

Exploiting Computation Skip to Reduce Energy Consumption by Approximate Computing, an HEVC Encoder Case Study

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

► **To cite this version:**

Alexandre Mercat, Justine Bonnot, Maxime Pelcat, Wassim Hamidouche, Daniel Menard. Exploiting Computation Skip to Reduce Energy Consumption by Approximate Computing, an HEVC Encoder Case Study. Design, Automation and Test in Europe, Mar 2017, Lausanne, Switzerland, France. hal-02136715

HAL Id: hal-02136715

<https://hal.archives-ouvertes.fr/hal-02136715>

Submitted on 22 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Exploiting Computation Skip to Reduce Energy Consumption by Approximate Computing, an HEVC Encoder Case Study

Alexandre Mercat¹, Justine Bonnot¹, Maxime Pelcat^{1,2}, Wassim Hamidouche¹, Daniel Menard¹

¹UBL, INSA Rennes, IETR CNRS UMR 6164, Rennes, France

² Institut Pascal, CNRS UMR 6602, Clermont-Ferrand, France

email: {amercat, jbonnot, mpelcat, whamidou, dmenard}@insa-rennes.fr

Abstract—Approximate computing paradigm provides methods to optimize algorithms with considering both computational accuracy and complexity. This paradigm can be exploited at different levels of abstraction, from technological to application levels. Approximate computing at algorithm level aims at reducing computational complexity by approximating or skipping block functions of the computation. Numerous applications in the signal and image processing domain integrate algorithms based on discrete optimization techniques. These techniques minimize a cost function by exploring the search space. In this paper, a new approach is proposed to exploit the computation-skipping approximate computing concept by using the Smart Search Space Reduction (SSSR) technique. SSSR enables early selection of the best candidate configurations to reduce the search space. An efficient SSSR technique adjusts configuration selectivity to reduce execution complexity while selecting the most suitable functions to skip. The 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 functions use discrete optimization to explore different solutions and select the one leading to the minimal cost in terms of bitrate/quality and computational energy: coding-tree partitioning and intra-mode prediction. By applying SSSR to this use case, energy reductions from 20% to 70% are explored through Pareto in Rate-Energy space.

I. INTRODUCTION

Nowadays, optimizing energy consumption in embedded systems is of primary concern for the design of autonomous devices. The emerging domain of 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 and Fully Depleted Silicon on Insulator (FD-SOI) 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. For modern System-on-Chip, there are two main power management techniques that minimize the energy consumption. By combining clock gating and power gating, Dynamic Power Management (DPM) [1] is used to turn a processing core into a low-power state when it is not in use. To reduce the consumed dynamic power, Dynamic Voltage and Frequency Scaling (DVFS) is used to

reduce both hardware clock frequency and supply voltage until real-time constraints are precisely met.

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 [2] and architecture level while algorithm level has not been widely investigated despite the significant energy gain opportunities. At algorithm level, incremental refinement [3] has been proposed for iterative process and loop perforation [4] has been proposed for repetitive structures. In this paper, we focus on algorithms using discrete optimization techniques which aim at minimizing a cost function by exploring the search space. These optimization techniques are referred to as Minimization based on Search Space Exploration (MSSE) in the rest of the paper. To decrease their energy consumption, the challenge is then to reduce the search space, skipping low value-added computation without sacrificing the application performance.

