

Context-Based Fractional Sample Refinement for HEVC Compliant Encoding

Georg Maier, Benjamin Bross, *Student Member, IEEE*, Dan Grois, *Senior Member, IEEE*,
Detlev Marpe, *Fellow, IEEE*, Heiko Schwarz, Remco C. Veltkamp, and
Thomas Wiegand, *Fellow, IEEE*

Abstract—The H.265/MPEG-H High Efficiency Video Coding compliant encoding process faces the challenge of high computational complexity. Particularly, in the case of inter-picture prediction, most of the computational resources are allocated for the motion estimation (ME) process. In turn, ME and motion compensation enable improving coding efficiency by addressing the blocks of video frames as corresponding displacements from one or more reference blocks. These displacements do not necessarily have to be limited to integer sample positions, but may have an accuracy of half sample or quarter sample positions, which are identified during fractional sample refinement. In this paper, a context-based scheme for fractional sample refinement is proposed. The scheme takes the advantage of already obtained information in prior ME steps and provides significant flexibility in terms of parameterization. In this way, it adaptively achieves a desired tradeoff between computational complexity and coding efficiency. According to the experimental results obtained for an example algorithm utilizing the proposed framework, a significant decrease in the number of search points can be achieved. For instance, considering only 6 instead of 16 fractional sample positions results in a tradeoff of only 0.4% Bjøntegaard-Delta-rate loss for high-definition video sequences compared with the conventional *interpolation-and-search* method.

Index Terms—Fractional sample refinement, H.265/MPEG-H High Efficiency Video Coding (HEVC), motion estimation (ME).

I. INTRODUCTION

THE H.265/MPEG-H High Efficiency Video Coding (HEVC) standard allows bit-rate savings of about 50% for essentially the same subjective quality compared with its predecessor, the H.264/MPEG-4 Advanced Video Coding (AVC) standard. Thereby, it efficiently tackles the challenges posed on modern communication networks and storage media resulting from the dramatically increasing video bandwidth demands.

Manuscript received April 1, 2016; revised June 28, 2016 and August 15, 2016; accepted September 15, 2016. Date of publication September 27, 2016; date of current version February 13, 2018. This paper was recommended by Associate Editor L. Zhou.

G. Maier is with the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (Fraunhofer IOSB), 76131 Karlsruhe, Germany (e-mail: georg.maier@iosb.fraunhofer.de).

B. Bross, D. Grois, D. Marpe, H. Schwarz, and T. Wiegand are with the Video Coding and Analytics Department, Fraunhofer Institute for Telecommunications–Heinrich Hertz Institute, 10587 Berlin, Germany (e-mail: benjamin.bross@hhi.fraunhofer.de; dan.grois@hhi.fraunhofer.de; detlev.marpe@hhi.fraunhofer.de; heiko.schwarz@hhi.fraunhofer.de; thomas.wiegand@hhi.fraunhofer.de).

R. C. Veltkamp is with the Department of Information and Computing Sciences, Utrecht University, 3584 CC Utrecht, The Netherlands (e-mail: r.c.veltkamp@uu.nl).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2016.2613910

The H.265/MPEG-H HEVC standard was designed to be applicable for almost all existing H.264/MPEG-4 AVC applications. Also, special emphasis is put on high-definition (HD) and ultra-HD (UHD) video content, since need for these formats is expected to increase significantly during the next years. However, the abovementioned coding performance gain comes at the cost of tremendously increased computational complexity, mainly due to supporting a relatively large number of coding modes [1], [2].

During an HEVC encoding process, the mode decision process determines whether a coding unit (CU) should be encoded by using the intra- or inter-picture prediction techniques in addition to determining the quadtree block partitioning [3], [4]. Thereby, either spatial or temporal redundancies are exploited. The decision process is commonly implemented by means of a rate-distortion optimization technique. A cost function, typically denoted as $J = D + \lambda R$, has to be minimized, whereas the overall cost J is based on the bit-rate cost R and on the distortion cost D , which are weighted by using a Lagrange multiplier λ [5]. Therefore, in order to determine the best coding mode, the CU is usually encoded in a plurality of coding modes, which leads to a high computational burden at the encoder end.

Inter-picture prediction plays a crucial role in modern video coding applications due to its high potential to significantly improve the coding efficiency. Temporal redundancy is removed by encoding a block in terms of a displacement to one or more reference blocks, which are located in prior encoded reference frames. The displacement information is encoded as a so-called *motion vector* (MV), which is identified by executing the motion estimation (ME) process. Generally, in recent video encoders, ME is usually regarded as a three-step process, including the *MV prediction*, *integer sample accuracy search*, and *fractional sample refinement*. Regarding the latter, the HEVC standard allows to address the motion information on a quarter sample precision level. Such fractional sample positions may be obtained by applying computationally costly interpolation methods, followed by a search around the position of the prior-determined integer sample MV. Many fast integer sample accuracy search algorithms have been presented in the past, which significantly reduce the encoding time at the cost of a reasonable small coding efficiency loss. However, when the time spent for integer motion search is reduced, the contribution of fractional motion search to the overall motion search time becomes more significant. In the HEVC reference software (HM), which already applies fast integer motion

search, interpolation makes up for approximately 19.5% of the encoding time [1]. Hence, fast fractional motion searches are of particular interest when fast integer motion searches are applied.

In this paper, a context-based software framework for fractional sample refinement is proposed. Given a certain *context*, conditional probabilities are derived. These probabilities indicate the likelihood that specific fractional sample positions yield a coding efficiency gain compared with the prior-selected integer sample position. More precisely, a context is defined in terms of a function that evaluates neighboring integer samples to determine the most promising fractional sample positions. To demonstrate the success of the proposed method, a specific context is presented, for which a good tradeoff in terms of computational complexity and coding efficiency is reported. While the algorithm itself has been implemented in an HEVC software encoder, suitable hardware implementations could be an interesting topic for future work.

This paper is organized as follows. The background along with an overview of related work is provided in Section II. Following that, a detailed description of the proposed, generic context-based fractional sample refinement framework is given in Section III. In Section IV, an example algorithm based on the proposed framework is introduced. The test methodology and experimental results for this specific context are discussed in detail in Section V. Finally, this paper is concluded in Section VI, further providing a brief overview of future research perspectives.

II. BACKGROUND AND RELATED WORK

A comprehensive overview of the HEVC standard is provided in [6] and a detailed analysis of the coding efficiency can be found in [7]. In addition, a comparison to other recent video coding schemes is presented in [8] with a particular focus on low-delay applications in [9]. Also, additional studies with regard to the HEVC decoding performance and complexity are shown in [10], while making a special emphasis on 4K-resolution videos.

Furthermore, with regard to video coding standards that employ inter-picture prediction, intensive research has been conducted for carrying out suboptimal strategies in the field of integer sample accuracy ME. Those strategies typically target achieving a tradeoff between computational complexity and coding efficiency. A recent survey on fast block matching algorithms is provided in [11]. Furthermore, the application of pattern-based approaches in the context of HEVC is studied in [12].

In the following, several traditional search techniques regarding fractional sample ME are discussed. Section II-A overviews the conventional *interpolation-and-search* method, Section II-B discusses the recent pattern-based approaches, and Section II-C reviews several error surface approximation techniques. Although this paper and related work reviewed in the following mainly focus on the software implementations of fast fractional sample ME schemes, a lot of research has also been conducted regarding efficient hardware designs. In [13], such an implementation for the interpolation task is presented.

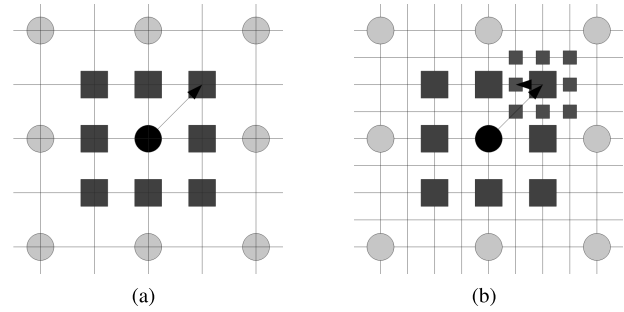


Fig. 1. Example of the conventional interpolation-and-search method. Circles: integer sample positions. Larger squares: half sample positions. Smaller squares: quarter sample positions.

A hardware design including both interpolation and search is presented in [14]. The proposed approach also includes a module responsible for integer sample ME. Also, in [15] a design for performing bilinear quarter pixel approximation and a search pattern based on it is presented.

A. Interpolation-and-Search Method

The traditional so-called *interpolation-and-search* method for fractional sample refinement is presented in Fig. 1. It relies on the common assumption that an optimal fractional sample position is located adjacent to the optimal integer sample position. This assumption in turn gives rise to 48 possible quarter sample positions. However, the corresponding search is typically divided into two steps as follows. First, the interpolation and search is performed on a half sample accuracy level, as shown in Fig. 1(a). Second, this procedure is repeated on a quarter sample accuracy level, while reducing the search space to the neighborhood of the previously selected half sample position, as further shown in Fig. 1(b). Consequently, this approach is limited to eight half sample positions and eight quarter sample positions. Also, if no better fractional sample position can be determined, the MV stays on an integer sample position.

