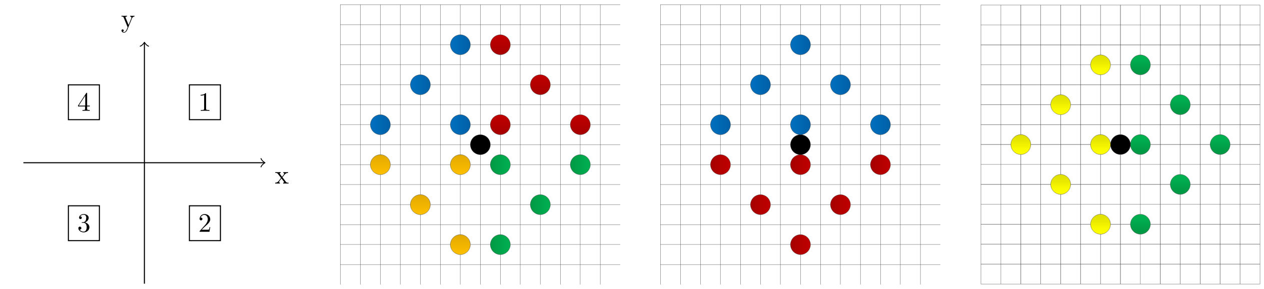
Recent works are based on the assumption of homogeneous motion fields, thereby leading to the inclusion of MVPs into the motion search.[5, 22–26] Such predictive techniques typically derive information from the prior selected MVP, such as the expected size and direction of motion. Based on this information, a conditional initial pattern application can be performed prior to execution of a repeating search pattern.[27] For this purpose, a quarter sparse diamond (QSD) pattern as well as a half sparse diamond (HSD) pattern, each of which exists in four possible rotations, is proposed. The corresponding pattern search options are further illustrated in Figure 2. The selection of the appropriate pattern and the rotation thereof is based on the largest angle existing between three correlated MVs, namely two spatial (left and upper) and one temporal adjacent MV, denoted as $\vec{p}_1$, $\vec{p}_2$, and $\vec{p}_3$.

Let $\alpha = \angle(\vec{v}, \vec{p}_i)$ be the angle between the y-axis and the MV $\vec{p}_i$ with $\vec{v} = (0,1)^\top$, and $\mathcal{A} = \{\alpha_1, \alpha_2, \alpha_3\}$, then the initial pattern is selected as follows:

- **QSD** if all three MVs are located within one quadrant:
    1. Red QSD pattern if $\forall \alpha_i \in \mathcal{A} : 0 < \alpha_i < \frac{\pi}{2}$;
    2. Green QSD pattern if $\forall \alpha_i \in \mathcal{A} : \frac{\pi}{2} < \alpha_i < \pi$;
    3. Yellow QSD pattern if $\forall \alpha_i \in \mathcal{A} : \pi < \alpha_i < \frac{3\pi}{2}$;
    4. Blue QSD pattern if $\forall \alpha_i \in \mathcal{A} : \frac{3\pi}{2} < \alpha_i < 2\pi$.
- **HSD** if all three MVs are located within two neighboring quadrants:
    1. Blue HSD pattern if $\forall \alpha_i \in \mathcal{A} : -\frac{\pi}{2} < \alpha_i < \frac{\pi}{2}$;
    2. Green HSD pattern if $\forall \alpha_i \in \mathcal{A} : 0 < \alpha_i < \pi$;
    3. Red HSD pattern if $\forall \alpha_i \in \mathcal{A} : \frac{\pi}{2} < \alpha_i < \frac{3\pi}{2}$;
    4. Yellow HSD pattern if $\forall \alpha_i \in \mathcal{A} : \pi < \alpha_i < 2\pi$.
- Blue HSD pattern if all three MVs are located in three different or two opposite quadrants.

In addition, scaling of a pattern in accordance with the expected size of motion has been proposed.[28] For this purpose, the spatially adjacent block on the left of the current block is selected as a predictor $\vec{p}$, and its length is formulated as shown in Equation (3). In this zero-biased approach, a cross-shaped pattern consisting of points $(0,0)$, $(0,d)$, $(0,-d)$, $(d,0)$ and $(-d,0)$ is used.

$$d = \max\{|p_x|, |p_y|\} \tag{3}$$



(a) Numbering of the quadrants relative to the center of search

(b) QSD pattern

(c) HSD pattern (1)

(d) HSD pattern (2)

Figure 2: Initial search pattern variations as proposed in the context of the modified diamond search.[27]

2.3 Thresholds for early termination

Thresholds are a convenient way to steer the decision process whether a MV identified at an early stage of the search yields acceptable RD performance, and hence avoid checking further candidates, which are unlikely to lead to a significantly better distortion performance. As a result, the computation time is reduced due to performing a smaller number of distortion calculations.

The related work with this regard may be further categorized according to the purpose of the threshold. For instance, some authors consider thresholds as a way to early identify a (close to) zero motion situation. This strategy is sometimes referred to as *zero motion prejudgement* (ZMP).[28] Thresholds may further serve the purpose of determining a distortion of an identified MV based on the likeliness of finding a better MV, which yields lower distortion. Adaptive calculation of thresholds is typically realized by designing a statistical model that describes the expected distortion in a best way, whereas spatial and / or temporal correlation is exploited (e.g. by using the distortion from neighboring MVs to derive the threshold).[22, 29–36]

Adaptive thresholding for motion vector search in the HEVC context has also been proposed.[37] The authors suggest the computation of 13 thresholds to cover possible combinations of the coding unit depth in the coding tree along with the partition sizes, and employ the coding costs obtained from the first picture in the same group of pictures (GOP).

3. PROPOSED INTEGER SAMPLE ME FRAMEWORK FOR HEVC

The integer sample ME strategies presented in the previous section to achieve reasonable trade-offs between RD performance and computational load are targeting a number of applications. In the course of this paper, an integer sample ME algorithm is developed targeting live encoding applications, which means that a small loss in coding efficiency is acceptable for a large speed up. In order to achieve that, a selection of approaches mentioned in Section 2 has been implemented in the HM reference encoder[38] and their performance has been evaluated. Although the HM encoder is far away from performing real-time encoding,[39] conclusions for real-time/live encoding can be drawn when considering only the speedup of the integer sample ME part. In this initial evaluation phase, where several technique have been implemented in HM, rather promising results were obtained for the *modified diamond search*[27] (MDS). Hence, the HM implementation of MDS serves as a basis for further optimizations and experimental results are reported in Section 5. Based on the different steps of the MDS algorithm, an optimized integer sample ME framework was derived, thereby enabling the utilization of beneficial strategies that have been proposed in other contexts.

