Smart Search Space Reduction for Approximate Computing: a Low Energy HEVC Encoder Case Study

Alexandre Mercat, Justine Bonnot, Maxime Pelcat, Karol Desnos, Wassim Hamidouche, Daniel Menard

Smart Search Space Reduction for Approximate Computing: a Low Energy HEVC Encoder Case Study

Alexandre Mercat[a,**], Justine Bonnot[a], Maxime Pelcat[a,b], Karol Desnos[a], Wassim Hamidouche[a], Daniel Menard[a,*]

[a] *UBL, INSA Rennes, IETR, CNRS UMR 6164, Rennes, France*
[b] *Institut Pascal, CNRS UMR 6602, Clermont-Ferrand, France*

Abstract

The approximate computing paradigm provides methods to optimize algorithms while considering both application quality of service and computational complexity. Approximate computing can be applied at different levels of abstraction, from algorithm level to application level. Approximate computing at algorithm level reduces the computational complexity by approximating or skipping computational blocks. A number of applications in the signal and image processing domain integrate algorithms based on discrete optimization techniques. These techniques minimize a cost function by exploring an application parameter search space. In this paper, a new methodology is proposed that exploits the computation-skipping approximate computing concept. The methodology, named Smart Search Space Reduction (SSSR), explores at design time the Pareto relationship between computational complexity and application quality. At run time, an approximation manager can then early select a good candidate configuration. SSSR reduces the run time search space and, in turn, reduces computational complexity. An efficient SSSR technique adjusts at design time the configuration selectivity while selecting at run time the most suitable functions to skip. The real time High Efficiency Video Coding (HEVC) encoder in All Intra (AI) profile is used as a case study to illustrate the benefits of SSSR. In this application, two discrete optimizations are performed. They explore different coding parameters and select the values leading to the minimal cost in terms of a tradeoff between bitrate, quality and computational energy by acting on both the HEVC coding-tree partitioning and the intra-modes. Combining two SSSRs iterations on this use case, the energy consumption is reduced by up to 77%. Moreover, the combination of the two SSSRs iterations in comparison to using only one reduces the BD-BR bitrate/quality metric by 4% for the same energy consumption.

Keywords: approximate computing, high efficiency video coding, energy optimization.

1. Introduction

Nowadays, optimizing energy consumption in embedded systems is of primary concern for the design of autonomous devices. The emerging domain of the Internet of Things (IoT) requires designing ultra low-power systems. To reduce the computational energy consumption, new techniques from circuit to system levels have been proposed in the last two decades. Technologies such as FinFet [33] and Fully Depleted Silicon on Insulator (FD-SOI) [29] continue the trend of shrinking down transistors and reducing their leakage current, providing more energy efficient hardware. At system level, methods are developed to adapt the instantaneous processing capacity to the requirements of the running applications. Two power management techniques are mainly used to minimize the energy consumption of modern Systems-on-Chips, namely DPM and DVFS. Dynamic Power Management (DPM) [1] consists in combining clock gating and power gating to turn a processing core into a low-power state when it is idle. Dynamic Voltage and Frequency Scaling (DVFS) [23] minimizes the consumed dynamic power by reducing both hardware clock frequency and supply voltage until real-time constraints are met.

Approximate computing relies on the ability of many applications and systems to tolerate some loss of quality in the computed result. The approximate computing paradigm has been emerging for a decade and can produce significant improvements from technological to system levels. Research works focus more on technological, logic [6], and architecture levels while algorithm level has not been widely investigated despite the significant energy gain opportunities. At the algorithm level, incremental refinement [22] has been proposed for iterative processes and loop perforation [37] for repetitive structures. In this paper, the focus is put on algorithms that use discrete optimization techniques, exploring a parameter search space, to minimize a cost function. The optimization process contributes in creating costly algorithms and most existing optimization methods tend to raise the data dependency of the considered algorithm complexity. These algorithms are referred to as Minimization based on Search Space Ex-

*Corresponding author
**Principal corresponding author
Email addresses: amercat@insa-rennes.fr (Alexandre Mercat), dmenard@insa-rennes (Daniel Menard)

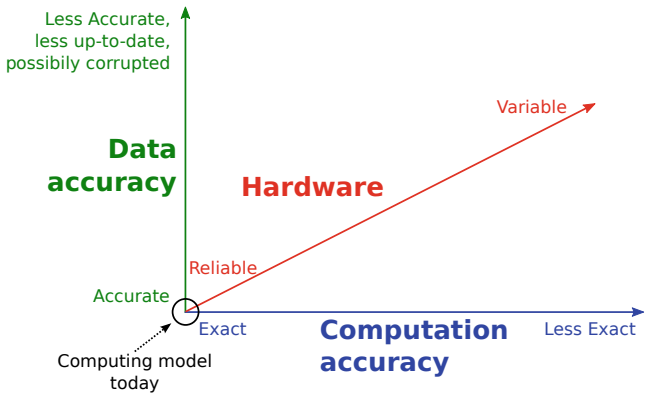

Figure 1: The dimensions of approximate computing [32]

ploration (MSSE) algorithms in the rest of the paper. To decrease the energy consumption of MSSE algorithms, a challenge is to reduce the search space

In this paper, a new approach is proposed to exploit the computation-skipping concept by using a Smart Search Space Reduction (SSSR) technique. This technique is not automatic but aims at classifying computing functions according to their potential cost gain to reduce the search space. Designing efficient SSSR techniques and adjusting the parameters that balance the complexity reduction and the efficiency of the approximate application is not trivial and needs an in-depth knowledge of the application. An All Intra (AI) profile High Efficiency Video Coding (HEVC) encoder is used as a case study to illustrate the benefits of the proposed approach. The HEVC encoder integrates several algorithms based on MSSE, including a coding-tree partitioning that finds a good decomposition of a block into smaller blocks through quad-tree partitioning, and an intra-mode prediction that finds good modes (configurations) to predict a block of pixels from its neighbor pixels. In this work, the two SSSRs are combined to manage efficiently the encoder.

The rest of the paper is organized as follows. We first present the related works in Section 2, and then we introduce the proposed method in Section 3. The method is then applied to analyze the HEVC encoder in Section 4. Finally, Section 5 concludes the paper.

2. Related Works

2.1. Approximate Computing Dimensions

To explore the energy-quality trade off, approximate computing provides three degrees of freedom to act on [32]: data, hardware, and computation (or algorithm) levels, as illustrated in Figure 1.

Approximation on the processed data dimension consists in reducing the quality of the application in a controlled way by using: less up-to-date data (temporal decimation), less input data (spatial decimation), less accurate data (word-length optimization) or even corrupted data.

On the hardware dimension of approximate computing techniques, the exactness of the computation support can be relaxed with different techniques. At the technological level, a voltage overscaling technique can be used to reduce energy consumption. Working at near threshold regime or even at sub threshold regime achieves major energy consumption reductions but leads to computation errors due to an increase in circuit delay. Nevertheless, these timing errors may be tolerable if their probability of occurrence is sufficiently low. Techniques have been proposed to compensate these errors [27] or to maintain these errors within reasonable bounds [19]. At a logic level, a functionality can be approximated by simplifying logic. An approximate hardware module is implemented with a truth table that slightly differs from the exact hardware module. In [35] probabilistic pruning technique is proposed to optimize the logic. The idea is to prune parts of the circuits which are rarely used and for which the approximation error due to pruning is moderate. Approximate computing circuits are designed for specific functional units like adder and multiplier operators. In [21], probabilistic pruning is used to design adders. Different adder circuits [10, 5] are approximated with speculative techniques. Speculative techniques aim at relaxing the constraint on the critical path of the adder corresponding to the carry propagation chain.

The third dimension related to algorithm level is investigated in this paper and analyzed in the next section.

2.2. Approximate Computing at Algorithm Level

A few approximate computing techniques have been proposed to decrease computational complexity with the objective to reduce energy consumption. On the one hand, the inherently error resilient blocks can be modified or skipped, computations can be stopped early to save power, or memory accesses can be ignored. On the other hand, some intricate computations inside these blocks can be approximated. To identify the computation error-resilient blocks (blocks where the impact of errors is limited), approximation -aware programming languages have been created like Eon [38], EnerJ [34], and Rely [7]. Approximation-aware programming languages allow the programmer to identify the parts of the code that can tolerate an error and the parts where higher accuracy is required. Once the different parts of the code of an application have been classified, the next presented methods can be used.