B. Pattern-Based Approaches

In light of the popularity of approaches that have been developed for integer sample search, various pattern-based strategies have also been developed with regard to fractional sample refinement. However, while adopting the assumption that the optimal fractional sample position is always adjacent to the selected integer sample position, it is important to note that the search space has rather a limited size. Additional attempts have been made to subsample this search space by first trying to predict a restricted search space and then to conduct a search within the predicted fractional sample space. For example, Zhang *et al.* [16] propose to reduce the number of search points from eight to six points both for the half sample and the quarter sample search. This is done by first checking only four “near-located” samples and subsequently two “far-located” samples, which in turn are positioned next to the best “near-located” sample. They treat samples that are adjacently located in the horizontal and vertical directions as “near-located” samples and those adjacently located in

the diagonal directions as “far-located” samples. As a result, this approach clearly favors the “near-located” samples over the “far-located” samples by providing them a larger weight during the decision process. Another example is shown in [17], where the authors propose to limit the search to one quadrant. The corresponding quadrant is determined by checking two fixed samples. In addition, in [18], the search space is restricted to only a few points depending on the direction of the integer sample accuracy MV. Furthermore, in [19], pattern application dependent on the distortion distribution of surrounding integer sample positions is proposed.

C. Error Surface Approximation Techniques

Another type of techniques related to fractional sample refinement attempts to approximate the error surface on a fractional sample level around the selected integer sample position. Suh *et al.* [20] propose three models for determining a fractional sample accuracy MV. This approach requires the computation of nine integer sample position errors. Depending on the applied integer sample search algorithm, some of those can be already known. Interpolation and search on a fractional sample level is omitted. While they present results and a half sample accuracy level, their approach can easily be extended to quarter sample accuracy. In addition, in [21], those models are applied by considering the obtained coding costs for the required integer sample positions rather than the distortions values. The approach also includes an early termination criterion. It is applied whenever, based on the shape of the resulting surface, no significant improvement through the fractional sample refinement is expected.

III. PROPOSED CONTEXT-BASED FRACTIONAL SAMPLE REFINEMENT FRAMEWORK

While the context-based approaches have been successfully applied in video compression, e.g., for entropy coding [22], neither applications in integer sample nor in fractional sample ME have been reported. In this section, a general software framework for context-based fractional sample refinement is established. The proposed approach consists of an offline as well as an online phase, as described in detail in Sections III-A and III-B, respectively. Briefly, the goal of the offline phase is to collect data for each possible fractional sample position. Based on this data, an average increase in encoding efficiency over the prior chosen integer sample position given a certain context can be calculated. During the actual encoding process, i.e., the online phase, those average gains are used to steer the fractional sample refinement, independent of the given sequence. In this section, the context is kept generic. A specific context, for which reasonable results in terms of the tradeoff between computational complexity reduction and coding efficiency loss are obtained, is introduced for an example algorithm in Section IV.

A. Offline Phase

During the offline phase, the goal is to order the fractional sample positions according to their average performance gain

in terms of coding efficiency. Let \vec{m} be the MV obtained from an arbitrary integer sample accuracy search algorithm and let \mathcal{S} be a set of allowed fractional sample displacements \vec{s} surrounding \vec{m} . Furthermore, let f be a cost function, calculating the cost of an MV, for instance, where $f(\vec{a}) > f(\vec{b})$ means that \vec{b} is to be preferred over \vec{a} . Also, the performance gain $g(\vec{s})$ that \vec{s} provides over \vec{m} is supposed to be defined according to

$$g(\vec{s}) = f(\vec{m}) - f(\vec{s}). \quad (1)$$

In addition, an arbitrary context $ctx(\mathcal{C})$ is assumed with \mathcal{C} being the input information and ctx being a function that processes this input information. In order to calculate the average gain, sample data need to be collected during the offline phase. Let \mathcal{N} be a set of N samples where each sample consists of the context input C_n and the gains $g_n(\vec{s})$ for each fractional sample displacement $\vec{s} \in \mathcal{S}$ with $n = 1 \dots N$. Then, for each $\vec{s} \in \mathcal{S}$ given a specific context $ctx(\mathcal{C})$, the average performance gain $\bar{g}(\vec{s}|ctx(\mathcal{C}))$ resulting from selecting \vec{s} over \vec{m} by using the set of samples \mathcal{N} is calculated as shown in

$$\bar{g}(\vec{s}|ctx(\mathcal{C})) = \frac{\sum_{n \in \mathcal{N}_{ctx}} g_n(\vec{s})}{|\mathcal{N}_{ctx}|}. \quad (2)$$

Typical example sequences can be used to obtain the sample set \mathcal{N} , in the remainder referred to as *training set*. As a result, the fractional sample positions can be ranked from the one yielding the highest to the one yielding the lowest average performance gain. Let \mathcal{S}_{rank} be the set of $rank - 1$ best fractional sample positions. Then, the recursive formula provided in (3) can be used to generate a lookup table (LUT) containing a fractional position $\vec{s}_{rank,ctx(\mathcal{C})}$ for a given $rank$ and context $ctx(\mathcal{C})$ as¹

$$\vec{s}_{rank,ctx(\mathcal{C})} = \arg \max_{\vec{s} \in \mathcal{S} \setminus \mathcal{S}_{rank}} \bar{g}(\vec{s}|ctx(\mathcal{C}))$$

$$\mathcal{S}_{rank} = \begin{cases} \emptyset & \text{rank} = 1 \\ \{\vec{s}_{i,ctx(\mathcal{C})} | i = 1 \dots rank - 1\} & \text{rank} > 1. \end{cases} \quad (3)$$

The effectiveness of the selected context model can be determined by applying the results on an alternative set of sequences. Also, different frames thereof can be used. Two metrics may be used for evaluation. One is the overall success rate, i.e., the relative frequency of finding optimal fractional sample positions. The second is the generated encoding cost overhead, i.e., the overhead of the encoding cost compared with the corresponding optimal position. It should be noted that a low overall success rate may still correspond to a well-performing context model in the case of generally selecting a near-optimal position. This is the case whenever the position chosen yields an encoding cost sufficiently close to the optimum. Obviously, provided a large training set (i.e., a large set \mathcal{N}), the described offline phase may require exhaustive simulations. As already mentioned in the introduction, the estimated LUT is not changed anymore and further used to steer the fractional sample refinement in the final algorithm as explained in Section III-B. Hence, simulations only need to be performed once to obtain the LUT when designing an algorithm.

¹If more than one $\vec{s} \in \mathcal{S} \setminus \mathcal{S}_{rank}$ minimize $\bar{g}(\vec{s}|ctx(\mathcal{C}))$, we let $\vec{s}_{rank,ctx(\mathcal{C})}$ be the first \vec{s} amongst them.

Algorithm 1 Context-Based Fractional Sample Search

```

bestMv  $\leftarrow \vec{m}$ 
bestCost  $\leftarrow f(\vec{m})$ 
rank  $\leftarrow 1$ 

while crit holds do
   $\vec{s}_{rank,ctx(C)} \leftarrow$  MV with rank rank given ctx(C) from the
  LUT calculated during offline phase based on the highest
  average gain.
  If necessary, perform up-sampling for  $\vec{s}_{rank,ctx(C)}$ 
  newCost  $\leftarrow f(\vec{s}_{rank,ctx(C)})$ 
  if newCost < bestCost then
    bestCost  $\leftarrow$  newCost
    bestMV  $\leftarrow \vec{s}_{rank,ctx(C)}$ 
  end if
  rank  $\leftarrow$  rank + 1
end while

return bestMv

```

B. Online Phase

During the encoding process, the information obtained from the offline phase is applied by using the algorithm skeleton shown in Algorithm 1. Here, the cost of the fractional sample position $\vec{s}_{1,ctx(C)}$ having the highest average gain (rank = 1) given a context *ctx*(*C*) is compared with the costs of the current best sample position. In this regard, this corresponds to the chosen integer sample position \vec{m} . Obviously, the calculation of the context may result in an overhead as additional information needs to be derived. However, compared with complexity caused by the interpolation methods, this overhead is negligible. According to [1], for a block of size $N \times N$, $8 + (56/N)$ 8-b and eight 16-b multiply accumulate operations are required per sample only for the luma component in software. In comparison, context calculation requires only 8×3 multiplications of integer distortions with the corresponding weights and two additions, provided that integer distortions have already been calculated during integer ME. The criterion *crit* decides whether the online search algorithm should proceed with the next fractional sample position from the ranking LUT determined during the offline phase (rank = 2, 3, ...). While the approach is fairly simple, its advantage lies in its flexibility in designing the termination criterion. An example criterion is presented in Section IV.