The flowchart of the proposed integer sample ME framework is illustrated in Figure 3. It consists of defining the starting point of the search by using motion vector predictors, i.e. the search center, an initial onetime execution of a search pattern, a second search pattern that can be applied repeatedly, a refinement phase at the end and placeholders for implementation of threshold-based early termination criteria between those. In turn, this modular breakdown of integer sample ME enables a combination of different approaches as well as a thorough experimentation to evaluate different combinations.

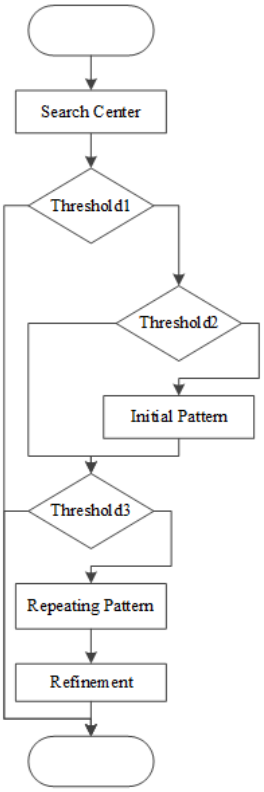


Figure 3: Flowchart of the proposed integer sample ME framework for HEVC.

In the next subsections, promising concepts and optimizations are presented for each stage of the proposed optimized integer sample ME framework followed by the description of an algorithm, which incorporates one solution for each stage . Results for the different combinations and the proposed algorithm are reported in Section 5. Particularly, the search center and motion vector predictors are presented in Section 3.1, followed by discussing the modified initial QSD and HSD patterns in Section 3.2. Then, the repeating patterns and thresholds for early termination are described in Section 3.3 and Section 3.4, respectively.

3.1 Search center and motion vector predictors

Following predictive techniques, it is suggested to use the MVP selected through AMVP as the origin of the search window instead of the zero-biased MDS algorithm as proposed in literature.[27] However, the MDS algorithm makes use of MVPs from neighboring MVs to steer the application of the initial search pattern. As intended in the MDS algorithm, the two spatial (left and upper) and one temporal adjacent MV are included in a set of additional motion predictors. This part is inherited but the search center MVP from AMVP is added to this set of predictors steering the search. In the course of this paper, the complete set of predictors is denoted as $\mathcal{P}$, where $1 \leq |\mathcal{P}| \leq 4$ holds and $\vec{p}_0$ corresponds to the AMVP. In summary the set consists of:

- Best AMVP candidate (search center);
- MV from left neighbor if available (A1 in AMVP);
- MV from upper neighbor if available (B1 in AMVP);
- MV from temporal neighbor if available.

3.2 Initial search patterns

In addition to the initial QSD and HSD patterns and a modification that includes scaling of the patterns as suggested in the literature, a simplification and a combination of both is suggested in this paper. Overall, the following alternations of the initial pattern application were evaluated:

- QSD and HSD pattern as suggested in MDS;[27]
- Scaled QSD and HSD pattern;[28]
- Simplified QSD pattern;
- Combination of scaled and simplified QSD pattern.

Firstly, the application of the QSD and HSD pattern is combined with a scaling technique in accordance to the expected size of motion d, which is defined as the maximum distance d_i according to Equation (3) of all available $p_i \in \mathcal{P}$. It should be noted that the original MDS algorithm also uses expected distance information in terms of a threshold whether the initial search pattern should be applied at all. This will be further detailed in Section 3.4 on thresholds for early termination. Equation (4a) to Equation (4d) illustrate where the search points are consequently located relative to the center of search for the QSD pattern, whereas QSD_j is applied whenever the predictors are located in quadrant j as depicted in Figure 2a. The corresponding locations of the search points for the adaptively scaled HSD pattern are provided in Equation (5a) to Equation (5d). Consequently, $\mathrm{HSD}_{i,j}$ is applied whenever the predictors are located in quadrant i and j. In accordance with the MVP set description provided above, the set of angles to be used for pattern selection is extended by the AMVP predictor $\vec{p}_0$ leading to $\mathcal{A} = \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\}$. However, the conditions for selecting the appropriate pattern and rotation remain unchanged with respect to the approach discussed in Section 2.2.

Adaptively scaled QSD patterns:

$$\mathrm{QSD}_1 = (1,1), \quad (1,d), \quad (d,1), \quad (\lfloor(d/2)\rfloor, \lfloor(d/2)\rfloor) \tag{4a}$$

$$\mathrm{QSD}_2 = (1,-1), \quad (1,-d), \quad (d,-1), \quad (\lfloor(d/2)\rfloor, -\lfloor(d/2)\rfloor) \tag{4b}$$

$$\mathrm{QSD}_3 = (-1,-1), \quad (-1,-d), \quad (-d,-1), \quad (-\lfloor(d/2)\rfloor, -\lfloor(d/2)\rfloor \tag{4c}$$

$$\mathrm{QSD}_4 = (-1,1), \quad (-1,d), \quad (-d,1), \quad (-\lfloor(d/2)\rfloor, \lfloor(d/2)\rfloor) \tag{4d}$$