The concept of incremental refinement, introduced in [22], consists in reducing the number of iterations of an iterative processing. By skipping part of the processing, the energy can be reduced. In [8], authors propose to explore the algorithm parameters in a Support Vector Machine algorithm to control application complexity.

Loop perforation is proposed in [37]. The loops that can tolerate approximation errors are identified in a first step, and then transformed to execute a certain number of iterations. In [37] a Pareto front enables the application developer to select the best perforation strategy depending

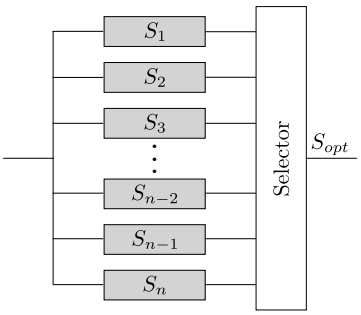

Figure 2: Msse algorithm with exhaustive search. Each branch represents a full solution computation.

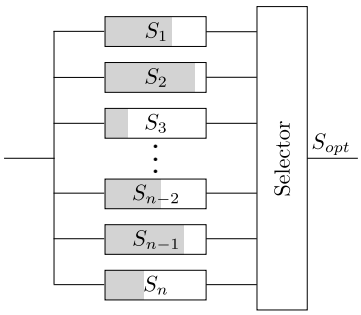

Figure 3: Msse algorithm with branch & bound technique. Early termination stop the exploration of branches which can not lead to the best solution.

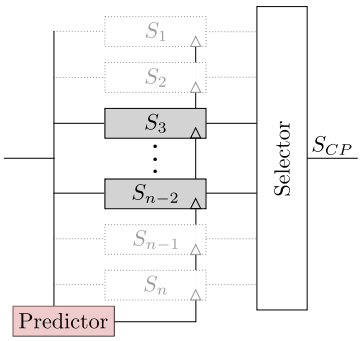

Figure 4: Msse algorithm with Sssr technique. Coarse estimation is carried-out and a refinement is applied to best candidates.

on his requirements and constraints. When applied to tunable loops on applications from the PARSEC benchmark suite [2], loop perforation provides important reduction in terms on computation time, up to seven times, whereas the accuracy has been decreased by less than 10%.

Another solution to reduce computational complexity consists of identifying blocks that can be skipped with a minimum impact on the output [30]. Selected blocks can be permanently or periodically skipped depending on the quality constraints of the application. Once a given trade-off between the required precision and the energy consumption has been reached, a parameter can be used to activate or not the selected blocks. This approximate computing technique can be applied to a domain conservation process [30]. This process is used to enhance a signal property. For example, in an Hevc decoder, the different filters have been approximated to decrease its energy consumption.

As another appoximate computing method, the computation of intricate mathematical functions can be replaced by its approximation. An important advantage of using function approximation is that the targeted precision of the approximated function can be precisely controlled, using well-known algorithms such as multipartite tables [9], CORDIC algorithm [42], or polynomial approximation [3].

2.3. Approximate Computing for Msse Algorithms

In this paper, we focus on processes achieving discrete optimization based on Minimization based on Search Space Exploration (Msse) algorithms for which the main purpose is to minimize a cost function by exploring a search space. As illustrated in Figure 2, the studied Msse algorithms consists of testing different candidate solutions S_i ($i \in [1, n]$) to select the optimal one that minimizes the cost function. Numerous applications in the image and signal processing domain integrate Msse algorithms. In telecommunications, channel decoding and MIMO decoding such as sphere decoding use Msse algorithms. For example, in [13], authors use the properties of an Msse algorithm to select the best transmission configuration including the modulation spectral efficiency and ECC code rate that maximises the quality of a wireless received scalable video over MIMO channels. Msse algorithms are also used in image processing applications for classification operations such as in the Nearest Neighbor classifier, and in video processing applications, for instance in motion estimation [40]. In video compression, as depicted in the case-study presented in Section 4, Msse algorithms are notably used for coding-tree partitioning and Intra-mode prediction.

An exhaustive search, presented in Figure 2, is a straightforward approach to process Msse but it may require much computation, depending on the search space size. A challenge for an Msse algorithm implementation is to reduce the search space while minimizing the impact on the approximated optimal solution $\tilde{S}_{opt}$ compared to the optimal solution under the full search space S_{opt}. Ideally, the optimal solution must be contained in the reduced search space ($\tilde{S}_{opt} = S_{opt}$). To the best of our knowledge, no approach has been proposed in literature that focuses on approximate computing based on Msse algorithms.

3. Approximate Computing Methodology

3.1. Search Space Reduction Techniques

For Msse algorithms, the search space reduction techniques can be classified into two categories: branch-and-bound and Sssr. The branch-and-bound techniques initially consider the entire solution search space. Then, based on intermediate results, the branch-and-bound technique automatically prunes the search space by excluding the least likely solutions. Hence, parts of the search space which can not lead to the optimal solution are removed. The discrete optimization problem can be formulated with a tree representing the different solutions S_i. The branch-and-bound technique can be used to reduce the search space. This technique exploits the concept of early termination. The exploration of the branch is stopped if the minimal cost which can be obtained for the exploration

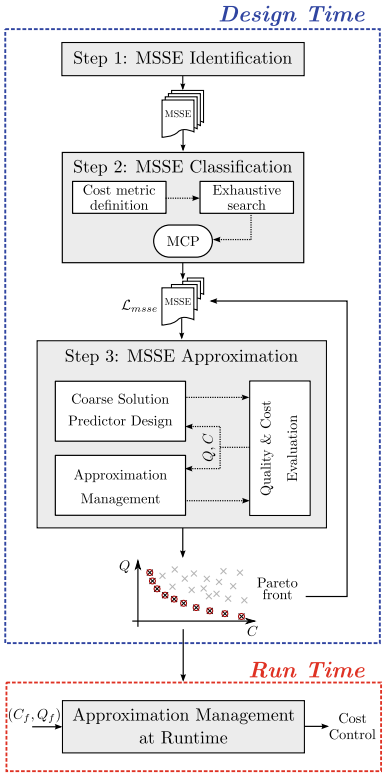

Figure 5: The Smart Search Space Reduction (SSSR) method

Algorithm 1: The Smart Search Space Reduction (SSSR) method

Data: An application $\mathcal{A}$ to optimize and a training data set $\mathcal{D}$
Result: A pareto point cloud over NFP cost C and application quality Q, with points tagged by approximation parameters.

1 Identify the MSSEs in $\mathcal{A}$;
2 Define the Cost metric C;
3 **for** *all MSSEs* **do**
4 Apply an exhaustive search on the MSSE using $\mathcal{D}$;
5 Get the optimal solutions of the MSSE;
6 Force the MSSE to only compute the optimal solution and measure the minimum cost;
7 Derive the MCP of the MSSE
8 Order the MSSEs according to their MCPs;
9 **while** *the approximation result is not satisfactory for the specific use case* **do**
10 Select the next MSSE in ascending MCP order;
11 Design a coarse solution predictor for this MSSE;
12 Define the approximation parameters Γ;
13 Define a set i of Γ values;
14 Evaluate the resulting cost and quality (C_i, Q_i) on $\mathcal{D}$;
15 Extract the set f of Γ configurations on the Pareto front;
16 **if** (C_f, Q_f) *is not satisfactory and* i *can be refined* **then**
17 Redefine a set i of Γ values;
18 Go back to line 14;

of this branch is higher than the best cost which has already been obtained during the exploration of the previous branches, as illustrated in Figure 3. The efficiency of this technique is based on the availability of a heuristic that quickly finds a good solution. This technique can guarantee that the optimal solution S_{opt} is found, even though the search space is pruned. However, the drawback of branch-and-bound techniques is the unpredictability of their execution time [13]. Thus, the gain in terms of energy can not be predicted or adjusted with parameters controlling the approximation.