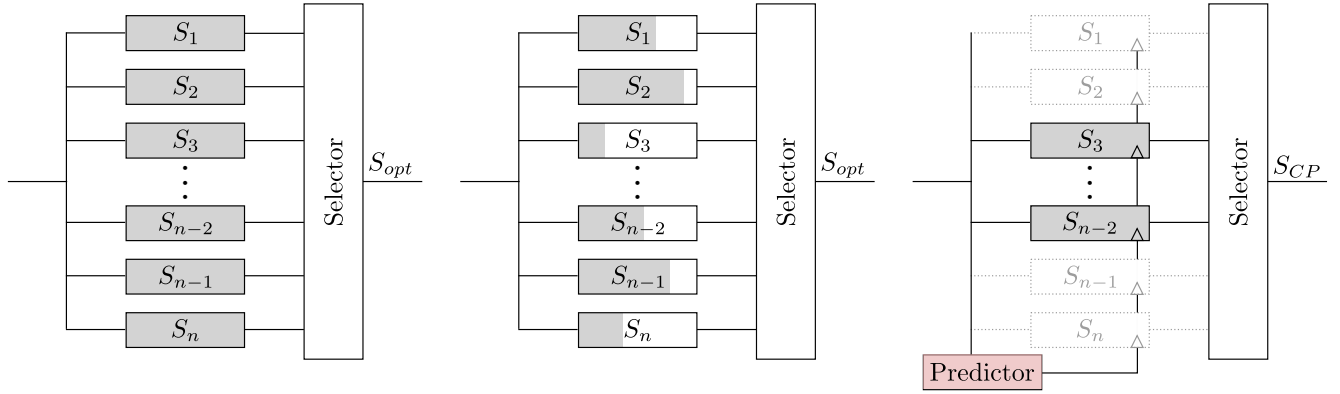
In this paper, a new approach is proposed to exploit the computation-skipping concept by using a Smart Search Space Reduction (SSSR) technique. This technique aims at early selecting the best performing processing functions and reducing the search space. Designing efficient SSSR techniques and adjusting the parameters that balance the complexity reduction and the efficiency of the approximate application is not trivial. A All Intra (AI) profile High Efficiency Video Coding (HEVC) encoder is used as a case study to illustrate the benefits of our approach. The HEVC encoder integrates several algorithms with MSSE. Among these, coding-tree partitioning aims at finding the best decomposition of a block of 64x64 pixels in smaller blocks and intra-mode prediction enables to find the best mode to predict a block from its neighbors.

The rest of the paper is organized as follows. We first provide the related works in Section II, and then we introduce the proposed method in Section III. The method is then applied to analyze the HEVC encoder in Section IV. Finally, Section V concludes the paper.

II. RELATED WORKS

A. Approximate computing at algorithmic level

In order to reduce energy consumption, a few approximate computing techniques have been proposed to decrease the



(a) MSSE algorithm with exhaustive search (b) MSSE algorithm with branch & bound technique. Early termination stop the exploration of branches which can not lead to the best solution (c) MSSE algorithm with SSSR technique. Coarse estimation is carried-out and a refinement is applied to best candidates

Fig. 1. Minimization based on Search Space Exploration (MSSE) algorithm categories

computational complexity or the computational load. On the one hand, the inherent 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, approximation-aware programming languages have been created (for instance Eon [5], EnerJ [6], and Rely [7]). These languages allow the programmer to identify the parts of the code that can tolerate an error and the parts where higher accuracy is required. From this information, the methods presented below can be used.

In [3], the concept of incremental refinement is introduced. The purpose is to reduce the number of iterations of an iterative processing. By skipping part of the processing, the energy can be reduced.

Loop perforation is proposed in [4]. The loops that can tolerate approximation errors are in a first step identified, and then transformed so as to execute a certain number of iterations. This paper proposes Pareto curves for the user to be able to choose the best perforation strategy depending on his requirements. On tunable loops, loop perforation has shown an important reduction in terms on computation time (up to seven times) whereas the accuracy has been decreased to less than 10%.

Another solution to reduce computational complexity consists in identifying blocks that can be skipped with a minimum impact on the output [8]. These blocks can be permanently or periodically skipped depending on the quality constraints of the application. Once a trade-off between the required precision and the energy consumption has been found, a parameter can be used to activate or not the selected blocks. This approximate computing method can be applied to a domain conservation process [8], used to enhance a signal property.

Computation of intricate mathematical functions can be replaced by their approximations. The use of function approximation has the asset of being able to target the required

precision, using well-known algorithms like multipartite tables [9], CORDIC algorithm [10] or polynomial approximation [11].