Adaptively scaled HSD patterns:

$$\mathrm{HSD}_{4,1} = (-d,1), \quad (0,1), \quad (d,1), \quad (-\lceil(d/2)\rceil, \lceil(d/2)\rceil), \quad (\lceil(d/2)\rceil, \lceil(d/2)\rceil), \quad (0, d+1) \tag{5a}$$

$$\mathrm{HSD}_{1,2} = (1,d), \quad (1,0), \quad (1,-d), \quad (\lceil(d/2)\rceil, \lceil(d/2)\rceil), \quad (\lceil(d/2)\rceil, -\lceil(d/2)\rceil), \quad (d+1, 0) \tag{5b}$$

$$\mathrm{HSD}_{2,3} = (-d,-1), \quad (0,-1), \quad (d,-1), \quad (-\lceil(d/2)\rceil, -\lceil(d/2)\rceil), \quad (\lceil(d/2)\rceil, -\lceil(d/2)\rceil), \quad (0, -d-1) \tag{5c}$$

$$\mathrm{HSD}_{3,4} = (-1,d), \quad (-1,0), \quad (-1,-d), \quad (-\lceil(d/2)\rceil, \lceil(d/2)\rceil), \quad (-\lceil(d/2)\rceil, -\lceil(d/2)\rceil), \quad (-d-1, 0) \tag{5d}$$

A second proposed modification of the algorithm is to simplify the search pattern selection. In the original MDS algorithm as presented in Section 2.2, the applied search pattern depends on the angle between the three initially proposed predictors. Now, instead of calculating the angle and selecting one of the QSD or HSD patterns, one QSD pattern is applied for each quadrant a predictor as well as the MVP is located in. This approach is motivated by the fact that a HSD pattern maps well into the two neighbouring QSD patterns. While this modification omits the already mentioned angle calculation, it increases the number of search points considered in this step, since more than one pattern might come to application. For example in the best case, all predictors are in one quadrant so that just one QSD pattern is used while in the worst case, all four QSD patterns are used when all three predictors and the MVP are pointing in different quadrants.

Thirdly and lastly, two above-mentioned modifications are combined. Each QSD pattern is therefore applied in a scaled manner, according to a corresponding predictor.

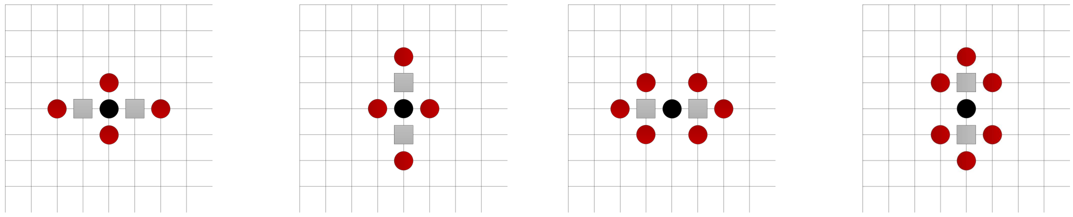

(a) hor. diamond (b) ver. diamond (c) hor. hexagon (d) ver. hexagon

Figure 4: Illustration of two rotating patterns:[20,21] red dots represent the search positions around the center (a black dot), while grey squares represent the additional positions to be checked; when the center is the optimum, then applying a repeated pattern is terminated.

3.3 Repeating search patterns

While the original MDS algorithm suggests repeating a small diamond shaped pattern, which consists of four new positions for each iteration as long as a new optimum may be identified, the adaption of directional irregular shaped patterns is proposed in this context. Consequently, the following patterns are considered:

- Small diamond pattern[27] consisting of four search points, in vertical and horizontal directions, which neighbor the current location with a distance of two samples;
- Rotating diamond pattern[20] (see Figures 4a and 4b);
- Rotating hexagonal pattern[21] (see Figures 4c and 4d).

The small diamond pattern comes with the advantage of low complexity due to the small number of search points to be considered during each iteration. However, the rotating diamond pattern consists of the same small number of search points, yet suggesting to better follow a line of motion, as it was already mentioned in Section 2.1. The same advantage may be obtained when considering the rotating hexagonal pattern, which consists of a larger number of search points and hence requires more calculations per iteration compared to the previous two discussed patterns. For both rotating patterns, two positions adjacent to the last optimum exist, but they are not considered due to the given search pattern (grey squares in Figure 4). Therefore, the refinement stage consists of comparing them with the optimum at the center.

3.4 Thresholds for early termination and conditional execution

Many early termination techniques proposed in the literature suffer from one major drawback, namely the complexity of the model steering the decision process.[31,32,34,36] Clearly, this is not the case for static thresholds, which fail to adopt to a given video sequence. For this reason, only adaptive, yet simple early termination formulations are considered in this context, as described in the following.

According to the proposed framework flowchart, presented in Figure 3, the early termination thresholds can be divided in three categories:

- Zero motion prejudgement (ZMP)-based for *threshold1* to terminate the search early;
- Predicted size of motion for *threshold2* for conditional execution of the initial search pattern, as intended in the MDS algorithm;[27]
- Predictive MV costs-based for *threshold3* to terminate the search early.

In the following Section 3.4.1 to Section 3.4.3, the above mentioned three categories are addressed in detail. It should be noted that the result of a cost function, in accordance to Equation (2), for the corresponding block of $\vec{p}_i$ is denoted below as c_i.

3.4.1 Zero motion prejudgement (ZMP)-based threshold1

The purpose of this model is to decide, when zero motion is expected. As a consequence, no search is conducted at all, if zero motion is expected. The ZMP can be based on MVs and costs of adjacent blocks, whereas four ZMP conditions are suggested in the following. In the first condition, the expected length of motion d is compared to a threshold denoted as t_{init} and the MVP is required to be zero, while alike the definition given in Section 3.2, d corresponds to the maximum distance d_i in Equation (3) of all available $p_i \in \mathcal{P}$. The following two conditions require all predictors of $\mathcal{P}$ to be equal to zero, while the third condition complements the basic condition by requiring the set to be of a certain size. The fourth condition also checks whether the MVP is zero as well as compares the costs associated to that MVP, denoted as c_0, to the costs of all other predictors c_i (with i being greater than zero).

1. Expect zero motion when $d < t_{\text{init}}$ && $|\vec{p}_0| = 0$. (ET1);
2. Expect zero motion when $\forall \vec{p}_i \in \mathcal{P} : |\vec{p}_i| = 0$. (ET2);
3. Expect zero motion when $\forall \vec{p}_i \in \mathcal{P} : |\vec{p}_i| = 0$ && $|\mathcal{P}| \geq 3$. (ET3);
4. Expect zero motion when $|\vec{p}_0| = 0$ && $c_0 \leq \min\limits_{\vec{p}_i \in \mathcal{P} \setminus \vec{p}_0} c_i$ (ET4).