The Smart Search Space Reduction (SSSR) techniques first select a subset of initial solutions in the search space, based on a coarse estimation of their cost, called prediction. Then a refinement of selected initial solutions is computed to find the best solution among them, as depicted in Figure 4. An efficient coarse solution predictor providing a good estimation with low computational complexity can improve the quality of such solutions. With SSSR techniques applied to energy minimization, the gain in terms of energy can be controlled by adjusting the search space around the coarse estimation. Nevertheless, optimality can not be ensured which depends on the accuracy of the prediction step. This SSSR second category of search space reduction technique is investigated in this paper.

3.2. The Smart Search Space Reduction (SSSR) Method

As explained previously, the branch-and-bound techniques aim to converge as fast as possible towards the best solution in a search space, with the objective to reduce the search cost. However, contrary to the SSSR design method, they do not offer features to control the search cost. Contrary to branch-and-bound techniques, the new SSSR design methodology detailed in this paper is control-oriented and lets the system, at run time, choose a searching methods in a set of solutions ranging from a minimal search to a full search.

The new SSSR design method for applications based on MSSEs algorithms is depicted in Figure 5 and formalized by Algorithm 1. It consists of the 3 first steps performed at design time to manage the approximation at the run time, as detailed below. The goal of the first step is to identify the MSSEs in the application. Then, these MSSEs are classified according to their potential cost gain. The third step consists of designing a predictor to manage the approximation and to reduce the cost of the MSSE according to an acceptable quality degradation. In the fourth step, a run-time management of the approximation is set-up.

3.2.1. Step 1: MSSE Identification

In the first step of the Smart Search Space Reduction (SSSR) method, the developer must manually identify the MSSE algorithms in the application $\mathcal{A}$. MSSEs can be independent or nested. The nesting of MSSE algorithms in the application has no consequence on the methodology, each MSSE being studied independently.

3.2.2. Step 2: MSSE Classification

The second step aims at ordering the identified MSSEs according to the cost reduction opportunity they offer. The cost metric, defined by the developer, can be energy, complexity, execution time, or a combination of several of these metrics. It is an implementation metric to optimize. The nature of quality metric, also defined by the developer, depends on the application domain and purpose. Quality metric may be a Bit Error Rate (BER) for a telecommunication application, a Signal-to-Noise Ratio (SNR) for a signal processing application, or a bitrate/quality ratio for a video compression application.

Depending on cost and quality metrics, different Pareto curves will be obtained, offering trade offs between cost and quality. Thus, the reduction of the search space will not lead to the same application configurations. Let the Minimal Cost Point (MCP), associated to an MSSE algorithm, be the theoretical lower bound of the implementation cost (e.g. the minimal energy) that enables the optimal quality (e.g. the optimal bitrate). Let C_{MCP} be the cost that can be obtained if the optimal solution is able to be perfectly predicted. In this case of optimal prediction, only one solution is computed, the optimal one. It leads to the minimum cost for an optimal quality.

The MCP is obtained by a two-pass approach. The first pass is an exhaustive search identifying the optimal solution in terms of quality over a representative training data set $\mathcal{D}$. This exhaustive search has two objectives: obtaining the worst case cost by exhaustive search, and the optimal solution over the training set $\mathcal{D}$ that will be used in the second pass. The exhaustive search reveals at design time a solution close to the optimal solution (under the hypothesis that the training data set $\mathcal{D}$ is representative of the run time processed data) that will be approached at run time by the Pareto prediction. In the second pass, the MSSE algorithm only executes the "optimal over training data" solution identified in the previous pass. The C_{MCP} is the cost of this second pass. The cost reduction opportunity is then defined by the C_{MCP} from the cost of the exhaustive search.

The output of this MSSE Classification step is $\mathcal{L}_{msse}$, the ordered list of MSSEs according to the cost reduction opportunities.

3.2.3. Step 3: MSSE Approximation

The approximation process is carried out for each MSSE algorithm of the ordered list $\mathcal{L}_{msse}$ until sufficient cost reduction is obtained. This process starts with the MSSE algorithms having the highest opportunity in terms of cost reduction and progressively the approximations associated to each MSSE algorithm can be combined.

Substep: 3.1: Coarse Solution Predictor Design. In this step, the developer has to design and develop an efficient coarse solution predictor. The challenge of this step is to define a predictor model with a moderate computation complexity overhead and able to provide a solution as close as possible to the optimal solution. In the context of approximate computing, coarse solution predictors with a limited computation complexity are preferred to precise and expensive cost reduction techniques in order to reduce significantly the implementation cost. Complex coarse solution predictors can annihilate the cost reduction obtained by the search space reduction. Moreover, a complex predictor solution requires both long design and development times. Let C_{CP} and Q_{CP} be respectively the normalized computation cost of Coarse Predictor (CP) and the quality degradation associated to this configuration. The difference between C_{MCP} and C_{CP} is likely to be mainly due to the overhead of the Coarse Predictor computation.

Substep 3.2: Approximation Management. This sub-step aims at expanding the search space around the solution obtained with the coarse solution predictor. Expanding the search space improves the quality because it increases the probability to include the optimal solution in the set of tested solutions. Nevertheless, this expansion is performed at the expense of some implementation cost. Firstly, the different approximation parameters Γ involved in exploring the cost-quality trade-off are enumerated and a set of parameter values i is defined. These approximation parameters Γ serve the coarse solution predictor and define how large the search space is from the coarse estimation. Secondly, a fast quality evaluation approach is used to extract the configurations that are close to the multi-objective Pareto front. This fast approach is used to quickly remove the configurations (C_f, Q_f) in all the configuration tested (C_i, Q_i) which are far from the Pareto front. The quality is evaluated on a subset of the approximation parameters Γ values to select the values leading to configurations (C_f, Q_f) close to the Pareto front. The fast quality evaluation is carried-out by limiting the amount of processed input data. This leads to a statistical estimation with a moderate accuracy, yet sufficient to detect configurations (C_f, Q_f) close to the Pareto front. Thirdly, the most interesting configurations of Γ values are refined by testing more approximation parameters values on the complete input data set.

At the end of this step, the developer can choose to go back at the beginning of the Step 3 to combine another MSSE in the approximation. Three main reasons can bring the developer to work on multiple MSSEs in an energy oriented use case:

- if the developer has energy constraints not achievable with one MSSE alone,
- if the developer wants to have a fine grain management of the energy to match an energy budget,
- if the developer wants to achieve the best possible quality for a given energy reduction.

3.2.4. Run-time Approximation Management

The aim of the last step is to design and implement the run-time management of the approximation. This last part is out of scope of this paper. This controller defines the strategy to control the approximation parameters at run-time. It determines the best parameter value configuration according to the requirements in terms of quality and cost. These parameter values are determined from the Pareto front obtained in the previous step. To measure the global gain in terms of,e.g. energy, the quality/energy trade-off is evaluated on a real scenario. The design of the run-time manager is not in the scope of this paper.

To conclude, by using the SSSR methodology, a designer follows a straightforward flow that exploits the computation-skip approximate computing concept to explore the trade-offs between algorithm degradation and computational cost reduction. The identification of the MSSE parts of the algorithm is the only prerequisite to apply the methodology. SSSR requires manual design steps that complicate the design process. However, they are the price to pay for obtaining large energy gains. Future works include an evaluation of the Design Productivity of the SSSR methodology.

*3.3. Application Example of S*SSR

The rest of this paper illustrates the MSSE concept on a video compression use case. The same methodology could be applied for instance to the stereo vision application from [26, 28] where a MSSE is present. Stereo matching algorithms aim at creating 3D measurement from two 2D images. Stereo matching algorithm measures the similarity of pixels from the two input images to deduce the disparity level for each pixel of the images. Local [26] and semi-local [28] methods perform their searches on a limiting number of candidate pixels which constitutes a MSSE algorithm. Applying our methodology on these Stereo matching algorithm can be briefly summarized as follow:

- design a coarse solution predictor which predicts the best candidate pixel,
- increase the number of tested pixels around the predicted one as approximation parameter,
- measure the degradation of depth map quality according to the number of tested pixels around the predicted one.

Such an additional study is kept as a future work on MSSE management.