IV. EXAMPLE FRACTIONAL SAMPLE REFINEMENT BASED ON THE PROPOSED FRAMEWORK

In this section, a fractional sample MV search algorithm is introduced as an example, which is built upon the generic framework illustrated in Section III. The fractional sample MV search was integrated on top of the fast integer sample MV search in the HEVC test model (HM) reference software encoder [25]. A specific context, which has to be evaluated in both the offline and the online phase, is derived first in Section IV-A. Then, the proposed framework is used to determine the best half sample position as described

\vec{x}_1	\vec{x}_2	\vec{x}_3
\vec{x}_4	\vec{m}	\vec{x}_5
\vec{x}_6	\vec{x}_7	\vec{x}_8

Fig. 2. Notation for the positions. (a) Half sample positions (b) Quarter sample positions.

in Section IV-B. Around the best half sample MV, the quarter sample MV search is performed as a second implementation of the framework and explained in detail in Section IV-C. This hierarchical two-level approach adopts the assumption that the optimal quarter sample position is adjacent to the optimal half sample position as has been discussed in the context of the conventional *interpolation-and-search* method, see Section II-A. It is noted that all the training data in this section was obtained for the first ten frames for each of the common test condition [24] sequences presented in Table III. In Section IV-D, an example search step illustrates the combination of half and quarter sample refinement.

A. Studied Context

In the course of this paper, it was observed that the optimal fractional sample positions are evenly distributed. Therefore, no assumptions regarding the general priorities of positions can be made. However, several studies, as those presented for example in [19] and [20], report a correlation between distortions obtained at integer sample positions adjacent to the chosen position and the best fractional sample position. In order to define the integer sample positions for the context evaluation input function, let $\vec{x}_1, \dots, \vec{x}_8$ be the integer positions surrounding \vec{m} from the top-left to bottom-right, as shown in Fig. 2. In HM, the sum of absolute differences (SAD) is used as a distortion metric for integer sample MVs. Hence, the distortion value for position \vec{x}_i is given by $d_{SAD}(\vec{x}_i)$. The context input *C* is defined in (4) as the vector containing the SAD distortion values for all adjacent integer sample positions

$$C = \vec{d}_{SAD} = \begin{pmatrix} d_{SAD}(\vec{x}_1) \\ d_{SAD}(\vec{x}_2) \\ d_{SAD}(\vec{x}_3) \\ \dots \\ d_{SAD}(\vec{x}_8) \end{pmatrix}. \quad (4)$$

Let M_i be the *i*th row of an 8×8 weighting matrix *M*, and then the calculation of context *ctx*(\vec{d}_{SAD}) is given by²

$$ctx(\vec{d}_{SAD}) = \arg \min_{i=1 \dots 8} (M_i \cdot \vec{d}_{SAD}). \quad (5)$$

Hence, *ctx*(\vec{d}_{SAD}) identifies the minimal weighted linear combination of the directly neighboring integer sample distortions \vec{d}_{SAD} as defined in (4). The intention of this approach is to utilize information obtained on the integer sample accuracy level. This choice is also in line with the results presented in [20] and [21]. From these works, it can be

²If more than one $i = 1 \dots 8$ minimize $M_i \cdot \vec{d}_{SAD}$, we let *ctx*(\vec{d}_{SAD}) be the smallest *i* amongst them.

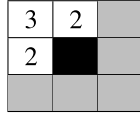


Fig. 3. Visual example of the weights and integer sample positions involved in the linear combination when multiplying the first row of the weighting matrix M with the distortion vector \vec{d} .

concluded that the distortions obtained from integer sample positions surrounding a candidate fraction sample position provide diagnostically conclusive information regarding the likeliness of it yielding an encoding efficiency gain. Based on this, numerous simulations to test different combinations of surrounding integer sample positions and their associated distortions have been carried out. Those included evaluation based on different weights for the surrounding integer sample distortions. Here, a weighting for which promising results have been obtained is presented in (6) and used in the following. However, optimization strategies, such as machine learning approaches, for instance, have not been carried out. Therefore, it is expected that even better performing weights can be found.

$$M = \begin{bmatrix} 3 & 2 & 0 & 2 & 0 & 0 & 0 & 0 \\ 2 & 3 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 3 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 3 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 & 3 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 3 & 2 \\ 0 & 0 & 0 & 0 & 2 & 0 & 2 & 3 \end{bmatrix}. \quad (6)$$

The notation is further schematically shown in Fig. 3 by providing an example for the multiplication of the first row of M with the vector \vec{d}_{SAD} . The black square position corresponds to the center of the search and the gray cells correspond to positions, where the distortion values are not considered. Scalars indicated within the white cells represent the distortion weights of each corresponding position, thereby resulting in the following overall distortion calculation as given in (7) for the example

$$(M_1 \cdot \vec{d}_{\text{SAD}}) = 3d_{\text{SAD}}(\vec{x}_1) + 2d_{\text{SAD}}(\vec{x}_2) + 2d_{\text{SAD}}(\vec{x}_4). \quad (7)$$

B. Half Sample MV Refinement

Given the hierarchical two-level approach, the half sample MV refinement needs to be performed before subsequent quarter sample refinement. Hence, in the following, the complete half sample MV refinement is described in detail, including the training set, the corresponding offline LUT derivation, and the online search algorithm.

1) *Training Set:* As already defined in Section III-A, each sample from the training set \mathcal{N} includes the context input C_n and the gains $g_n(\vec{s})$ for the considered fractional samples positions \vec{s} . Here, the context input C_n equals the SAD values $\vec{d}_{\text{SAD},n}$ of the integer MVs adjacent to the selected integer MV \vec{m}_n . The fractional sample positions \vec{s} considered here are all eight half sample positions \vec{h}_i ($i = 1 \dots 8$) surrounding \vec{m} . It should be noted that HM uses the sum of Hadamard transformed differences as a distortion measure for all fractional

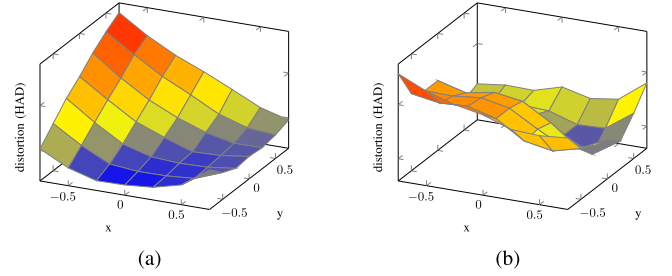


Fig. 4. Examples of the quarter sample distortion maps, where (0,0) position corresponds to the selected integer sample MV. Preferably none! (a) Quarter sample distortion map (b) Quarter sample distortion map.

TABLE I
HALF SAMPLE POSITIONS $\vec{h}_{rank,ctx(\vec{d}_{\text{SAD}})}$ ORDERED BY LIKELINESS TO YIELD AN ENCODING PERFORMANCE GAIN OVER INTEGER SAMPLE MV (*Rank*) GIVEN A CONTEXT $ctx(\vec{d}_{\text{SAD}})$

$ctx(\vec{d}_{\text{SAD}}) \setminus rank$	1	2	3	4	5	6	7	8
1	\vec{h}_1	\vec{h}_2	\vec{h}_4	\vec{h}_3	\vec{h}_6	\vec{h}_5	\vec{h}_7	\vec{h}_8
2	\vec{h}_2	\vec{h}_3	\vec{h}_1	\vec{h}_5	\vec{h}_4	\vec{h}_6	\vec{h}_8	\vec{h}_7
3	\vec{h}_3	\vec{h}_2	\vec{h}_5	\vec{h}_1	\vec{h}_8	\vec{h}_4	\vec{h}_7	\vec{h}_6
4	\vec{h}_4	\vec{h}_1	\vec{h}_6	\vec{h}_7	\vec{h}_2	\vec{h}_8	\vec{h}_5	\vec{h}_3
5	\vec{h}_5	\vec{h}_7	\vec{h}_8	\vec{h}_6	\vec{h}_4	\vec{h}_3	\vec{h}_1	\vec{h}_2
6	\vec{h}_6	\vec{h}_7	\vec{h}_4	\vec{h}_8	\vec{h}_1	\vec{h}_5	\vec{h}_2	\vec{h}_3
7	\vec{h}_7	\vec{h}_8	\vec{h}_6	\vec{h}_4	\vec{h}_5	\vec{h}_1	\vec{h}_3	\vec{h}_2
8	\vec{h}_8	\vec{h}_5	\vec{h}_7	\vec{h}_2	\vec{h}_3	\vec{h}_1	\vec{h}_4	\vec{h}_6

sample positions. Consequently, the calculation of the gain is based on HAD distortion values for both the half sample and the integer sample positions as shown in (8). Two examples of such HAD fractional sample distortion maps are provided in Fig. 4.

$$g_n(\vec{h}_i) = d_{\text{HAD}}(\vec{m}) - d_{\text{HAD}}(\vec{h}_i). \quad (8)$$