3.4.2 Expected motion size-based threshold2

As has been mentioned, the MDS algorithm as proposed in[27] includes a criterion for conditional execution of the initial search pattern. In the presented approach, this threshold is maintained, although its application depends on the strategy chosen regarding the initial search pattern. Hence, for results presented in Section 5, *threshold2* is implemented as $d \geq t_{init}$.

3.4.3 Predictive MV costs-based threshold3

Furthermore, simplistic models to terminate the search after application of the initial search pattern are considered. Let c_0 be the obtained cost, when employing the MVP $\vec{p}_0$, and let c_{init} be the best cost obtained upon applying the initial search pattern.

1. Terminate search if $c_{\text{init}} < c_0$ (ET5);
2. Terminate search if $c_{\text{init}} < \max\limits_{\vec{p}_i \in \mathcal{P}} c_i$ (ET6);
3. Terminate search if $c_{\text{init}} < \frac{1}{|\mathcal{P}|} \sum_{i=0}^{|\mathcal{P}|} c_i$ (ET7);
4. Terminate search if $c_{\text{init}} < \min\limits_{\vec{p}_i \in \mathcal{P}} c_i$ (ET8).

3.5 Proposed algorithm

In addition to the evaluation of modifications as discussed above, an algorithm for the proposed integer sample ME framework is defined, thereby combining the presented techniques and tools.

According to the proposed algorithm as presented in Algorithm 1, first the integer sample search is conducted only if a non-static block is estimated by using the strictest proposed ZMP technique (ET3). In case the condition does not hold true and the search is executed, the scaled QSD patterns are only applied in each quadrant of an existing predictor if the condition of *threshold2* holds true. Also, if the best cost is thereby achieved, and its value is lower than the cost associated with the corresponding predictors of $\mathcal{P}$, the search is terminated at this point (ET8). At any other case, the rotating diamond search pattern is applied as long as a new optimum is found, including the previously described refinement phase subsequent to the execution of these repeating search patterns.

In the following Section 4, the test methodology and simulation settings are defined and discussed in detail.

Algorithm 1 Example configuration of new integer search

```
Calculate cost for MVP position and set this position as starting point and centre of search
Calculate length of two spatial adjacent MVs and temporal adjacent MV

if at least three predictors including the MVP are available and all have length 0 then
    Terminate the search
end if

if based on these lengths, a huge motion is expected then
    for each of the available predictors do
        Apply QSD pattern in quadrant of the predictor, scaled in accordance to the length of this MV
    end for
end if

if current best cost found is less than all costs associated with the predictors then
    Terminate the search
end if

while a better candidate was identified during the last iteration do
    Apply the DDS search pattern
end while

Calculate costs for directly neighbouring positions that have not been checked yet
```

4. TEST METHODOLOGY AND SIMULATION SETTINGS

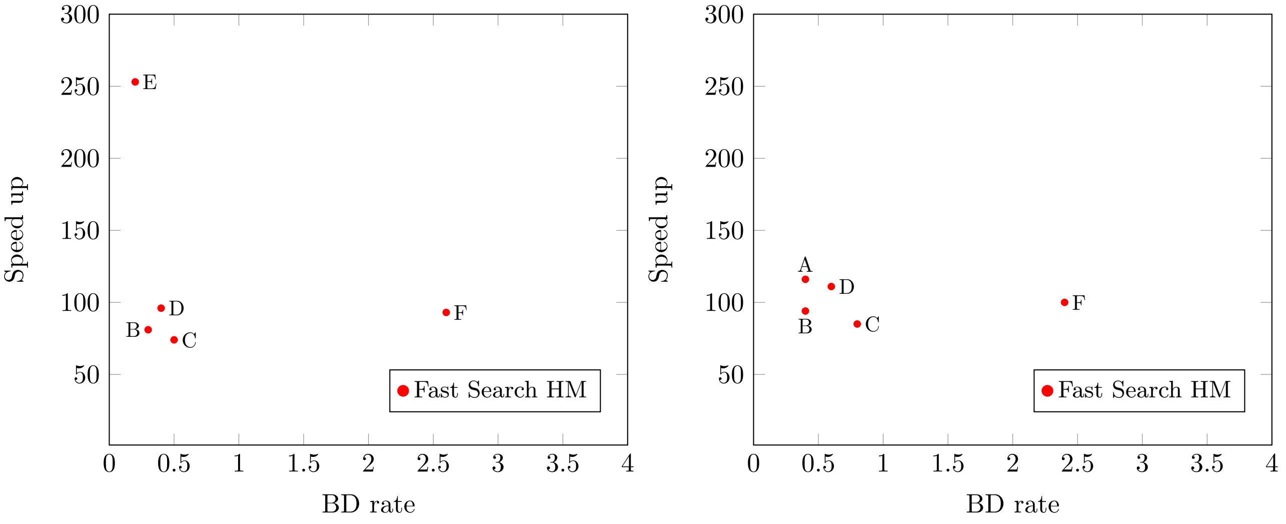
The HEVC reference software (HM) serves as a basis for the experimental validation, for instance when considering optimization strategies for a compliant encoder. Particularly, HM 12.0[38] is employed for this purpose, and the Random Access (RA) and Low Delay P (LDP) configurations, as defined in the HEVC Common Test Conditions (CTC),[8] are used for obtaining the experimental results. Regarding the motion estimation, these two configurations are significantly different, mostly in terms of the following two aspects.[40] First, the RA configuration allows *B-frame* (bi-directional) prediction, which means that the final motion parameters may consist of a weighted combination of two MVs from potentially two different reference blocks, while the LDP configuration is generally restricted to the *P-frame* prediction, where the motion parameters are limited to one MV from one reference block. Second, the coding order of the frames also is significantly different.

Table 1 further provides an overview of the test sequences, which have been considered. Regarding the content of the sequences, it is noted that *Class E* sequences represent typical video conferencing content, which corresponds to relatively low motion activity and *Class F* sequences mainly consist of Computer-Generated Imagery (CGI) content.