4. Case study: The Hevc All Intra Encoder

4.1. Overview of HEVC Encoding

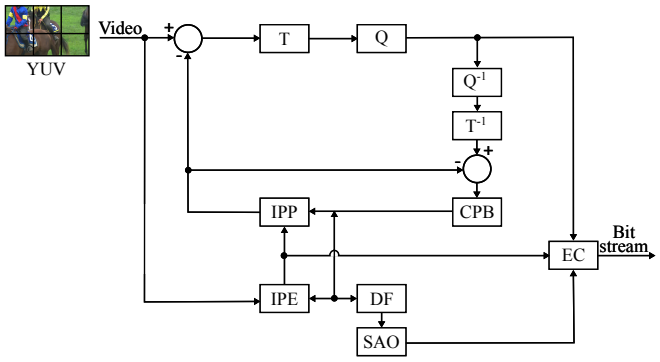

Figure 6: Block diagram of HEVC intra encoder composed by several blocks: Intra Picture Process (IPP), Intra Picture Estimation (IPE), Transform (T), Quantization (Q), Inverse Quantization (Q^{-1}), Inverse Transform (T_{-1}), Current Picture Buffer (CPB), Deblocking Filter (DF), Sample-Adaptive Offset (SAO) and Entropy Coding (EC)

An HEVC encoder is classically based on a *hybrid video encoder* structure that combines Inter and Intra predictions. This work is only focused on Intra encoding, Figure 6 illustrates the block diagram of an HEVC intra encoder. While encoding in HEVC, each frame is split into equally-sized blocks named Coding Tree Units (CTUs) (Figure 7). Each CTU is then divided into Coding Units (CUs), themselves nodes in a quad-tree. In HEVC, the size of CUs is equal to $2N \times 2N$ with $N \in \{32, 16, 8, 4\}$. The HEVC encoder starts by predicting the blocks from their environment (in time and space). To perform the predictions, CUs may be split into Prediction Blocks (PBs) of smaller size. In intra prediction mode, PBs are square and may take the size of $2N \times 2N$ (or $N \times N$ only when $N = 4$). The HEVC intra-frame prediction is complex and supports a total of 35 modes (illustrated on Figure 8) performed at the level of PB including planar (surface fitting) mode, DC (flat) mode and 33 angular modes [39]. Figure 8 shows an example of an intra-prediction with $N \times N$ PB size of 8×8 and the intra-prediction modes. After computing this prediction, the encoder calculates the residuals (prediction error) by subtracting the prediction from the original samples. The residual is then transformed by a linear spatial transform, quantized, and finally entropy coded.

The HEVC encoder also contains a decoder processing loop since the decoded picture is required by the encoder to perform Intra and Inter predictions. This decoder loop is composed of inverse quantization and inverse transform steps that reconstruct the residual information (i.e. the error of the prediction). The residuals are added to the predicted samples to generate a decoded picture (also called reconstructed samples). In the case of Intra encoding, reconstructed samples are stored in the current picture buffer and used for predicting future blocks. Fi-

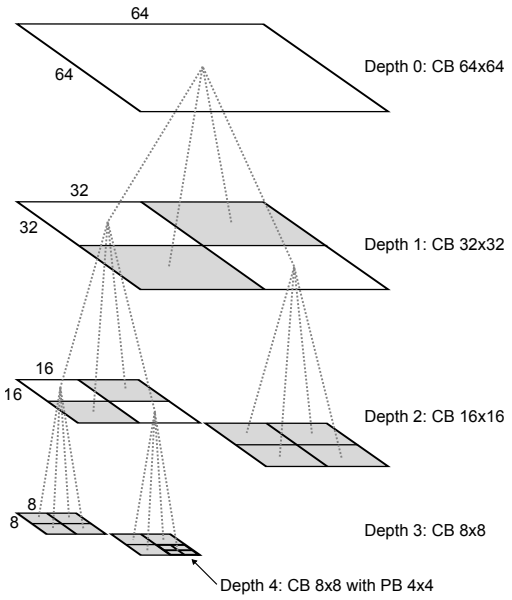

Figure 7: Quad-tree structure of a CTB divided into CBs

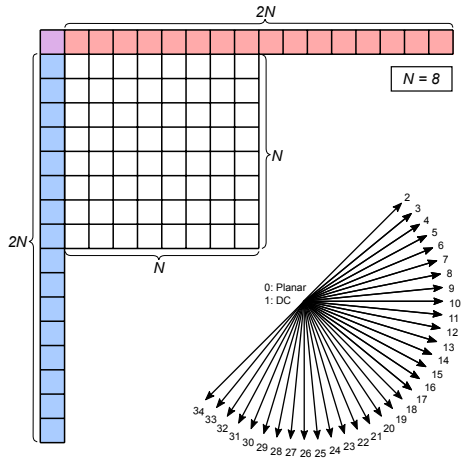

Figure 8: Neighbouring samples used for intra-prediction in an $N \times N$ PB with $N = 8$ and intra-prediction modes

nally, reconstructed samples are post-processed by a deblocking filter and a Sample Adaptive Offset filter (SAO) (used for Inter prediction) that generates the parameters of the decoding filter and appends them to the bitstream. To achieve the best Rate-Distortion (RD) performance, the encoder performs an exhaustive search process, named Rate-Distortion Optimization (RDO), testing every possible combination of partitioning structures combined with the 35 Intra prediction modes. This exhaustive search constitutes an MSSE algorithm.

In order to decrease the computational complexity of HEVC Intra encoding, a fast intra mode decision called Rough Mode Decision (RMD) [44, 45] was added in the reference software HEVC test Model (HM) [15]. This technique splits the Intra prediction process into two successive steps: RMD and RDO as illustrated in Figure 9. RMD consists in constructing a candidate mode list which is then tested in the full RDO process. RMD method computes for each mode m a cost $J_{RMD}(m)$ described by Equation 1, where D_P is the sum of absolute values of the Hadamard transformed coefficients of the prediction residue, λ_P is a Lagrangian multiplier (which depends of the Quantization Parameter (QP)) and R_P is the number of bits necessary to encode the prediction mode information.

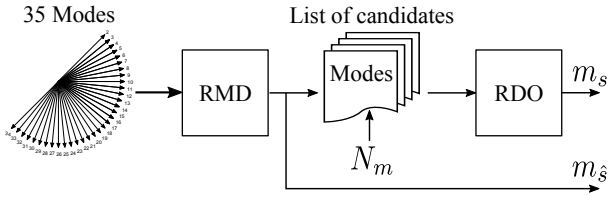

Figure 9: Intra prediction steps

$$J_{RMD}(m) = D_P + \lambda_P \cdot R_P \quad (1)$$

R_P is constant and equal for all modes different from the three Most Probable Modes (MPMs), which require lower signalling (rate). The N_m modes with the lowest costs $J_{RMD}(m)$ are then evaluated by the full RDO process to select the best among them. N_m depends on the CU size N and is defined by Equation 2. The RDO step is much more complex than the RMD step. As the RMD step orders the modes according to their costs, the RDO step can be skipped to limit the encoding complexity. In this work, only the RMD step is applied and the mode $m_{\hat{s}}$ with the smallest cost $J_{RMD}(m)$ is selected.

$$N_m \begin{cases} 8, N \in \{4, 8\} \\ 3, N \in \{16, 32\} \\ 3, N \in \{64\} \end{cases} \quad (2)$$

4.2. Related Work on Methodologies

Several approaches have been proposed to reduce the complexity by shrinking search spaces of HEVC encoder specific part. In [14], authors propose a methodology to optimize the coding mode evaluation (between Intra and Inter prediction according to the Prediction Unit (PU) partitioning of the image). The modes are ordered according to the order of appearance. The modes are tested one by one according to the previous order but the next mode is only tested if the previous one has improved the obtained cost. This methodology exploits the concept of early-termination (detailed in Section 3.1) and is not adapted to control the energy consumption.

For Inter prediction, the motion estimation process uses Block Matching algorithms to determine the motion vectors representing the displacement of a block between a reference frame and the current frame. Many algorithms were proposed for motion estimation over the last 20 years, as for example *Three Step Search*, *New Three Step*

Table 1: Energy reduction opportunities (in J) [25]

Res.	Energy for exhaustive search	Energy for Mcp		Reduction		Reduction (in %)	
		IM	CT	IM	CT	IM	CT
2k	9710	7438	3398	2272	6311	23.4	65.0
1080p	4813	3663	1560	1150	3253	23.9	67.6
720p	2204	1722	911	483	1294	21.9	58.7
480p	1120	833	317	287	803	25.6	71.7
240p	291	209	69	81	222	27.9	76.3
				Average		**24.5**	**67.9**

Search [20], *Four Step Search* [31], *Diamond Search* [46], etc. These techniques reduce the number of tested blocks when compared to a full search. They all rely on a branch-and-bound approach and, because they are iterative and the number of blocks tested is not predictable, they can not, contrary to our proposed method, be used to predict the complexity of this part of the Inter prediction process.