2) *Offline Phase:* Using the samples from the training set \mathcal{N} , average performance gains given the context $ctx(\vec{d}_{\text{SAD}})$ from (5) are obtained as illustrated in

$$\mathcal{N}_{ctx} = \{n | ctx(\vec{d}_{\text{SAD},n}) = ctx(\vec{d}_{\text{SAD}})\} \quad (9)$$

$$\bar{g}(\vec{h}_i | ctx(\vec{d}_{\text{SAD}})) = \frac{\sum_{n \in \mathcal{N}_{ctx}} g_n(\vec{h}_i)}{|\mathcal{N}_{ctx}|}. \quad (10)$$

The result of (9) can be used to generate an LUT as described in (3). Table I shows the resulting LUT containing the half sample positions $\vec{h}_{rank,ctx(\vec{d}_{\text{SAD}})}$, ordered by *rank* for each possible context $ctx(\vec{d}_{\text{SAD}})$ from (5). Here, again a rank of 1 denotes the highest and a rank of 8 denotes the lowest average performance gain. One interesting observation is that the best-ranked half sample position \vec{h}_i for each context always lies in the middle between the current integer sample position \vec{m} and the adjacent integer sample position \vec{x}_i that influences mainly the corresponding context. Taking $\vec{h}_{1,1} = \vec{h}_1$ as an example, the context $ctx(\vec{d}_{\text{SAD}}) = 1$ means that the linear combination of the distortions $M_1 \cdot \vec{d}_{\text{SAD}}$ minimizes (5). From (7), it can be seen that this linear combination is centered around the top-left neighboring integer sample position \vec{x}_1 that also has the highest weight. This illustrates the correlation between the neighboring integer sample distortions and the

Algorithm 2 Online Phase Implementation for Half Sample Search

```

bestMv ←  $\vec{m}$ 
bestCost ←  $d_{\text{HAD}}(\vec{m})$ 
rank ← 1

while rank ≤ u do
     $\vec{h}_{\text{rank}, \text{ctx}(\vec{d}_{\text{SAD}})}$  ← MV from Table I
    If necessary, perform up-sampling for  $\vec{h}_{\text{rank}, \text{ctx}(\vec{d}_{\text{SAD}})}$ 
    newCost ←  $d_{\text{HAD}}(\vec{h}_{\text{rank}, \text{ctx}(\vec{d}_{\text{SAD}})})$ 
    if newCost < bestCost then
        bestCost ← newCost
        bestMV ←  $\vec{h}_{\text{rank}, \text{ctx}(\vec{d}_{\text{SAD}})}$ 
    end if
    rank ← rank + 1
end while

return bestMv
    
```

best fractional sample position. The precalculated LUT serves as the basis for any sequence to be encoded using the algorithm described in the online phase.

In order to obtain this information for the quarter sample accuracy level, the online phase needs to be simulated to obtain its result in the proposed algorithm. Hence, before proceeding to the quarter sample accuracy level, the online phase for the half sample accuracy level is described.

3) *Online Phase*: The online phase is implemented according to Algorithm 1 and context calculation is performed as illustrated in (5). A detailed description of the algorithm is provided in Algorithm 2. It can be seen that the cost function f is replaced by the HAD distortion measure d_{HAD} used for fractional sample positions. For the search termination criterion crit discussed in Section III-B, a fairly simple implementation is considered. An upper bound u of ranks to be searched is used. It is possible to either use a fixed u to limit the number of fractional sample position checks, or the encoder may choose a value adaptively. Clearly, a reasonable choice depends on the application of a corresponding encoder. For instance, in times of high computational load and tight real-time requirements, an encoder may favor low values of u over large ones, while in loadwise more relaxed situations the preference would be vice versa. However, such an adaptive mechanism would require a way of monitoring the systems state, which is not part of the work presented here. In Section V-A, results for various fixed values of u are presented.

C. Quarter Sample MV Refinement

After having discussed the implementation for half sample MV refinement, the following provides a detailed description of the corresponding components, i.e., training set, offline phase, and online phase, for the quarter sample MV refinement. As has been mentioned, the algorithm follows a hierarchical structure; hence, quarter sample refinement is performed dependent on the result of the half sample refinement.

TABLE II

QUARTER SAMPLE POSITIONS $\vec{q}_{\text{rank}, \vec{h}}$ ORDERED BY LIKELINESS TO YIELD AN ENCODING PERFORMANCE GAIN OVER THE INTEGER SAMPLE MV (*Rank*) WHEN \vec{h} IS SELECTED BY PRIOR HALF SAMPLE REFINEMENT GIVEN THE CONTEXT $\text{ctx}(\vec{d}_{\text{SAD}}) = 1$

$\vec{h} \setminus \text{rank}$	1	2	3	4	5	6	7	8
\vec{h}_1	$\vec{q}_{8,1}$	$\vec{q}_{7,1}$	$\vec{q}_{5,1}$	$\vec{q}_{4,1}$	$\vec{q}_{6,1}$	$\vec{q}_{2,1}$	$\vec{q}_{1,1}$	$\vec{q}_{3,1}$
\vec{h}_2	$\vec{q}_{6,2}$	$\vec{q}_{7,2}$	$\vec{q}_{4,2}$	$\vec{q}_{8,2}$	$\vec{q}_{1,2}$	$\vec{q}_{2,2}$	$\vec{q}_{5,2}$	$\vec{q}_{3,2}$
\vec{h}_4	$\vec{q}_{2,3}$	$\vec{q}_{3,3}$	$\vec{q}_{5,3}$	$\vec{q}_{1,3}$	$\vec{q}_{8,3}$	$\vec{q}_{4,3}$	$\vec{q}_{7,3}$	$\vec{q}_{6,3}$
\vec{m}	$\vec{q}_{1,4}$	$\vec{q}_{2,4}$	$\vec{q}_{4,4}$	$\vec{q}_{3,4}$	$\vec{q}_{5,4}$	$\vec{q}_{7,4}$	$\vec{q}_{6,4}$	$\vec{q}_{8,4}$

1) *Training Set*: After half sample refinement has been performed, $\text{ctx}(\vec{d}_{\text{SAD},n})$ is known for each sample in the training set \mathcal{N} . With respect to quarter sample refinement, each sample is now characterized by four sets of eight quarter sample performance gains $g_n(\vec{q}_{i,j})$ with i ranging from 1 to 8 and j ranging from 1 to 4. Due to the hierarchical nature of the algorithm, the first three sets include the performance gains for the eight quarter sample positions $\vec{q}_{i,\text{rank}}$ adjacent to the three best-ranked half sample positions given the prior determined context $\text{ctx}(\vec{d}_{\text{SAD},n})$, i.e., $\vec{h}_{\text{rank}, \text{ctx}(\vec{d}_{\text{SAD},n})}$ with rank ranging from 1 to 3. Using the three best-ranked half sample positions is motivated by the fact that only up to three best half sample position are checked during the online phase of the half sample MV refinement. This corresponds to limiting u to 3 in Algorithm 2. Independent of $\text{ctx}(\vec{d}_{\text{SAD},n})$, the fourth set includes the performance gains for the eight quarter sample positions $\vec{q}_{i,4}$ adjacent to the integer sample position \vec{m} resulting from the prior integer sample MV search.

2) *Offline Phase*: The average performance gain for quarter sample positions is obtained by using the information from the prior half sample step described in Section IV-B. More precisely, the average performance gain for each quarter sample position is calculated on the basis of every possible prior-selected half sample position as shown in

$$\begin{aligned}
 \mathcal{N}_{\text{ctx}} &= \{n | \text{ctx}(\vec{d}_{\text{SAD},n}) = \text{ctx}(\vec{d}_{\text{SAD}})\} \\
 \bar{g}(\vec{q}_{i,j} | \text{ctx}(\vec{d}_{\text{SAD}})) &= \frac{\sum_{n \in \mathcal{N}_{\text{ctx}}} g_n(\vec{q}_{i,j})}{|\mathcal{N}_{\text{ctx}}|}. \quad (11)
 \end{aligned}$$

Consequently, for each training sample $n \in N$, half sample MV refinement as described in Section IV-B is simulated dependent on the calculated context $\text{ctx}(\vec{d}_{\text{SAD},n})$. In comparison to the result of the offline phase for half sample accuracy as illustrated in Table I, this results in one table for each of the eight possible outcomes of $\text{ctx}(\vec{d}_{\text{SAD}})$. As already mentioned in the description of the current training set, the half sample MV refinement may lead to four possible half sample positions for each context. This includes the three best-ranked half sample positions and the integer sample position in the case no better half sample position is found. Assuming that the context was calculated to be equal to 1, the resulting table for quarter sample accuracy is provided in Table II.