Table 1: Evaluated Common Test Conditions (CTC) sequences[8]

Class	Resolution	Nr. of sequences	RA	LDP
A	2660x1600	4	✓	
B	1920x1080	5	✓	✓
C	832x480	4	✓	✓
D	416x240	4	✓	✓
E	1280x720	3		✓
F	1280x720	2	✓	✓
	1024x768	1		

HM incorporates a full search scheme, where the brute force algorithm is applied as well as a fast search



(a) LDP (b) RA

Figure 5: Illustration of the encoding efficiency when applying the fast integer search algorithm as present in HM 12.0 compared to the full search.

scheme, which is a derivation of the EPSZ scheme,[22] however not a strict implementation of it. Usage of the full search scheme is optimal in terms of RD performance, and hence can be seen as a benchmark reference, while the fast search strategy reduces computational complexity at the cost of RD performance. The difference in RD performance is described by means of the Bjøntegaard-Delta (BD)-rate,[41] whereas negative values correspond to a coding efficiency gain. Figure 5 depicts coding efficiency losses, when choosing the fast search algorithm over the full search. As a measure of computational complexity, the time required to perform the integer search was measured, whereas the difference in time is formulated as shown in Equation (6). Moreover, for all experiments conducted in the course of this work, the search range was restricted to 64 luma samples.

$$\text{speed up} = \frac{\text{time}_{\text{reference}}}{\text{time}_{\text{test}}} \tag{6}$$

In the following Section 5, the experimental results are presented and discussed in detail.

5. EXPERIMENTAL RESULTS

The results presented in this section were obtained by using the test methodology and simulation settings, as defined in Section 4. All options (variations) for the different stages of the integer sample motion search framework, as presented in Section 3, are listed in Table 2. For all experiments, the value of the initial threshold t_{init} was determined empirically to be equal to 4, when it is applicable.

In the course of this section, results for all these search options are provided, followed by the results of executing the proposed algorithm for the optimized integer sample ME framework, which in turn combines the most promising ones.

5.1 Comparison results of different search options

The impact of all discussed search options to obtain the afore-described algorithm for the proposed integer sample ME framework is presented in Figure 6, whereas the results are averaged over all four quantization parameters for all sequences of the corresponding CTC class,[8] which is considered of a particular interest for the corresponding configuration. It should be noted that class E sequences represent the LDP configuration, and class A sequences represent the RA configuration. In addition, the reference/anchor with zero BD-rate and zero speed up is the fast search of HM. Results for the originally proposed MDS algorithm are included for comparison as well.

Table 2: Notation of the search options, which are discussed in Section 3.

Option	**Thresh. 1**	**Thresh. 2**	**Initial pattern**	**Thresh. 3**	**Repeating pattern**
MDS		$d \geq t_{init}$	QSD / HSD		small diamond pattern
1		$d \geq t_{init}$	scaled QSD / HSD		small diamond pattern
2		$d \geq t_{init}$	QSD		small diamond pattern
3		$d \geq t_{init}$	scaled QSD		small diamond pattern
4		$d \geq t_{init}$	QSD / HSD		rotating diamond pattern
5		$d \geq t_{init}$	QSD / HSD		rotating hexagonal pattern
6	ET1	$d \geq t_{init}$	QSD / HSD		small diamond pattern
7	ET2	$d \geq t_{init}$	QSD / HSD		small diamond pattern
8	ET3	$d \geq t_{init}$	QSD / HSD		small diamond pattern
9	ET4	$d \geq t_{init}$	QSD / HSD		small diamond pattern
10		$d \geq t_{init}$	QSD / HSD	ET5	small diamond pattern
11		$d \geq t_{init}$	QSD / HSD	ET6	small diamond pattern
12		$d \geq t_{init}$	QSD / HSD	ET7	small diamond pattern
13		$d \geq t_{init}$	QSD / HSD	ET8	small diamond pattern
Algorithm 1	ET3	$d \geq t_{init}$	scaled QSD	ET8	rotating diamond pattern

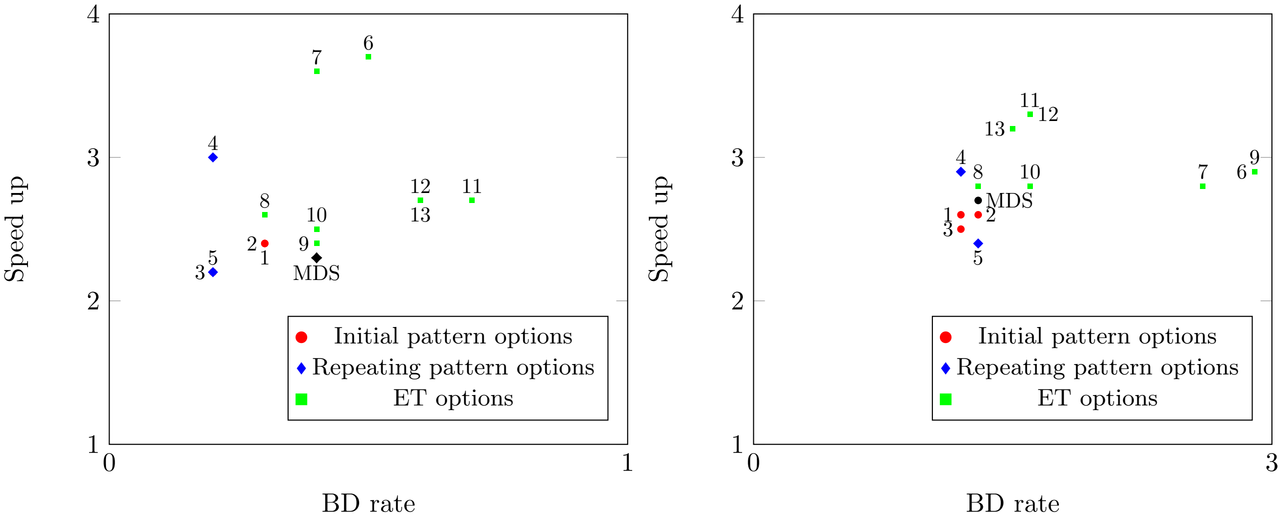
As it can be seen from Figure 6a for LDP, all initial search pattern variations as well as the repeating search pattern variations achieve a better trade-off in terms of computation time and RD performance, when compared to MDS. For RA, only option 4 (i.e. the rotating diamond pattern) provides a better trade-off compared to MDS, while option 3 (i.e. the simplified and scaled QSD initial pattern) has only a marginal reduced speed up as shown in Figure 6b. Therefore, the simplified and scaled initial pattern together with the rotating diamond pattern are considered to be part of the proposed algorithm Algorithm 1. With respect to the discussed early termination techniques (see Section 3.4), it can be concluded that all variations result in a reduction of required computation time. However, there is a significant difference in terms of RD performance, and also a dependency on the encoding configuration (RA or LDP). Particularly, for LDP, the ZMP is very effective in terms of computation time reduction, while keeping losses in RD performance rather low. This holds especially for ET1 and ET2. Yet, for RA, the results are quite the opposite, and for these conditions, the performance is the worst in terms of RD. The search options, which yield acceptable losses in RD performance, while reducing computation time, are ET3 and ET8.