B. Approximate computing for algorithms with MSSE

In this paper, we focus on algorithms solved by discrete optimization techniques and which aim at minimizing a cost function by exploring a search space. These MSSE algorithms are based on the concept of testing different candidates (S_i) and selecting the best one as illustrated in Figure 1a. Numerous applications in image and signal processing integrate MSSE. In telecommunications, channel decoding and MIMO decoding such as sphere decoding use MSSE algorithms. This concept is also used in image processing for classification operations such as in a Nearest Neighbor classifier and in video processing, for instance for motion estimation. In video compression, as depicted in our case-study presented in section IV, MSSE algorithms are used for coding-tree partitioning and Intra-mode prediction.

An exhaustive search is a straightforward approach to process MSSE but it can require much more computation depending on the search space size. A challenge for an MSSE algorithm implementation is to reduce the search space without decreasing the performance of the optimization solution (S_{opt}). The best solution must be contained in the reduced search space. To the best of our knowledge, no approach has been proposed in literature that focuses on approximate computing for MSSE algorithms.

III. APPROXIMATE COMPUTING METHODOLOGY

A. Search space reduction techniques

For MSSE algorithms, the search space reduction techniques can be classified into two categories. The first one considers initially the entire search space and then incrementally reduces the search space from decisions taken from intermediate results. Parts of the search space which can not lead to the best solution are removed. The discrete optimization problem can be formulated with a tree representing the different solutions

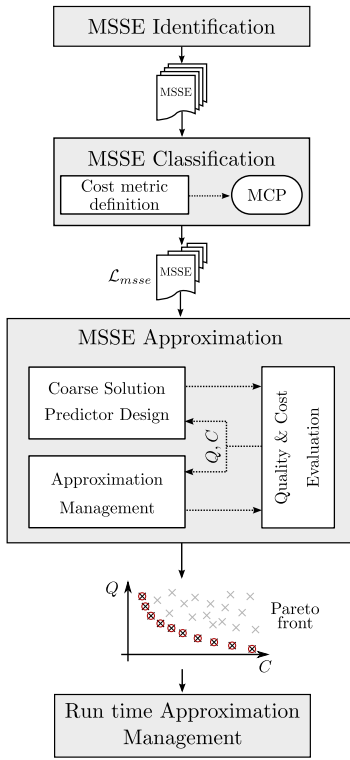


Fig. 2. The Smart Search Space Reduction (SSSR) method

(S_i). The branch and bound strategy can be used to reduce the search space as illustrated in Figure 1b. This strategy exploits the concept of early termination. The exploration of the tree branch is stopped if the minimal cost which can be obtained for the exploration of this branch is higher than the best cost which has already been obtained for the exploration of the rest of branches. The efficiency of this technique is based on the availability of an heuristic that quickly finds a good solution. This kind of technique can ensure to find the optimal solution (S_{opt}) even if the search space is cut-down. However, the drawback of such solutions is the unpredictability of the execution time. Thus the gain in terms of energy can not be predicted or adjusted with parameters controlling the approximation.

The second category is based on a coarse estimation and a refinement of this initial solution to find the best configuration as depicted in Figure 1c. An efficient coarse solution predictor providing a good estimation with low computational complexity can improve the performance of such solutions. With this technique applied for 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. This second category of search space reduction techniques is investigated in this paper.

B. The Smart Search Space Reduction (SSSR) method

The new SSSR design method of approximate computing for applications with MSSEs is depicted in Figure 2. It consists of the steps described below.

1) *MSSE Identification*: In the first step of the Smart Search Space Reduction (SSSR) method, the developer must identify the MSSE algorithms in the application. MSSEs can be independent or interlocked.

2) *MSSE Classification According to Potential Cost Gain*: The second step aims at ordering each MSSE according to the cost reduction opportunity it offers. The cost metric, defined by the developer, can be either energy, complexity, execution time, etc. It is an implementation metric to optimize. The quality metric, also defined by the developer, 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 for a video compression application. It is an application performance metric to optimize.

Depending on the cost and quality metrics, different Pareto curves will be obtained 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 process is able to perfectly predict the optimal solution. In this case of optimal prediction, only one solution is tested, the optimal one. It leads to the minimum cost for optimal quality.