3) *Online Phase*: The implementation of the refinement on a quarter sample accuracy level is rather similar to the one for half sample accuracy as described in Section IV-B3. However, instead of only checking positions with increasing ranks from

Algorithm 3 Online Phase Implementation for Quarter Sample Search

 $\vec{h} \leftarrow \text{bestMV}$ from Algorithm 2
 $\text{bestMv} \leftarrow \vec{h}$
 $\text{bestCost} \leftarrow d_{\text{HAD}}(\text{bestMv})$
 $u \leftarrow 3$
 $\text{rank} \leftarrow 1$
while $\text{rank} - u \leq 0$ **do**
 $\vec{q}_{\text{rank}, \vec{h}} \leftarrow \text{MV}$ from according LUT table dependent on $\text{ctx}(d_{\text{SAD}})$ (e.g. Table II for $\text{ctx}(d_{\text{SAD}}) = 1$)
 If necessary, perform up-sampling for $\vec{q}_{\text{rank}, \vec{h}}$
 $\text{newCost} \leftarrow d_{\text{HAD}}(\vec{q}_{\text{rank}, \vec{h}})$
if $\text{newCost} < \text{bestCost}$ **then**
 $\text{bestCost} \leftarrow \text{newCost}$
 $\text{bestMV} \leftarrow \vec{q}_{\text{rank}, \vec{h}}$
end if
 $\text{rank} \leftarrow \text{rank} + 1$
end while
return bestMv

\vec{x}_1				\vec{x}_2					\vec{x}_3
	$\vec{q}_{1,1}$	$\vec{q}_{2,1}$	$\vec{q}_{3,1}$						
	$\vec{q}_{4,1}$	\vec{h}_1	$\vec{q}_{5,1}$	\vec{h}_2		\vec{h}_3			
	$\vec{q}_{6,1}$	$\vec{q}_{7,1}$	$\vec{q}_{8,1}$						
\vec{x}_4		\vec{h}_4		\vec{m}		\vec{h}_5			\vec{x}_5
		$\vec{q}_{6,1}$		\vec{h}_7		\vec{h}_8			
\vec{x}_6				\vec{x}_7					\vec{x}_8

Fig. 5. Example of a search step performed by the introduced algorithm.

the LUT row indicated by the context, the LUT needs to be selected first based on the outcome of the context calculation. Then, the corresponding LUT row is chosen based on the result of the half sample refinement \vec{h} . The full algorithm is outlined in Algorithm 3.

D. Example of a Search Step

To further illustrate the interaction of half and quarter sample refinement as described in Sections IV-B and IV-C, an example search step of the overall algorithm is shown in Fig. 5. In this example, it is assumed that the context calculation according to (5) yields $\text{ctx}(d_{\text{SAD}}) = 1$. The upper bound u of both the half sample and the quarter sample refinement is set to 3. According to Table I, the three highest-ranked half

 TABLE III
 EVALUATED COMMON TEST CONDITIONS SEQUENCES [24]

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		

 TABLE IV
 ADDITIONAL SEQUENCES USED FOR EXPERIMENTAL EVALUATION (CLASS HD)

Name	Description
FountainWoman	Woman dancing around a fountain
Hannah	Dance sequence from the movie "Hannah Montana"
MiBA	Sequence from the movie "Men in Black 3"
MiBB	Sequence from the movie "Men in Black 3"
Surfers	Two wind surfers
Harbour	A ship shooting water in the air in an harbour
Garden	Shots from flowers and animals in a garden
Mosaic	Shots from walls decorated with mosaics
Buildings	A couple of buildings
FootballFans	A crowd of football fans in a stadium
FootballA	Sequence from a football game
FootballB	Sequence from a football game

sample positions for that context are \vec{h}_1 , \vec{h}_2 , and \vec{h}_4 . These half sample positions are indicated by green squares in Fig. 5. Hence, during half sample refinement, these three positions in addition to the initially selected integer sample position \vec{m} are checked. It is further assumed that \vec{h}_1 was found to be the best half sample position, i.e., its distortion $d_{\text{HAD}}(\vec{h}_1)$ is lower than the distortion of the other three. Since context calculation yields $\text{ctx}(d_{\text{SAD}}) = 1$, Table II is chosen to be the quarter sample refinement LUT. Given \vec{h}_1 as the outcome of the half sample refinement, the corresponding row is selected, which leads to the three highest-ranked quarter sample positions \vec{q}_8 , \vec{q}_7 , and \vec{q}_5 . These quarter sample positions are indicated by red squares in Fig. 5. If the HAD distortion of one of these three positions is lower than $d_{\text{HAD}}(\vec{h}_1)$, this position is selected as the final fractional MV, otherwise it is set to \vec{h}_1 .

V. TEST METHODOLOGY AND EXPERIMENTAL RESULTS

The experimental results have been obtained by using the HEVC reference software HM 12.0 [25]. Although at the time of writing HM is available in version 16.10, it is important to note that this is a valid baseline since no further changes regarding fractional sample refinement were included in between these versions. The random access (RA) and low delay P (LDP) configurations have been considered, according to the HEVC common test conditions, as defined in [24] and specified in Table III. It should be noted that the *Class E* sequences represent typical video conferencing content, which corresponds to relatively low motion activity. Also, the *Class F* sequences mainly consist of computer-generated imagery content. Additionally, sequences as listed in Table IV, all of which in a resolution of 1920×1080 with 50 frames/s and a bit depth of 8, have been used for experimental verification. They are referred to as *Class HD* sequences in the remainder.

TABLE V

CODING EFFICIENCY GAIN (BD-RATE) FOR PERFORMING FRACTIONAL SAMPLE REFINEMENT COMPARED TO CONSIDERING INTEGER SAMPLE POSITIONS ONLY IN HM

	A	B	C	D	E	F	HD
RA	-7.6%	-8.1%	-11.2%	-12.4%	-	-2.8%	-7.4%
LDP	-	-13.8%	-19.7%	-23.5%	-9.4%	-4.9%	-16.1%

TABLE VI

CODING EFFICIENCY GAINS (BD-RATE) FOR THE RA CONFIGURATION FOR SEQUENCES FROM TABLE III AND TABLE IV COMPARED WITH INTEGER ME IN HM

Random Access							
Class	$u = 1$	Proposed $u = 2$	$u = 3$	Cross	2plus3	Direction	DField9
A	-6.5%	-7.1%	-7.4%	-7.5%	-7.2%	-6.0%	-6.7%
B	-6.8%	-7.5%	-7.8%	-8.1%	-7.7%	-5.6%	-7.0%
C	-9.1%	-10.0%	-10.6%	-11.0%	-10.6%	-8.8%	-9.1%
D	-9.7%	-10.9%	-11.8%	-12.2%	-11.5%	-9.4%	-9.6%
F	-2.2%	-2.4%	-2.5%	-2.8%	-2.7%	-2.4%	-1.9%
HD	-6.3%	-6.8%	-7.0%	-7.2%	-7.0%	-5.4%	-6.3%
AVG	-6.7%	-7.3%	-7.6%	-7.9%	-7.6%	-6.0%	-6.6%

TABLE VII

CODING EFFICIENCY GAINS (BD-RATE) FOR THE LDP CONFIGURATION FOR SEQUENCES FROM TABLE III AND TABLE IV COMPARED WITH INTEGER ME IN HM

Low Delay P							
Class	$u = 1$	Proposed $u = 2$	$u = 3$	Cross	2plus3	Direction	DField9
B	-12.1%	-12.8%	-13.4%	-13.7%	-13.2%	-11.0%	-12.3%
C	-17.0%	-18.1%	-19.0%	-19.5%	-18.9%	-16.7%	-17.2%
D	-19.9%	-21.6%	-22.7%	-23.2%	-22.5%	-20.5%	-19.9%
E	-8.2%	-8.7%	-9.2%	-9.2%	-8.9%	-8.7%	-8.1%
F	-4.1%	-4.4%	-4.6%	-4.7%	-4.7%	-4.3%	-4.0%
HD	-14.5%	-15.8%	-15.8%	-16.0%	-15.6%	-13.6%	-14.9%
AVG	-13.2%	-14.3%	-14.7%	-14.9%	-14.5%	-12.8%	-13.4%

Table V presents the corresponding coding efficiency gains for both the RA and LDP configurations by enabling the fractional sample refinement compared to considering the integer sample positions only, when applying the already mentioned *interpolation-and-search* method presented in Section II-A. The difference in RD performance is described by the means of the Bjøntegaard-Delta (BD)-rate as proposed in [23]. It should be noted that the RA configuration allows a weighted combination of two integer sample accuracy prediction signals (B-frames). Although both prediction signals are obtained by integer sample precision ME, the combination may represent a prediction obtained by a fractional sample displacement. In turn, this justifies the significantly higher performance gains for the LDP configuration, as shown in Table V, since in such a case only one inter-picture prediction signal is allowed (P-frame).

Regarding computational complexity, this approach incorporates eight search points for the half sample refinement and additional eight points for the quarter sample accuracy search, including the corresponding up-sampling operations.