5.2 Results for implementing the proposed algorithm

In Section 3.5, an algorithm combining several of the fast search options reported above was introduced and outlined in Algorithm 1. Results for this algorithm are presented in Figure 7 averaged over classes of sequences and in Table 3 for each sequence individually. As can be seen, the algorithm achieves a noteworthy speed up for both RA and LDP configurations; however, the losses in RD performance are considerably higher for the RA configuration compared to the LDP configuration. In addition, for both configurations the RD performance is particularly high for class F sequences, which can be explained by the high amount of CGI content (i.e. unnatural motion within the frames), as already mentioned in Section 4. Another observation that is valid for both RA and LDP configurations is that, generally, the higher the resolution of the test sequences is, the better the resulting trade-off. This aspect is considered to be very important for a potential use of the proposed algorithm in an HEVC compliant live encoder, since HEVC was developed with a particular scope on high resolution videos.

6. CONCLUSION AND OUTLOOK

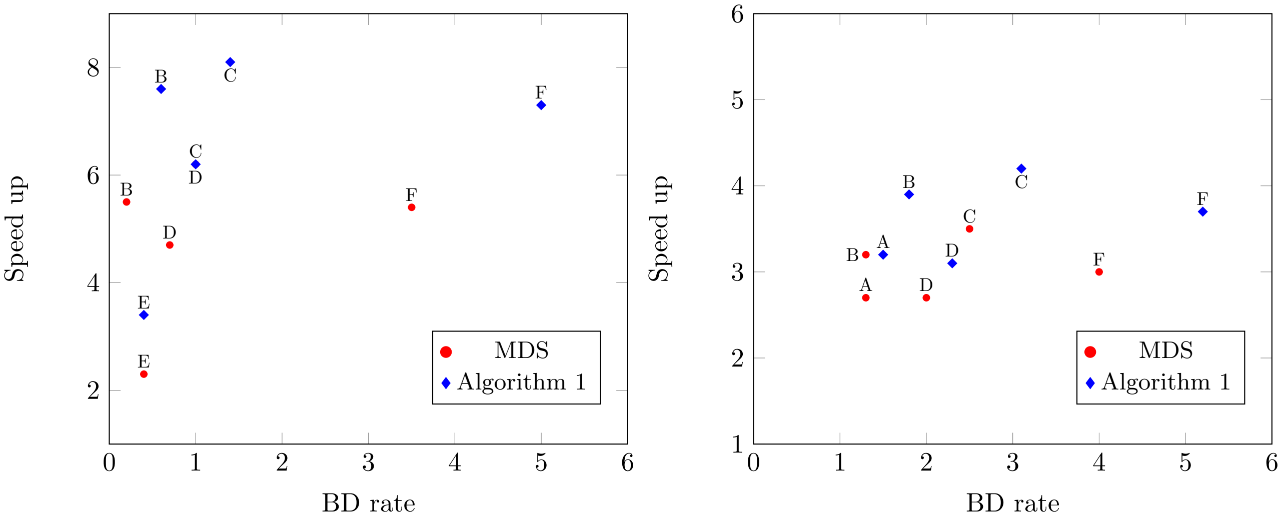
In this work, a framework for an integer sample search algorithm was developed, further implementing and evaluating different variations (options) for the optimized motion search. In general, the proposed search options may be grouped into a number of categories, such as the initial search pattern, repeating search pattern, and early termination approaches. The experimental results for both JCT-VC RA and LDP coding configurations were



(a) LDP, class E sequences

(b) RA, class A sequences

Figure 6: Results for algorithm options as discussed in Section 3 compared to the fast search option in HM 12.0. Numbering corresponds to Table 2.



(a) LDP

(b) RA

Figure 7: Results obtained when applying integer sample search as outlined in Algorithm 1 and MDS[27] for sequences listed in Table 1 using the fast search scheme of HM as a reference.

Table 3: Detailed overview of BD rates [%] and speed-ups for the MDS algorithm and the algorithm outlined in Algorithm 1 for each sequence of the test set in reference to the fast search scheme of HM.