Contrary to our propose methodology in this paper, these techniques do not allow to control the energy consumption from the minimal search to a full search.

4.3. Experimental Setup

All experimentations are performed on one core of the embedded *EmETXe-i87M0* platform from *Arbor Technologies* based on an Intel Core i5-4402E processor at 1.6 GHz. The studied Hevc software encoder is Kvazaar [17, 18, 41] in AI configuration. Each of the tested configurations is used to encode 100 frames of 4 high-resolution (1080p) reference video sequences: *Cactus*, *BasketballDrive*, *BQTerrace* and *ParkScene* with QPs 22, 27, 32, 37 [4].

4.3.1. Cost Metric

The energy consumption is used as the optimized cost in the rest of this study. To measure the energy consumed by the platform, Intel Running Average Power Limit (Rapl) interfaces are used to get the energy of the CPU package, which includes cores, IOs, DRAM and integrated graphic chipset. As shown in [12], Rapl power measurements are coherent with external measurements and [11] proves the reliability of this internal measure across various applications. In this work, only the power gap between IDLE state and video encoding is measured. The CPU is considered to be in IDLE state when it spends more than 90% of its time in the C7 C-states mode. The C7 state is the deepest C-state of the CPU characterized by all core caches being flushed, the PLL and core clock being turned off as well as all uncore domains.

4.3.2. Quality Metric

Bjøntegaard Delta Bit Rate (BD-BR) and Bjøntegaard Delta Psnr (BD-Psnr) [43] is commonly used in video compression to measure the compression efficiency difference between two encodings. The BD-BR reports the average bit rate difference in percent for two encoding at the same quality: Peak Signal-to-Noise Ratio (Psnr). Similarly, the BD-Psnr measure the average Psnr difference in decibels (dB) for two different encoding algorithms considering the same bit rate. In the rest of this work, BD-BR is used as the quality metric where positive BD-BR values correspond to BD-BR loss.

*4.4. Experimental Results on Applying S*ssr *to the Case Study*

*4.4.1. M*sse *Algorithm Identification*

In Hevc Intra encoding, the selections of Rate-Distortion (RD)-wise best PB size and Intra prediction mode are determined by the RDO process. The RDO process is composed of two nested Msses: Coding-tree partitioning (CT) and Intra-mode prediction (IM). Coding-tree partitioning (CT) aims at finding the best quad-tree decomposition of a CTU of 64x64 pixels into CUs as illustrated in Figure 7. Then, for all CUs, Intra-mode prediction (IM) aims at finding the best mode to predict blocks from its neighbors.

*4.4.2. M*sse *Classification*

In this work, an energy metric is used to classify and evaluate the Msses. We define the theoretical lower bound of the energy consumption called in the methodology Mcp(CT) and Mcp(IM) for the two Msses of the RDO process: respectively Coding-tree partitioning (CT) and Intra-mode prediction (IM). The Mcp is the energy obtained when the encoder is able to perfectly predict the best partitioning solution and thus only the optimal solution is processed to encode the CTU [25]. Therefore, the energy consumption of the search process is reduced to the energy consumption of the solution and the Mcp is the minimal energy consumption point that can be achieved for the highest encoding quality.

Table 1 summarizes the energy reduction opportunities between optimal (best complexity case) and full search (worst case) solutions at different video resolutions. The

results are extracted from [25]. They are obtained by applying the two-pass approach as defined in the MSSE Classification Step 2 presented in Section 3.2.2. The results show that the search space is similar across all resolutions and the largest energy reduction search space occurs when optimizing the *Coding-tree partitioning*, with up to 76.3% of potential energy reduction while working on the *Intra-mode prediction* offers 27.9% at best. The results lead to the conclusion that the energy problematic can be more efficiently addressed by reducing complexity at the *Coding-tree partitioning*.

4.4.3. Coding-Tree Partitioning MSSE Approximation

Coarse Solution Predictor Design. In this case, the coarse solution predictor aims to predict the coding-tree partitioning from video frame content. Authors of [16, 36] show the relationship between CU size and the corresponding block variance of the image. Based on this observation, they propose a variance-aware coding-tree prediction. The energy reduction technique used in this paper follows a similar algorithm. A video sequence is split into equal Groups of Frames (GOF) of size F. The first frame of a GOF is encoded with a full RDO process (unconstrained in terms of energy). Then the variance of the selected CUs according their sizes are used to compute variance thresholds *on-the-fly*. For following frames of the GOF, the variance of each CU of each size are recursively compared to the thresholds to choose if the CU has to be split. The coding-tree partitioning is built by this process.

Approximation Management. The first parameter that impacts the encoding quality and energy consumption is the number of frames F in the GOF. The second parameter N_d defines the number of depth values tested around the prediction for each constrained CTU [24]. Since applying the RDO process on the predicted depth map is the result of a coarse estimation, it is possible, without compromising too much the complexity, to improve the process by exploring more depths around the predicted optimum.

Table 2: First set of parameters to explore CT MSSE

Parameter	Values
F	1, 2, 4, 8, 16, 32, 50
N_d	1, 2, 3, 4

Since video encoding is time consuming, a fast quality evaluation approach with a restricted parameter set is used to extract the configurations close to the Pareto front. Table 2 summaries the first set of parameters used to explore the trade-off between energy consumption and BD-BR. $F = 1$ represents an encoding without a constrained frame (anchor). The anchor encoding is used to normalize the energy consumption (upper bound) and compute BD-BRs.

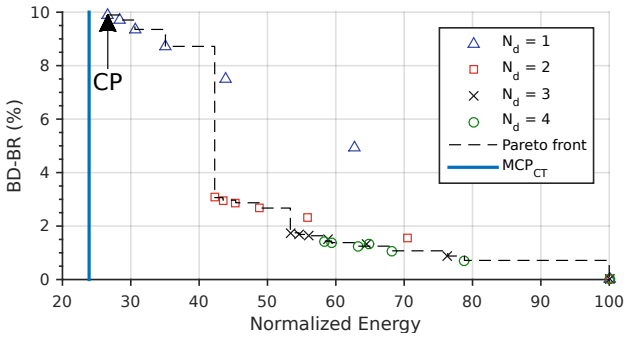

Figure 10: Pareto in Rate-Energy space from the first set of parameters defined in Table 2 (exploring the CT MSSE)

Quality & Cost Evaluation. Figure 10 shows the Rate-Energy space of all the combinations of parameters defined in Table 2. Identical markers correspond to different values of F for a fixed value of N_d. As shown in Figure 10, the coarse solution predictor is able to predict a solution close to the MCP(CT) in term of energy; the difference between $C_{\text{MCP(CT)}}$ and C_{CP} is around 2% of energy. The results also show that the relation between energy consumption and BD-BR seems close to linear compared to F for a fixed value of N_d. Nevertheless, a significant gap in term of BD-BR divides points between $N_d = 1$ and $N_d = 2$. This observation requires to refine N_d and to use non-integer values. To explore non-integer numbers of depths, CTUs in a constrained frame are split into two categories [24]: $(N_d - \lfloor N_d \rfloor) \times 100$ per cent of CTUs are encoded with $\lceil N_d \rceil$ depths and the rest with $\lfloor N_d \rfloor$ depths.

Table 3 summarizes the second set of parameters used to explore the trade-off between energy consumption and quality. Figure 11 shows the Rate-Energy space for all the combinations of parameters defined in Table 3. This figure shows that for normalized energy reduction of up to 60%(below 40% in Figure 11), the points of the Pareto front are generated with a high value of F and a low value of N_d. On the other hand, for normalized energy reductions of less than 40% (higher than 60% in Figure 11) the configurations are obtained with $F = 2$ and a high value of N_d. The encoder has to play on both F and N_d parameters respectively the size of the GOF and the number of explored depths to control the energy consumption of the HEVC encoder.