TABLE VIII

CODING EFFICIENCY LOSSES (BD-RATE) FOR THE RA CONFIGURATION FOR SEQUENCES FROM TABLE III AND TABLE IV COMPARED WITH THE INTERPOLATION-AND-SEARCH ALGORITHM IN HM

Random Access							
Class	$u = 1$	Proposed $u = 2$	$u = 3$	Cross	2plus3	Direction	DField9
A	1.1%	0.6%	0.3%	0.2%	0.4%	1.7%	1.0%
B	1.5%	0.8%	0.4%	0.0%	0.5%	2.8%	1.3%
C	2.4%	1.4%	0.7%	0.2%	0.7%	2.8%	2.3%
D	3.0%	1.8%	0.7%	0.2%	1.0%	3.5%	3.2%
F	0.7%	0.4%	0.3%	0.0%	0.0%	0.5%	0.9%
HD	1.3%	0.6%	0.4%	0.2%	0.4%	2.2%	1.2%
AVG	1.7%	1.0%	0.5%	0.1%	0.5%	2.3%	1.7%

TABLE IX

CODING EFFICIENCY LOSSES (BD-RATE) FOR THE LDP CONFIGURATION FOR SEQUENCES FROM TABLE III AND TABLE IV COMPARED WITH THE INTERPOLATION-AND-SEARCH ALGORITHM IN HM

Low Delay P							
Class	$u = 1$	Proposed $u = 2$	$u = 3$	Cross	2plus3	Direction	DField9
B	2.1%	1.2%	0.5%	0.1%	0.7%	3.3%	1.8%
C	3.3%	1.9%	0.8%	0.3%	0.9%	3.7%	3.1%
D	4.8%	2.6%	1.1%	0.4%	1.4%	4.1%	4.7%
E	1.3%	0.8%	0.3%	0.3%	0.6%	0.8%	1.4%
F	0.9%	0.5%	0.3%	0.2%	0.2%	0.6%	1.0%
HD	2.1%	0.9%	0.3%	0.1%	0.6%	3.4%	1.6%
AVG	2.5%	1.4%	0.6%	0.3%	0.8%	2.5%	2.3%

A. Experimental Results

In the following, experimental results by applying Algorithms 2 and 3 with $u = 1$, $u = 2$, and $u = 3$ under previously described test conditions are presented. Tables VI and VII demonstrate the coding efficiency gain achieved by the algorithms compared to applying integer ME only. In comparison with Table V, it can be seen that for $u = 1$ most of the coding efficiency gain achieved by fractional sample refinement can be preserved. For instance, under the RA configuration, HM achieves $-12.4%$ BD-rate using fractional sample refinement compared with integer ME only for *Class D*. Comparing the proposed approach to integer ME only still results in $-9.7%$ BD-rate, which is $\sim 78%$ of the gain achieved by the reference algorithm. Especially, for the HD resolution sequences (i.e., *Classes B and HD*), the proposed approach achieves $-6.48%$ BD-rate, resulting in approximately 85% of the encoding gain obtained by applying the *interpolation-and-search* algorithm. For results presented in Tables VIII and IX, the *interpolation-and-search* algorithm of HM was used as a baseline. As can be seen, for $u = 3$, only small BD-rate losses are obtained for both the RA and LDP configurations. Furthermore, it is important to note that, as has been mentioned in Section IV, the first ten frames of sequences listed in Table III formed the training set; hence, these frames were part of both the training and the testing phase which generally should be avoided. However, the number of frames was exceeded during the testing phase as all the test sequences are longer than ten frames. In order to further demonstrate that the

TABLE X

SEARCH POINTS FOR CONFIGURATIONS PRESENTED IN TABLE VIII AND TABLE IX FOR HALF AND QUARTER SAMPLE SEARCH EACH

	Proposed $u = 1$	Proposed $u = 2$	Proposed $u = 3$	Cross	2plus3	Direction	DField9
min	1	2	3	4	3	1	1
max	1	2	3	6	5	8	1

approach indeed is applicable to sequences outside the training set, sequences from Table IV have been added to the testing set. Results for these are also presented separately. As can be seen, the algorithm performs as good for these sequences.

The computational complexity of the proposed context-based fractional sample refinement algorithm is a function of search points to be examined, which clearly depends on the value of u . In this respect, Table X provides an overview regarding the reduction of search points of the proposed context-based fractional sample refinement algorithm compared with the conventional *interpolation-and-search* method for both the RA and LDP configurations. As can be seen, the number of search points is reduced significantly. It should also be noted that, similar to approaches discussed in Section II-C, for $u = 1$ no interpolation is required.

B. Comparison to Other Fast Fractional Sample Refinement Schemes

In order to demonstrate the advantage of the proposed approach, simulations were performed to compare the coding efficiency as well as the run time of the presented example implementation with other recently proposed fast fractional sample search algorithms. For this purpose, simulations were performed for representative algorithms from categories as presented in [26] and [27]:

- 1) fast full search algorithms;
- 2) reduction of searching candidate points in search range;
- 3) modeling the matching error surface into mathematical models.

With respect to 1), the *interpolation-and-search* method is considered. Corresponding results have been presented in Section V-A. Algorithms matching category 2) have been discussed in Section II-B and simulations were performed for the pattern-based approaches in the following referred to as *Cross* [16], *2plus3* [17], and *Direction* [18]. Regarding category 3), an approach approximating the error surface based on surrounding integer sample position distortions, here referred to as *DField9* [20] and discussed in Section II-C, was used.

The comparison of coding efficiency losses for the RA configuration is presented in Table VIII and for the LDP configuration in Table IX. With respect to the RA and the LDP configuration, it can be seen that in average only the approach referred to as *Cross* outperforms our approach letting $u = 3$ in terms of coding efficiency. However, the proposed approach performs only three checks on fractional sample positions, while for *Cross* at least four and in the worst case six are necessary. Also, it can be seen that using $u = 1$ performs better

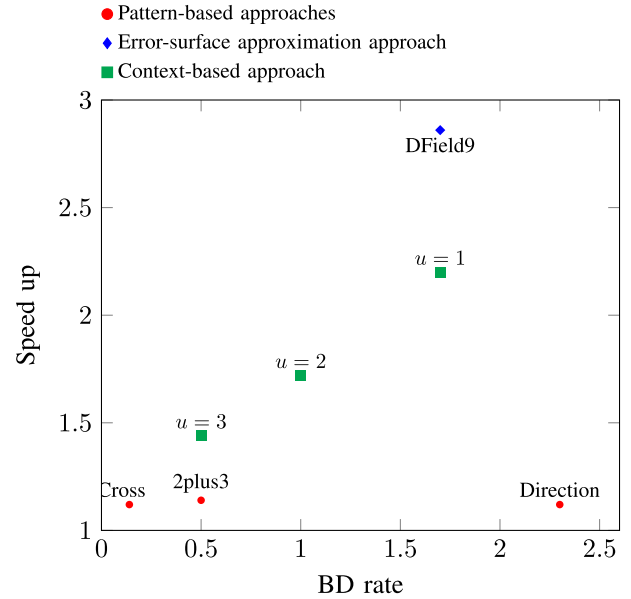


Fig. 6. Comparison of algorithms for the RA configuration using all sequences listed in Table III and Table IV averaged over all four quantization parameters in HM.

than or equal to the algorithm presented in *Direction* both in terms of encoding efficiency and required position checks.

An overview comparing the number of fractional sample positions to be checked is provided in Table X. It is important to note that for the proposed approach depending on the value of u , the number of best-case checks equals the number of worst case checks. For instance, selecting $u = 2$ will result in always two positions being considered. The error surface approximation algorithm denoted as *DField9* requires only one position being considered, making it equal in this respect to using the approach proposed in this paper with $u = 1$. In the course of this paper, the error surface approximation method was implemented using SAD as the distortion measure.

Besides the number of fractional sample search positions listed in Table X, the time required for fractional sample refinement is measured in the simulations as another indicator for computational complexity. A comparison of total encoding times would be less conclusive since the overall run times strongly depend on the degree of encoder optimization. HM as reference software for instance performs full RD mode decisions which leads to a very high overall encoding time. An optimized encoder would spend less time on mode decisions by applying fast decision heuristics. Hence, the contribution of fractional motion search to the overall encoding time would be larger in an optimized encoder compared with HM.