		LDP				**RA**			
		MDS		**Proposed**		**MDS**		**Proposed**	
Class	**Sequence**	**BD[%]**	**S.-up**	**BD[%]**	**S.-up**	**BD[%]**	**S.-up**	**BD[%]**	**S.-up**
A	Traffic	n/a	n/a	n/a	n/a	2.8	1.6	3.1	2.1
	PeopleOnStreet	n/a	n/a	n/a	n/a	2.2	4.0	2.6	4.5
	Nebuta	n/a	n/a	n/a	n/a	0.2	2.7	0.2	3.2
	SteamLocomotive	n/a	n/a	n/a	n/a	0.0	2.1	0.1	2.5
B	Kimono	0.1	7.9	0.8	11.2	1.6	4.2	2.0	4.9
	ParkScene	0.0	3.6	0.7	5.0	1.0	1.8	1.4	2.2
	Cactus	0.3	4.5	0.4	6.3	1.0	2.7	1.3	3.3
	Basketball Drive	0.2	7.2	0.7	10.1	2.4	4.4	3.6	5.4
	BQTerrace	0.4	3.2	0.6	4.4	0.6	2.3	0.6	2.9
C	Basketball Drill	0.9	5.8	1.6	7.4	2.1	3.3	2.8	4.0
	BQMall	1.0	4.7	1.2	6.1	2.2	2.7	2.2	3.2
	PartyScene	0.6	4.8	0.8	6.4	0.9	2.9	1.3	3.6
	RaceHorses	1.5	8.8	2.0	10.9	4.8	5.0	6.1	5.7
D	Basketball Pass	0.2	3.0	1.0	3.9	1.5	1.7	2.0	2.3
	BQSquare	0.5	1.6	0.5	2.5	0.3	1.4	0.3	1.7
	BlowingBubbles	0.5	5.1	0.5	7.0	1.1	3.1	1.3	3.6
	RaceHorses	1.4	7.9	2.2	9.6	5.2	4.3	5.8	4.6
E	FourPeople	0.3	2.0	0.4	2.8	n/a	n/a	n/a	n/a
	Johnny	0.6	2.2	0.6	3.3	n/a	n/a	n/a	n/a
	KristenAndSara	0.3	2.5	0.1	3.9	n/a	n/a	n/a	n/a
F	BasketballDrillText	1.0	5.5	1.8	7.3	2.2	3.2	2.8	4.1
	ChinaSpeed	3.0	7.1	2.8	9.5	4.2	3.9	4.6	4.6
	SlideEditing	8.5	3.5	11.9	4.9	8.1	2.1	11.0	2.6
	SlideShow	1.5	5.3	2.3	7.2	1.6	2.9	2.4	3.4

presented with a set of test sequences corresponding to the targeted field of application. Furthermore, an example implementation utilizing various proposed search options has also been suggested. The presented results show that a significant computation time reduction can be achieved as a trade-off of a bearable RD performance loss for high resolution sequences, while losses for sequences mainly containing CGI content are significantly higher. For instance, for the LDP configuration, which targets low delay applications and corresponding video conferencing sequences, a BD-rate of only 0.4% is achieved, while performing ~ 3.4 faster than the fast search scheme. Also, for the class B high resolution sequences, a BD-rate of 0.6% with a speed up of ~ 7.6 was obtained. With respect to the RA configuration, losses in coding efficiency are generally higher and the reduction of computation time is lower. Yet, for the high resolution sequences, the best results are obtained, such that 1.5% BD-rate and ~ 3.2 faster performance is achieved for class A sequences, and 1.8% BD-rate and ~ 3.9 faster performance is achieved for class B sequences. In addition, for both configurations, worst results are obtained for CGI sequences (class F). Further, the losses of up to 5.0% are reported for the LDP configuration, and up to 5.2% for the RA configuration.

With respect to future work, it is suggested to use the presented different search options in terms of a model kit for an integer sample search algorithm, which adaptively selects the best strategy provided a certain video sequences, while maintaining certain restrictions. On one hand, especially in the field of applications which require low delay encoding, a strategy may be selected based on the available time, such that in situations were computational load is considerably high, fast strategies can be favored at the cost of coding efficiency; on the other hand, in more relaxed situations, the strategies yielding better coding efficiency can be selected.

REFERENCES

[1] Marpe, D., Schwarz, H., Bosse, S., Bross, B., Helle, P., Hinz, T., Kirchhoffer, H., Lakshman, H., Nguyen, T., Oudin, S., Siekmann, M., Sühring, K., Winken, M., and Wiegand, T., "Video compression using nested quadtree structures, leaf merging, and improved techniques for motion representation and entropy coding," *Circuits and Systems for Video Technology, IEEE Transactions on* **20**(12), 1676–1687 (2010).

[2] Helle, P., Oudin, S., Bross, B., Marpe, D., Bici, M., Ugur, K., Jung, J., Clare, G., and Wiegand, T., "Block Merging for Quadtree-Based Partitioning in HEVC," *Circuits and Systems for Video Technology, IEEE Transactions on* **22**, 1720–1731 (Dec 2012).

[3] Sullivan, G. J. and Wiegand, T., "Rate-distortion optimization for video compression," *Signal Processing Magazine, IEEE* **15**(6), 74–90 (1998).

[4] Sullivan, G. J. and Baker, R., "Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks," in [*Global Telecommunications Conference, 1991. GLOBECOM'91.'Countdown to the New Millennium. Featuring a Mini-Theme on: Personal Communications Services*], 85–90, IEEE (1991).

[5] Hosur, P. I. and Ma, K.-K., "Motion vector field adaptive fast motion estimation," in [*Second International Conference on Information, Communications and Signal Processing (ICICS'99)*], 7–10 (1999).

[6] Laroche, G., Jung, J., and Pesquet-Popescu, B., "RD optimized coding for motion vector predictor selection," *Circuits and Systems for Video Technology, IEEE Transactions on* **18**(9), 1247–1257 (2008).

[7] Schwarz, H., Sullivan, G., Tan, T., and Wiegand, T., "Comparison of the Coding Efficiency of Video Coding Standards–Including High Efficiency Video Coding (HEVC)," *Circuits and Systems for Video Technology, IEEE Transactions on* **22**(12), 1669–1684 (2012).

[8] Bossen, F. and Common, H., "Test conditions and software reference configurations, document JCTVC-L1100," tech. rep., ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) (2013).

[9] Sullivan, G., Ohm, J., Han, W., and Wiegand, T., "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits and Systems for Video Tech* (2012).

[10] Wiegand, T., Sullivan, G., Bjontegaard, G., and Luthra, A., "Overview of the H. 264/AVC video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on* **13**(7), 560–576 (2003).

[11] Grois, D., Marpe, D., Mulayoff, A., Itzhaky, B., and Hadar, O., "Performance comparison of H. 265/MPEG-HEVC, VP9, and H. 264/MPEG-AVC encoders.," in [*PCS*], **13**, 8–11 (2013).

[12] Grois, D., Marpe, D., Nguyen, T., and Hadar, O., "Comparative assessment of H. 265/MPEG-HEVC, VP9, and H. 264/MPEG-AVC encoders for low-delay video applications," in [*SPIE Optical Engineering+ Applications*], 92170Q–92170Q, International Society for Optics and Photonics (2014).