Table 3: Second set of parameters to explore CT MSSE

Parameter	Values
F	1, 2, 4, 8, 12, 32, 50
N_d	0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.2, 1.4, 1.6, 1.8, 2, 2.5, 3 ,3.5

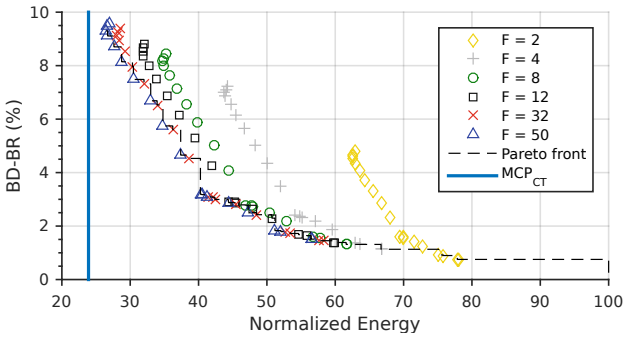

Figure 11: Pareto in Rate-Energy space from the second set of parameters defined in Table 3 (exploring the CT Msse)

4.4.4. Intra-Mode Prediction Msse Approximation

Coarse Solution Predictor Design. The Kvazaar encoder includes a feature that reduces the computational complexity of RMD. This feature reduces the number of angular prediction modes candidates and is divided in two successive steps illustrated in Figure 12:

1. Coarse Step: for each PU $N \times N$, the number of angular modes tested in RMD is reduced by increasing the angular step-size (θ_N). Lets Θ be the set of $(\theta_{32}, \theta_{16}, \theta_8, \theta_4)$. In Kvazaar: Θ is fixed to $(8, 8, 4, 2)$. The Coarse Step always test DC, Planar and MPM modes.
2. Refinement Step: the goal of this step is to refine the dominant prediction direction $m_{\hat{s}}$ obtained from the previous step. The angular step size is reduced by half $\theta'_N = \frac{\theta_N}{2}$ and the RMD process computes the cost $J_{RMD}(m_{\hat{s}} \pm \theta'_N)$ of the direction around the prediction mode obtained from the previous step. This step is repeated with the new dominant direction until the angular step size becomes 1.

Figure 12 illustrates the process for $\theta = 8$. In the Coarse step, angular modes $\{2, 10, 18, 26, 34\}$ are tested and the angular mode 10 which lead to the minimal cost is selected. Then, the Refinement step is iterated 3 times. In the first iteration, the angular step size is reduced to 4: modes $\{6, 14\}$ are tested and modes 14 is selected. In the second iteration, the step size is reduced to 2: modes $\{12, 16\}$ are tested and modes 16 is selected. In the last iteration, the step size is reduce to 1: modes $\{15, 17\}$ are tested and modes 15 is finally selected.

In the reference configuration, this feature is disabled and all modes are tested in RMD process.

Approximation Management. The minimal and maximal number of modes tested by RMD according to θ_N are given respectively by Equations 3 and 4. The number of modes tested by RMD depends on whether the MPMs is already included in the set of modes. The first and second terms of Equations 3 and 4 correspond to the first and

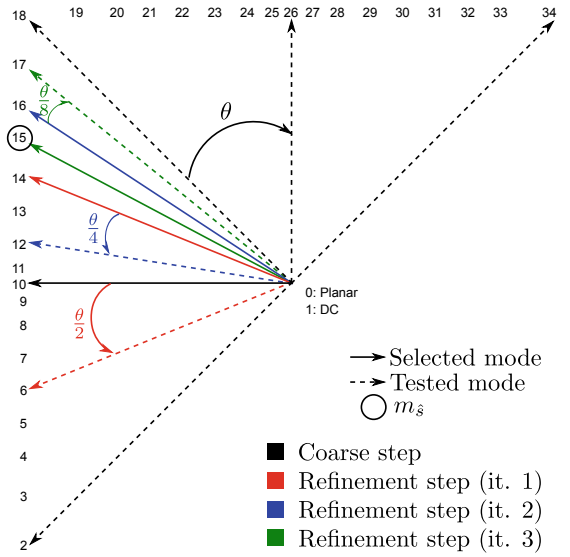

Figure 12: RMD complexity reduction steps description for $\theta = 8$

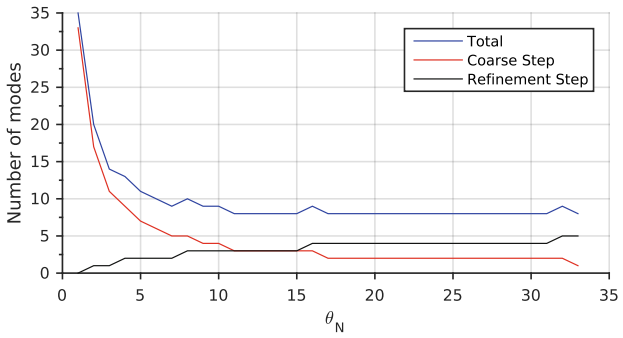

Figure 13: Minimal number of modes tested by the RMD process according to θ_N

second steps of the RMD algorithm while the third term adds the number of no angular modes plus the MPM.

Figure 13 shows the minimal number of modes tested by the RMD process according to θ_N during the two steps and the total as described by Equation 3.

$$min_{mode}(\theta_N) = \left\lceil \frac{33}{\theta_N} \right\rceil + \lfloor \log_2(\theta_N) \rfloor + 2 \quad (3)$$

$$max_{mode}(\theta_N) = \left\lceil \frac{33}{\theta_N} \right\rceil + \lfloor \log_2(\theta_N) \rfloor + 5 \quad (4)$$

To explore the Msse linked to the Intra mode prediction, a set of $\theta_N \in \{2, 4, 8, 12\}$ (corresponding to testing respectively 20, 13, 10 and 8 modes) is defined.

Quality & Cost Evaluation. For $N \in \{32, 16, 8, 4\}$, 4096 encodings are needed to try all combinations of Θ with $\theta_N \in \{2, 4, 8, 12\}$. The number of experimentations is reduced to study the impact of θ_N for each size N of CU independently. The video sequences are encoded with $\theta_N \in \{2, 4, 8, 12\}$ for a fixed value of $N \in \{32, 16, 8, 4\}$ one at a time. The other angular step-sizes are fixed to the

default value of Kvazaar: $\Theta = (8, 8, 4, 2)$.

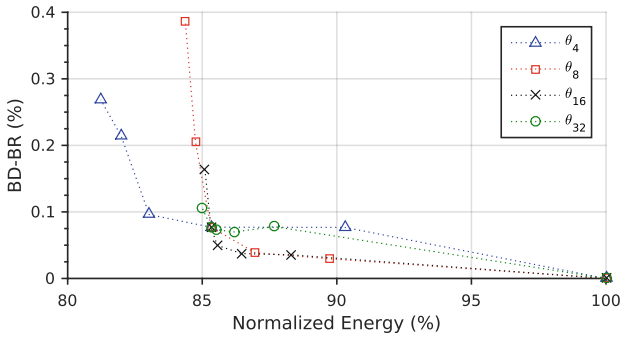

Figure 14: Pareto in Rate-Energy space when $\theta_N \in \{2, 4, 8, 12\}$ for a fixed value of N and other fixed (exploring the IM MSSE)

Figure 14 shows the Rate-Energy space for each CU size N. In other words, Figure 14 shows the impact of θ_N for each value of N independently. Results show that the relation between energy consumption and BD-BR according to θ_N is not linear, and this for all CU sizes. The configurations with bad trade-off between energy reduction and BD-BR increase are then removed to build a new set of θ_N parameters summarized in Table 4.