To provide for a comparison of required computation time, the time spent on fractional sample ME was taken for each algorithm. Results obtained for all aforementioned algorithms have been averaged over all four quantization parameters [24]. The average results for all sequences are also illustrated in Figs. 6 and 7 in relation to coding efficiency. Speedup values obtained for the different classes in no case differed by more than 0.1 from these averages. The speedup expressed in aforementioned tables and figures was obtained using the

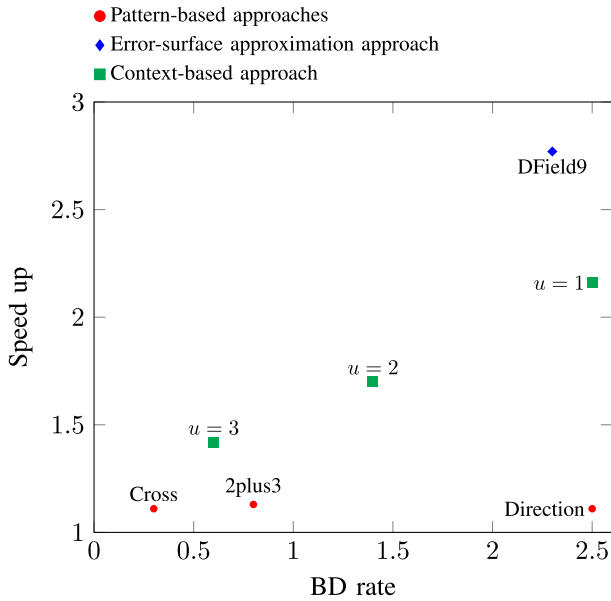


Fig. 7. Comparison of algorithms for the LDP configuration using all sequences listed in Table III and Table IV averaged over all four quantization parameters in HM.

interpolation-and-search method as discussed in Section II-A as a baseline. As can be seen, the pattern-based approaches respected in this simulation do not achieve a significant decrease in fractional sample ME time. However, they cause losses in BD performance. *DField9* and the proposed context-based approach on the other hand yield a good tradeoff between RD performance and computational complexity, especially with respect to high-resolution sequences. From results presented before, it becomes clear that the algorithm indeed can achieve performance comparable to the error surface approximation method. The computational overhead caused by context calculation is negligible when regarded in comparison with the overall speedup. However, as has already been argued, the context-based approach can be implemented in a much more flexible manner, allowing for an adaptive decision regarding how many samples to consider.

VI. CONCLUSION AND FUTURE WORK

In this paper, a generic context-based fractional sample MV refinement framework and an example application for an HEVC encoder are presented. The proposed example algorithm reduces the 16 fractional sample search points of the *interpolation-and-search* method employed in the HEVC HM reference encoder to a few most promising ones by evaluating context information based on the neighboring integer sample distortions. When only the two most promising search points are considered, i.e., a search point reduction of 87.5%, coding efficiency losses between 1.3% and 2.1% BD-rate are observed for HD test sequences. Considering the six most promising search points, i.e., the three most promising for half sample MV refinement plus the three most promising for quarter sample MV refinement, coding efficiency losses range between 0.3% and 0.4% BD-rate. Compared with state-of-the-art pattern-based fast fractional sample refinements, the

proposed example algorithm always provides a better tradeoff in terms of search point reduction versus coding efficiency loss. When comparing the example algorithm with fractional sample MV refinement by error surface approximation, the advantage of the proposed framework lies in its adaptivity, which allows operation points with lower coding efficiency loss at the cost of decreased speedup.

For future work, this adaptivity is of particular interest, e.g., by determining the number of promising search points to be evaluated based on an optimal tradeoff between the coding efficiency loss and/or a given budget of computation time. For instance, in addition to defining a fixed upper bound for promising search points, a minimal performance gain can be defined. Then, most promising fractional sample positions given a certain context are only considered until the upper bound is reached or the current performance gain is less than the defined minimal performance gain. Also, in order to meet real-time processing constraints, an adaptive time-constrained parameter may be further employed. In this strategy, an additional search point is considered only when it is worth to invest additional processing time in terms of computational complexity. Besides adaptive termination, further investigation is encouraged to improve the context model $ctx(C)$ which might lead to a better prediction of promising search points. Finally, the number of fractional sample positions to be searched in the proposed framework is constant. This is of particular interest for hardware implementations that always consider the worst case, i.e., the maximum number of search positions. Hence, the investigation of a corresponding hardware design is another interesting topic for future work.

REFERENCES

- [1] F. Bossen, B. Bross, K. Sühling, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685–1696, Dec. 2012.
- [2] B. Bross, H. Schwarz, and D. Marpe, "The new High-Efficiency Video Coding standard," *SMPTE Motion Imag. J.*, vol. 122, no. 4, pp. 25–35, 2013.
- [3] D. Marpe *et al.*, "Video compression using nested quadtree structures, leaf merging and improved techniques for motion representation and entropy coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1676–1687, Dec. 2010.
- [4] P. Helle *et al.*, "Block merging for quadtree-based partitioning in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1720–1731, Dec. 2012.
- [5] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.
- [6] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
- [7] J. 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 Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.
- [8] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar, "Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders," in *Proc. Picture Coding Symp.*, Dec. 2013, pp. 394–397.
- [9] D. Grois, D. Marpe, T. Nguyen, and O. Hadar, "Comparative assessment of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders for low-delay video applications," *Proc. SPIE*, vol. 9217, p. 92170Q, Sep. 2014.
- [10] B. Bross *et al.*, "HEVC performance and complexity for 4K video," in *Proc. IEEE 3rd Int. Conf. Consum. Electron. (ICCE)*, Berlin, Germany, Sep. 2013, pp. 44–47.

- [11] H. A. Choudhury and M. Saikia, "Survey on block matching algorithms for motion estimation," in *Proc. Int. Conf. Commun. Signal Process. (ICCSPP)*, Apr. 2014, pp. 036–040.
- [12] G. Maier *et al.*, "Pattern-based integer sample motion search strategies in the context of HEVC," *Proc. SPIE*, vol. 9599, p. 95991A, Sep. 2015.
- [13] D. Kang, Y. Kang, and Y. Hong, "VLSI implementation of fractional motion estimation interpolation for High Efficiency Video Coding," *Electron. Lett.*, vol. 51, no. 15, pp. 1163–1165, 2015.
- [14] G. Sanchez, M. Corrêa, D. Noble, M. Porto, S. Bampi, and L. Agostini, "Hardware design focusing in the tradeoff cost versus quality for the H.264/AVC fractional motion estimation targeting high definition videos," *Analog Integr. Circuits Signal Process.*, vol. 73, no. 3, pp. 931–944, 2012.
- [15] G. He, D. Zhou, Y. Li, Z. Chen, T. Zhang, and S. Goto, "High-throughput power-efficient VLSI architecture of fractional motion estimation for ultra-HD HEVC video encoding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 12, pp. 3138–3142, Dec. 2015.
- [16] Y. Zhang, W.-C. Siu, and T. Shen, "Fast sub-pixel motion estimation based on directional information and adaptive block classification," in *Proc. 5th Int. Conf. Vis. Inf. Eng. (VIE)*, Jul./Aug. 2008, pp. 622–627.
- [17] H. Nisar and T.-S. Choi, "Fast and efficient fractional pixel motion estimation for H.264/AVC video coding," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 1561–1564.
- [18] Z. Wei, F. Fen, W. Xiaoyang, and Z. Weile, "Directionality based fast fractional pel motion estimation for H.264," *J. Syst. Eng. Electron.*, vol. 20, no. 3, pp. 457–462, Jun. 2009.
- [19] T. Sotetsumoto, T. Song, and T. Shimamoto, "Low complexity algorithm for sub-pixel motion estimation of HEVC," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Aug. 2013, pp. 1–4.
- [20] J. W. Suh and J. Jeong, "Fast sub-pixel motion estimation techniques having lower computational complexity," *IEEE Trans. Consum. Electron.*, vol. 50, no. 3, pp. 968–973, Aug. 2004.
- [21] W. Lin, K. Panusopone, D. M. Baylon, M.-T. Sun, Z. Chen, and H. Li, "A fast sub-pixel motion estimation algorithm for H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 2, pp. 237–242, Feb. 2011.
- [22] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.
- [23] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-M33, ITU-T, 2001.
- [24] F. Bossen, *Common Test Conditions and Software Reference Configurations*, document JCTVC-L1100, Joint Collaborative Team on Video Coding (JCT-VC), 2013.
- [25] *Subversion Repository for the HEVC Test Model (HM) Reference Software*, accessed on Nov. 29, 2013. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware
- [26] X. Liyin, S. Xiuqin, and Z. Shun, "A review of motion estimation algorithms for video compression," in *Proc. Int. Conf. Comput. Appl. Syst. Modeling (ICCSAM)*, Taiyuan, China, 2010, pp. V2-446–V2-450.
- [27] S. Ashwin, S. J. Sree, and S. A. Kumar, "Study of the contemporary motion estimation techniques for video coding," *Int. J. Recent Technol. Eng.*, vol. 2, no. 1, pp. 190–194, Mar. 2013.



Georg Maier received the M.Sc. degree in computer science in the program game and media technology from Utrecht University, Utrecht, The Netherlands, in 2013.

He was a Research Assistant with the Image and Video Coding Group, Fraunhofer Institute for Telecommunications–Heinrich Hertz Institute, Berlin, Germany, where he was involved in motion estimation and efficient implementations thereof for the H.265/MPEG High Efficiency Video Coding standard. Since 2014, he has been with the Fraun-

hofer Institute of Optronics, System Technologies and Image Exploitation, (Fraunhofer IOSB), Karlsruhe, Germany, where he has been involved in visual inspection systems for the automated sorting of bulk goods. His research interests include different aspects of image processing, in particular algorithmic aspects, with a focus on real-time capabilities.