[13] Bross, B., George, V., Alvarez-Mesa, M., Mayer, T., Chi, C. C., Brandenburg, J., Schierl, T., Marpe, D., and Juurlink, B., "HEVC performance and complexity for 4k video," in [*Consumer Electronics Berlin (ICCE-Berlin), 2013. ICCEBerlin 2013. IEEE Third International Conference on*], 44–47, IEEE (2013).

[14] McCann, K., Han, W., and Kim, I., "Samsung's Response to the Call for Proposals on Video Compression Technology, document JCTVC-A124," *Joint Collaborative Team on Video Coding (JCT-VC)* (2010).

[15] Bross, B., Helle, P., Lakshman, H., and Ugur, K., "Inter-picture prediction in HEVC," in [*High Efficiency Video Coding (HEVC)*], Sze, V., Budagavi, M., and Sullivan, G., eds., 113–140, Springer International Publishing, Switzerland (2014).

[16] Koga, T., Iinuma, K., Hirano, A., Iijima, Y., and Ishiguro, T., "Motion compensated interframe coding for video conferencing," in [*Proc. National Telecommunications Conference*], (1981).

[17] Li, R., Zeng, B., and Liou, M. L., "A new three-step search algorithm for block motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on* **4**(4), 438–442 (1994).

[18] Po, L.-M. and Ma, W.-C., "A novel four-step search algorithm for fast block motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on* **6**(3), 313–317 (1996).

[19] Zhu, S. and Ma, K.-K., "A new diamond search algorithm for fast block-matching motion estimation," *Image Processing, IEEE Transactions on* **9**(2), 287–290 (2000).

[20] Jia, H. and Zhang, L., "Directional diamond search pattern for fast block motion estimation," *Electronics Letters* **39**(22), 1581–1583 (2003).

[21] Cheung, C.-H. and Po, L.-M., "Novel cross-diamond-hexagonal search algorithms for fast block motion estimation," *Multimedia, IEEE Transactions on* **7**(1), 16–22 (2005).

[22] Tourapis, A. M., "Enhanced predictive zonal search for single and multiple frame motion estimation," in [*Electronic Imaging 2002*], 1069–1079, International Society for Optics and Photonics (2002).

[23] Tourapis, A. M., Au, O. C., and Liou, M. L., "New results on zonal based motion estimation algorithms-advanced predictive diamond zonal search," in [*Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*], **5**, 183–186, IEEE (2001).

[24] Tourapis, A. M., Au, O. C., and Liou, M. L., "Fast motion estimation using circular zonal search," in [*Electronic Imaging'99*], 1496–1504, International Society for Optics and Photonics (1998).

[25] Tourapis, A. M., Au, O. C., and Liou, M. L., "Predictive motion vector field adaptive search technique (PMVFAST): enhancing block-based motion estimation," in [*Photonics West 2001-Electronic Imaging*], 883–892, International Society for Optics and Photonics (2000).

[26] Tourapis, A. M., Au, O. C., Liou, M. L., Shen, G., and Ahmad, I., "Optimizing the MPEG-4 encoder-advanced diamond zonal search," in [*Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*], **3**, 674–677, IEEE (2000).

[27] So, H., Kim, J., Cho, W.-K., and Kim, Y.-S., "Fast motion estimation using modified diamond search patterns," *Electronics Letters* **41**(2), 62–63 (2005).

[28] Nie, Y. and Ma, K.-K., "Adaptive rood pattern search for fast block-matching motion estimation," *Image Processing, IEEE Transactions on* **11**(12), 1442–1449 (2002).

[29] Tourapis, A. M., Au, O. C., and Liou, M. L., "Highly efficient predictive zonal algorithms for fast block-matching motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on* **12**(10), 934–947 (2002).

[30] Li, X., Li, E. Q., and Chen, Y.-K., "Fast multi-frame motion estimation algorithm with adaptive search strategies in H. 264," in [*Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*], **3**, iii–369, IEEE (2004).

[31] Sarwer, M. G. and Wu, Q. J., "Adaptive variable block-size early motion estimation termination algorithm for H. 264/AVC video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on* **19**(8), 1196–1201 (2009).

[32] Chen, H.-M., Chen, P.-H., Lin, C.-T., and Liu, C.-C., "Content-adaptive thresholding early termination scheme on directional gradient descent searches for fast block motion estimation," *Optical Engineering* **51**(11), 117401–117401 (2012).

[33] Lee, H., Jung, B., Jung, J., and Jeon, B., "Fast subpel motion estimation for H.264/advanced video coding with an adaptive motion vector accuracy decision," *Optical Engineering* **51**(11), 117001–117001 (2012).

[34] Xu, J., Chen, Z., and He, Y., "Efficient fast ME predictions and early-termination strategy based on H. 264 statistical characters," in [*Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*], **1**, 218–222, IEEE (2003).

[35] Ismail, Y., Shaaban, M., and Bayoumi, M., "An adaptive block size phase correlation motion estimation using adaptive early search termination technique," in [*Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*], 3423–3426, IEEE (2007).

[36] Chen, L.-F., Yang, S.-P., and Lai, Y.-K., "Model-based early termination scheme for H. 264/AVC inter prediction," in [*Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*], 597–600, IEEE (2009).

[37] Purnachand, N., Alves, L. N., and Navarro, A., "Fast Motion Estimation Algorithm for HEVC," in [*Consumer Electronics-Berlin (ICCE-Berlin), 2012 IEEE International Conference on*], 34–37, IEEE (2012).

[38] "HEVC software repository (main at HHI)." `https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-12.0/`.

[39] Bossen, F., Bross, B., Sühring, K., and Flynn, D., "HEVC Complexity and Implementation Analysis," *IEEE Transactions on Circuits and Systems for Video Technology* **22**, 1669–1684 (Dec. 2012).

[40] McCann, K., Bross, B., Han, W., Kim, I., Sugimoto, K., and Sullivan, G., "High Efficiency Video Coding (HEVC) Test Model 12 (HM12) Encoder Description, document JCTVC-N1002,"

[41] Bjontegard, G., "Calculation of average PSNR differences between RD-curves," *ITU-T VCEG-M33* (2001).