Table 4: Set of θ_N parameters to explore the IM MSSE

Parameter	θ Values
θ_{32}	2, 4, 8
θ_{16}	2, 4, 8
θ_8	2, 4
θ_4	2, 4

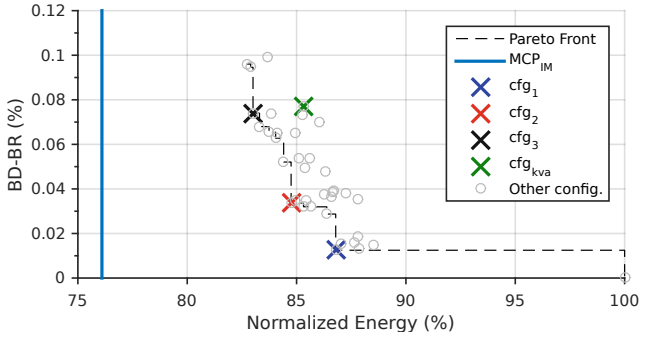

Figure 15: Pareto in Rate-Energy space generated from all $3 \times 3 \times 2 \times 2 = 36$ combinations of parameter values defined in Table 4 (exploring the IM MSSE)

Figure 15 shows the results of the 36 configurations defined by the Table 4. The difference between the energy reduction opportunities $C_{\text{MCP(IM)}}$ and C_{CP} of the Intra-Mode Prediction MSSE is around 5% of energy. Figure 15 shows that for the set $(\theta_{32}, \theta_{16}, \theta_8, \theta_4)$, a better configuration than the Kvazaar default one $(8, 8, 4, 2)$ (cfg_{kva} in green in Figure 15) can be used for the same energy reduction.

4.4.5. Combination of MSSEs

The goal of this part is to study the combination of the two MSSEs: Coding-tree partitioning (CT) and Intra-mode prediction (IM). As explained in Section 4.4.3, the MSSE linked to the CT can be explored with two parameters F and N_d. From results of Figure 11, the configuration of the parameters F and N_d of the Pareto Front are extracted.

The MSSE linked to the IM depends on a set of θ_N which is viewed as one parameter to combine the MSSEs. In addition to the Kvazaar default configuration (cfg_{kva}), 3 other configurations (cfg_1, cfg_2, cfg_3) are extracted from results of the Figure 15 which correspond to significant gap in the Pareto front. Table 5 summarizes the configurations.

Table 5: Configurations extracted from results of the IM MSSE analysis

Configuration	Θ
cfg_{kva}	$(8, 8, 4, 2)$
cfg_1	$(8, 4, 2, 2)$
cfg_2	$(8, 4, 2, 4)$
cfg_3	$(8, 4, 4, 4)$

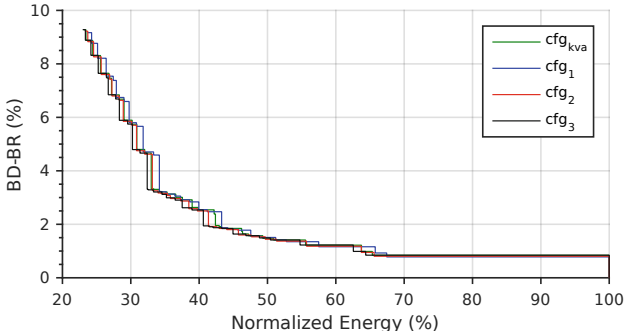

Figure 16: Pareto in Rate-Energy space from the set of Θ defined in Table 5 (exploring the combination of the CT and IM MSSEs)

Figure 16 shows the results when the configurations described in Table 5 are applied on the configurations extracted from the front of the Rate-Energy space of the Figure 11. From 100% to 45% of normalized energy consumed, the results of the 4 configurations are interwined. In the other hand, for an energy consumed less than 45%, the cfg_3 have better results for a major part of the Rate-Energy space. As for the CT MSSE, Figure 16 shows that it is possible to control the energy consumed from 100% to 23%. The results of Figure 15 are finally added in the Rate-Energy space as shown in Figure 17.

Figure 17 summarizes the best results (extracted from the Pareto front) of the CT MSSE study of the Section 4.4.3, the IM MSSE study of the Section 4.4.4 and the

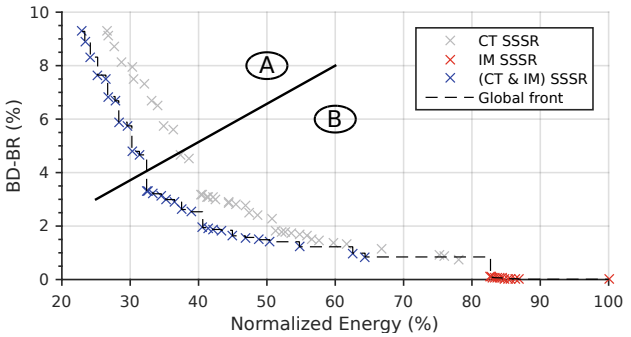

Figure 17: Pareto in Rate-Energy space from the CT Msse, the IM Msse and the combination of the two Msses: CT & IM

combination of these two Msses CT & IM, i.e. when the three parameters F, N_d and Θ are used.

Figure 17 shows that for all normalized energy target, the combination of the two Msses (CT & IM) obtains better results than the exploration of the CT Msse alone. For example, for 32.5% of normalized energy consumed, the combination of the two Sssrs compared to the case of CT alone reduces the BD-BR by 4%: from 7.3% to 3.3%. Figure 17 shows that the Pareto front has an inflection point (illustrated by the black line in Figure 17). This inflection point splits the Pareto front into two parts (A and B). In part A, a normalized energy reduction of up to 23% of energy consumed has a strong impact on the quality. In the other hand, in part B, the quality degradation is less impacted when the consumed energy is reduced.

To conclude on these results, playing with the two Msses of the HEVC use case has been demonstrated to yield better energetic results than just using one Msse, and the Sssr methodology has been shown to give precise answers on the opportunities of gain brought by each Msses. These results motivate for the Sssr methodology that provides a systematic mechanism to explore and evaluate the approximation opportunities of Msse-based applications.

On the considered use case, inflexion points on the Pareto curves guide the designer when choosing the right configuration that does not suffer significantly from a low of diminishing return. This is the case for instance in Figure 17 where a designer is advised to target the left-hand side of region B where energy gains are relatively high and BD-BR losses are low.

5. Conclusion

This paper has proposed a new methodology to exploit the computation-skip approximate computing concept by using an Sssrs technique to explore the trade-offs between degradation and cost reduction in applications with Minimization based on Search Space Exploration (Msse). The methodology is applied to an Hevc Intra video encoder. By using Sssr on the two discrete optimization loops of this use case (Coding-tree partitioning (CT) and Intra-mode prediction (IM)), energy consumption has been reduced by up to 68% with a degradation of 3.8% of BD-BR (as shown at the left-hand side of region B in Figure 17) and up to 77% with a degradation of 9.2% of BD-BR (as shown at the left-hand side of region A in Figure 17). Moreover, the combination of the two Sssrs in comparison to using CT alone reduces the BD-BR from 7.3% to 3.3% for the same energy consumption of 32.5%. Future work will use this methodology and its results to implement the last step of run-time approximation management and control the energy consumption of an Hevc Intra encoder for a given energy consumption budget.

Acknowledgments

This work is partially supported by the French ANR ARTEFaCT project, by the COVIBE project funded by the Brittany region and by the European Celtic-Plus project 4KREPROSYS funded by Finland, Flanders, France, and Switzerland.

References

[1] Benini, L., De Micheli, G., 1998. Dynamic Power Management. Springer US, Boston, MA.

[2] Bienia, C., Kumar, S., Singh, J. P., Li, K., 2008. The PARSEC benchmark suite: characterization and architectural implications. In: Proceedings of the 17th international conference on Parallel architectures and compilation techniques. ACM, pp. 72–81.

[3] Bonnot, J., Nogues, E., Menard, D., 2016. New Non-Uniform Segmentation Technique for Software Function Evaluation.

[4] Bossen, F., 2013. Common HM test conditions and software reference configurations. In: JCTVC-L1100. Geneva, Switzerland.

[5] Camus, V., Schlachter, J., Enz, C., 2015. Energy-efficient inexact speculative adder with high performance and accuracy control. In: 2015 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, pp. 45–48.

[6] Camus, V., Schlachter, J., Enz, C., 2016. A low-power *carry cut-back* approximate adder with fixed-point implementation and floating-point precision. ACM Press, pp. 1–6.

[7] Carbin, M., Misailovic, S., Rinard, M. C., 2013. Verifying quantitative reliability for programs that execute on unreliable hardware. ACM Press, pp. 33–52.

[8] Chippa, V. K., Mohapatra, D., Raghunathan, A., Roy, K., Chakradhar, S. T., 2010. Scalable effort hardware design: exploiting algorithmic resilience for energy efficiency. In: Proceedings of the 47th Design Automation Conference. ACM, pp. 555–560.