Benjamin Bross (S'11) received the Dipl.-Ing. degree in electrical engineering from RWTH Aachen University, Aachen, Germany, in 2008, with a focus on 3D image registration in medical imaging and on decoder side motion vector derivation in H.264/MPEG-4 Advanced Video Coding.

In 2009, he joined the Fraunhofer Institute for Telecommunications–Heinrich Hertz Institute, Berlin, Germany, where he is currently a Project Manager with the Video Coding and Analytics Department. Since the development of the new H.265/MPEG High Efficiency Video Coding (HEVC) standard, which started in 2010, he was very actively involved in the standardization process as a Technical Contributor and Coordinator of core experiments. In 2011, he was a part-time Lecturer with the HTW University of Applied Sciences, Berlin. In 2012, he was appointed as a Co-Chair with the Editing Ad Hoc Group and became the Chief Editor of the HEVC video coding standard. He is currently with the Fraunhofer Institute for Telecommunications–Heinrich Hertz Institute, Berlin, where he is responsible for developing HEVC conforming real-time encoders and decoders as well as investigating new video coding techniques for the next generation of video coding standards. He has authored or co-authored several fundamental HEVC-related publications, and authored two book chapters on HEVC and Inter-Picture Prediction Techniques in HEVC.

Mr. Bross received the IEEE Best Paper Award at the 2013 IEEE International Conference on Consumer Electronics, Berlin, and the SMPTE Journal Certificate of Merit in 2014.



Dan Grois (SM'11) received the Ph.D. degree from the Communication Systems Engineering Department, Ben-Gurion University of the Negev (BGU), Beersheba, Israel, 2011.

From 2011 to 2013, he was a Senior Researcher with the Communication Systems Engineering Department, BGU. Since 2013, he has been a Post-Doctoral Senior Researcher with the Video Coding and Analytics Department, Fraunhofer Institute for Telecommunications–Heinrich Hertz Institute, Berlin, Germany. He has authored or co-authored

about 40 publications in the area of image/video coding and data processing, which have been presented at the top-tier international conferences and were published in various scientific journals and books. His research interests include image and video coding and processing, video coding standards, particularly H.265/MPEG High Efficiency Video Coding, region-of-interest scalability, computational complexity and bit-rate control, network communication and protocols, and future multimedia applications/systems.

Dr. Grois is a member of the ACM and SPIE societies. He was received various fellowships, including Kreitman Fellowships and the ERCIM Alain Bensoussan Fellowship, which was provided by the FP7 Marie Curie Actions COFUND Programme. He is currently a fellow of the PROVISION ITN Project, which is a part of the European Unions Marie Skłodowska-Curie Actions of the European Commission. He is a Referee of top-tier conferences and international journals, such as IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON SIGNAL PROCESSING, *Journal of Visual Communication and Image Representation* (Elsevier), IEEE SENSORS, and *Optical Engineering* (SPIE). In 2013, he also served as the Guest Editor of *Optical Engineering* (SPIE).



Detlev Marpe (M'00–SM'08–F'15) received the Dipl.-Math. degree (Hons.) from Technical University of Berlin, Berlin, Germany, in 1990 and the Dr.-Ing. degree from University of Rostock, Rostock, Germany, in 2004.

He joined Fraunhofer Institute for Telecommunications–Heinrich Hertz Institute, Berlin, in 1999, where he is currently the Head of the Video Coding & Analytics Department and Head of the Image and Video Coding Research Group. He was a major Technical Contributor to

the entire process of the development of the H.264/MPEG-4 Advanced Video Coding (AVC) standard and the H.265/MPEG High Efficiency Video Coding (HEVC) standard, including several generations of major enhancement extensions. In addition to the CABAC contributions for both standards, he particularly contributed to the Fidelity Range Extensions (which include the High Profile that received the Emmy Award in 2008) and the Scalable Video Coding Extensions of H.264/MPEG-4 AVC. During the recent development of its successor H.265/MPEG-HEVC, he also successfully contributed to the first model of the corresponding standardization project and further refinements. He also made successful proposals to the standardization of its Range Extensions and 3D Extensions. His academic work includes over 200 publications in image and video coding. He holds over 250 internationally issued patents and numerous patent applications in this field. His research interests include still image and video coding, signal processing for communications and computer vision, and information theory.

Dr. Marpe is a member of the Informationstechnische Gesellschaft of the Verband der Elektrotechnik Elektronik Informationstechnik e.V. He was a co-recipient of two Technical Emmy Awards as a Key Contributor and a Co-Editor of the H.264/MPEG-4 AVC standard in 2008 and 2009, respectively. He received the IEEE Best Paper Award at the 2013 IEEE International Conference on Consumer Electronics, Berlin, and the SMPTE Journal Certificate of Merit in 2014. He was nominated for the German Future Prize in 2012. He was a recipient of the Karl Heinz Beckurts Award in 2011, the Best Paper Award of the IEEE Circuits and Systems Society in 2009, the Joseph von Fraunhofer Prize in 2004, and the Best Paper Award of the Informationstechnische Gesellschaft in 2004. As a Co-Founder of the Berlin-based daviko GmbH, he received the Prime Prize of the Multimedia Start-Up Competition of the German Federal Ministry of Economics and Technology in 2001. Since 2014, he has been an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.



Heiko Schwarz received the Dipl.-Ing. degree in electrical engineering and the Dr.-Ing. degree from the University of Rostock, Rostock, Germany, in 1996 and 2000, respectively.

In 1999, he joined the Image and Video Coding Group, Fraunhofer Institute for Telecommunications–Heinrich Hertz Institute, Berlin, Germany. He has contributed successfully to the standardization activities of the ITU-T Video Coding Experts Group (ITU-T SG16/Q.6-VCEG) and the ISO/IEC Moving Pictures Experts Group (ISO/IEC JTC 1/SC 29/WG 11- MPEG).

Dr. Schwarz has been appointed as a Co-Editor of ITU-T H.264 and ISO/IEC 14496-10 and as a Software Coordinator for the SVC reference software. During the development of the scalable video coding extension of H.264/AVC, he co-chaired several ad hoc groups of the Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, investigating the particular aspects of the scalable video coding design.



Remco C. Veltkamp obtained a M.Sc. degree in computer Science at Leiden University, and a Ph.D. degree in computer science at Erasmus University Rotterdam, the Netherlands. He is currently a Full Professor of Multimedia with Utrecht University, Utrecht, The Netherlands. He has authored over 150 refereed papers in reviewed journals and conferences, and supervised 15 Ph.D. thesis. His research interests include the analysis, recognition and retrieval of, and interaction with, music, images, and 3D objects and scenes, in particular the algorithmic and experimentation aspects.

and experimentation aspects.



Thomas Wiegand (M'05–SM'08–F'11) received the Dipl.-Ing. degree in electrical engineering from Technical University of Hamburg, Hamburg, Germany, in 1995 and the Dr.-Ing. degree from University of Erlangen–Nuremberg, Erlangen, Germany, in 2000.

He was a Visiting Researcher with Kobe University, Kobe, Japan; University of California at Santa Barbara, Santa Barbara, CA, USA; and Stanford University, Stanford, CA, USA, where he also returned as a Visiting Professor. He was a Consultant with Skyfire, Inc., Mountain View, CA, USA. Since 1995, he has been an active participant in standardization for multimedia with many successful submissions to ITU Telecommunication Standardization Sector and International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC). In 2000, he was appointed as the Associated Rapporteur of the ITU-T Video Coding Experts Group. He is currently a Professor with the Department of Electrical Engineering and Computer Science, Technical University of Berlin, Berlin, Germany. He is also the Head of the Fraunhofer Institute for Telecommunications–Heinrich Hertz Institute, Berlin. He is also a Consultant with Vidyio, Inc., Hackensack, NJ, USA.

Dr. Wiegand was a recipient of the ITU150 Award. Thomson Reuters named him in their list of the World's Most Influential Scientific Minds 2014 as one of the most cited researchers in his field. The projects that he co-chaired for the development of the H.264/Moving Pictures Experts Group–Advanced Video Coding standard have been recognized by the ATAS Primetime Emmy Engineering Award and a pair of the NATAS Technology and Engineering Emmy Award. For his research in video coding and transmission, he received numerous awards, including the Vodafone Innovations Award, the EURASIP Group Technical Achievement Award, the Eduard Rhein Technology Award, the Karl Heinz Beckurts Award, the IEEE Masaru Ibuka Technical Field Award, and the IMTC Leadership Award. He received multiple best paper awards for his publications. From 2005 to 2009, he was the Co-Chair of the ISO/IEC MPEG Video.

From 2005 to 2009, he was the Co-Chair of the ISO/IEC MPEG Video.