[9] De Dinechin, F., Tisserand, A., 2005. Multipartite table methods. IEEE Transactions on Computers 54 (3), 319–330.

[10] Du, K., Varman, P., Mohanram, K., 2012. High performance reliable variable latency carry select addition. In: 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, pp. 1257–1262.

[11] Efraim, R., Alon, N., Doron, R., Avinash, A., Eliezer, W., Apr. 2012. Power-Management Architecture of the Intel Microarchitecture Code-Named Sandy Bridge. IEEE Computer Society 32 (2), 20–27.

[12] Hackenberg, D., Schone, R., Ilsche, T., Molka, D., Schuchart, J., Geyer, R., May 2015. An Energy Efficiency Feature Survey of the Intel Haswell Processor. In: Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International. IEEE, pp. 896–904.

[13] Hamidouche, W., Olivier, C., Pousset, Y., Perrine, C., Apr. 2013. Optimal resource allocation for Medium Grain Scalable video transmission over MIMO channels. Journal of Visual Communication and Image Representation 24 (3), 373–387.

[14] Herglotz, C., Rosales, R., Gla\s s, M., Teich, J., Kaup, A., 2016. Multi-objective design space exploration for the optimization of the HEVC mode decision process. In: Picture Coding Symposium (PCS), 2016. IEEE, pp. 1–5.

[15] JCT-VC, 2016. HEVC reference software. https://hevc.hhi.fraunhofer.de/.

[16] Khan, M. U. K., Shafique, M., Henkel, J., 2013. An adaptive complexity reduction scheme with fast prediction unit decision for HEVC intra encoding. In: Image Processing (ICIP), 2013 20th IEEE International Conference on. IEEE, pp. 1578–1582.

[17] Koivula, A., Viitanen, M., Lemmetti, A., Vanne, J., Hämäläinen, T. D., 2015. Performance evaluation of Kvazaar HEVC intra encoder on Xeon Phi many-core processor. In: Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on. IEEE, pp. 1250–1254.

[18] Koivula, A., Viitanen, M., Vanne, J., Hamalainen, T. D., Fasnacht, L., 2015. Parallelization of Kvazaar HEVC intra encoder for multi-core processors. In: Signal Processing Systems (SiPS), 2015 IEEE Workshop on. IEEE, pp. 1–6.

[19] Krause, P. K., Polian, I., 2011. Adaptive voltage over-scaling for resilient applications. In: 2011 Design, Automation & Test in Europe. IEEE, pp. 1–6.

[20] Li, R., Zeng, B., Liou, M. L., 1994. A new three-step search algorithm for block motion estimation. IEEE transactions on circuits and systems for video technology 4 (4), 438–442.

[21] Lingamneni, A., Enz, C., Nagel, J.-L., Palem, K., Piguet, C., 2011. Energy parsimonious circuit design through probabilistic pruning. In: 2011 Design, Automation & Test in Europe. IEEE, pp. 1–6.

[22] Ludwig, J., Nawab, S., Chandrakasan, A., 1996. Low-power digital filtering using approximate processing. IEEE Journal of Solid-State Circuits 31 (3), 395–400.

[23] Macken, P., Degrauwe, M., Van Paemel, M., Oguey, H., Feb. 1990. A voltage reduction technique for digital systems. p. 29.

[24] Mercat, A., Arrestier, F., Hamidouche, W., Pelcat, M., Menard, D., 2017. Constrain the Docile CTUs: an In-Frame Complexity Allocator for HEVC Intra Encoders. In: Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE.

[25] Mercat, A., Arrestier, F., Hamidouche, W., Pelcat, M., Menard, D., 2017. Energy Reduction Opportunities in an HEVC Real-Time Encoder. In: Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, pp. 1158–1162.

[26] Mercat, A., Nezan, J.-F., Menard, D., Zhang, J., 2014. Implementation of a stereo matching algorithm onto a manycore embedded system. In: Circuits and Systems (ISCAS), 2014 IEEE International Symposium on. IEEE, pp. 1296–1299.

[27] Mohapatra, D., Chippa, V. K., Raghunathan, A., Roy, K., 2011. Design of voltage-scalable meta-functions for approximate computing. In: 2011 Design, Automation & Test in Europe. IEEE, pp. 1–6.

[28] Nezan, J.-F., Mercat, A., Delmas, P., Gimelfarb, G., Feb. 2016. Optimized Belief Propagation Algorithm onto Embedded Multi and Many-Core Systems for Stereo Matching. IEEE, pp. 332–336.

[29] Nguyen, B.-y., Mazure, C., Delprat, D., Aulnette, C., Daval, N., Andrieu, F., Faynot, O., 2009. Overview of FDSOI technology from substrate to device. Semiconductor Device Research Symposium, 2009. ISDRS '09. International, 1–2.

[30] Nogues, E., Menard, D., Pelcat, M., 2016. Algorithmic-level Approximate Computing Applied to Energy Efficient HEVC Decoding. IEEE Transactions on Emerging Topics in Computing, 1–1.

[31] Po, L.-M., Ma, W.-C., 1996. A novel four-step search algorithm for fast block motion estimation. IEEE transactions on circuits and systems for video technology 6 (3), 313–317.

[32] Renganarayana, L., Srinivasan, V., Nair, R., Prener, D., 2012. Programming with relaxed synchronization. In: Proceedings of the 2012 ACM workshop on Relaxing synchronization for multicore and manycore scalability. ACM, pp. 41–50.

[33] Sachid, A. B., Francis, R., Baghini, M. S., Sharma, D. K., Bach, K.-H., Mahnkopf, R., Rao, V. R., 2008. Sub-20 nm gate length FinFET design: Can high-k spacers make a difference? In: 2008 IEEE International Electron Devices Meeting. IEEE, pp. 1–4.

[34] Sampson, A., Dietl, W., Fortuna, E., Gnanapragasam, D., Ceze, L., Grossman, D., 2011. EnerJ: Approximate data types for safe and general low-power computation. In: ACM SIGPLAN Notices. Vol. 46. ACM, pp. 164–174.

[35] Schlachter, J., Camus, V., Enz, C., Palem, K. V., 2015. Automatic generation of inexact digital circuits by gate-level pruning. In: 2015 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, pp. 173–176.

[36] Shafique, M., Henkel, J., 2014. Low power design of the next-generation high efficiency video coding. In: Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific. IEEE, pp. 274–281.

[37] Sidiroglou-Douskos, S., Misailovic, S., Hoffmann, H., Rinard, M., 2011. Managing performance vs. accuracy trade-offs with loop perforation. In: Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering. ACM, pp. 124–134.

[38] Sorber, J., Kostadinov, A., Garber, M., Brennan, Berger, E. D., Corner, M. D., 2007. Eon: A Language and Runtime System for Perpetual Systems. Proceedings of the 5th international conference on Embedded networked sensor systems 400, 600.

[39] Sullivan, G. J., Ohm, J.-R., Han, W.-J., Wiegand, T., Dec. 2012. Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Transactions on Circuits and Systems for Video Technology 22 (12), 1649–1668.

[40] Sze, V., Budagavi, M., Sullivan, G. J. (Eds.), 2014. High Efficiency Video Coding (HEVC). Integrated Circuits and Systems. Springer International Publishing, Cham.

[41] Viitanen, M., Koivula, A., Lemmetti, A., Vanne, J., Hamalainen, T. D., 2015. Kvazaar HEVC encoder for efficient intra coding. In: Circuits and Systems (ISCAS), 2015 IEEE International Symposium on. IEEE, pp. 1662–1665.

[42] Volder, J. E., 1959. The CORDIC trigonometric computing technique. IRE Transactions on Electronic Computers (3), 330–334.

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

[44] Yinji, P., Junghye, M., Jiangle, C., Jul. 2010. Encoder improvement of unified intra prediction. Guangzhou.

[45] Zhao, L., Zhang, L., Ma, S., Zhao, D., 2011. Fast mode decision algorithm for intra prediction in HEVC. In: Visual Communications and Image Processing (VCIP), 2011 IEEE. IEEE, pp. 1–4.

[46] Zhu, S., Ma, K.-K., 1997. A new diamond search algorithm for fast block matching motion estimation. In: Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on. Vol. 1. IEEE, pp. 292–296.