The MCP may be obtained by a two-pass approach. The first pass is an exhaustive search identifying the optimal solution in terms of quality. This exhaustive search has two objectives: obtaining the worst case cost due to exhaustive search, and the optimal solution that will be used in the second pass. In the second pass, the MSSE algorithm only executes the optimal 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) *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.

a) *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 predictor with limited computation complexity are preferred to precise and costly reduction cost techniques in order to reduce significantly the implementation cost. Complex coarse solution predictor can annihilate the cost reduction obtained by the search space reduction. Moreover, complex solution predictor requires both long design and development times. Let C_{CP} and Q_{CP} be respectively the normalized

computation cost of coarse predict (CP) solution and the quality degradation associate to this configuration. The difference between C_{MCP} and C_{CP} is manly due to the over-cost of the coarse predictor computation.

b) Approximation Management: This step aims at expanding the search space around the solution obtained with the coarse solution predictor. Expanding the search space improves the quality because the probability to select the optimal solution is higher. Nevertheless, this is at the expense of the implementation cost. Firstly, the different parameters allowing to explore the cost-quality trade-off are enumerated and the parameter bounds are defined. These parameters are the coarse solution predictor parameters and those which define how the search space is expanded from the coarse estimation. Secondly, a fast quality evaluation approach is used to extract the configurations closed to the Pareto front. This fast approach is used to quickly remove the configurations which are far from the Pareto front. The quality is evaluated on a subset of the parameter values to select the values leading to configurations close to the Pareto front. The fast quality evaluation is carried-out by limiting the amount of input data processed. This leads to a statistical estimation with a moderate accuracy but sufficient to detect configurations close to the Pareto front. Thirdly, the interesting configurations are refined by testing more parameter values on the complete input data set.

4) Run-time Approximation Management: The aim of the last step is to design and implement the run-time management of the approximation. This controller defines the strategy to control the approximation parameters at run-time. It determines the best parameter value configuration according to the requirement 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 energy the quality/energy trade-off is evaluated on a real scenario.

IV. CASE STUDY: HEVC ALL INTRA ENCODER

A. Overview of HEVC Encoding

An HEVC encoder is based on a classical *hybrid video encoder* structure that combines Inter and Intra predictions. While encoding in HEVC, each frame is split into equally-sized blocks named Coding Tree Units (CTUs) (Figure 3). 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 prediction, 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 performed at the level of PB including planar (surface fitting) mode, DC (flat) mode and 33 angular modes [12]. After computing this prediction, the encoder calculates the residuals (prediction error) by subtracting the prediction from the original samples.

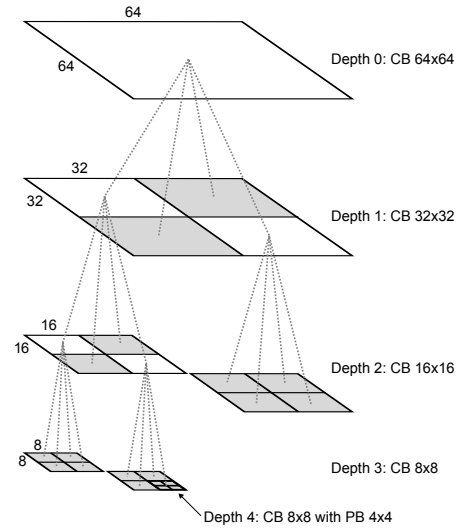


Fig. 3. Quad-tree structure of a CTB divided into CUs

The residual is then transformed by a linear spatial transform, quantized, and entropy coded.

The HEVC encoder contains also a decoder processing loop since the decoded picture is required by the encoder 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. Finally, 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 with the 35 Intra prediction modes. This exhaustive search constitutes an MSSE algorithm.

B. 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* [13]–[15] 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 Quantization Parameters (QPs) 22, 27, 32, 37 [16].

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 [17], RAPL power measurements are coherent with external

measurements and [18] proves the reliability of this internal measure across various applications.

C. Experimental Results on Applying SSSR to the Case Study

1) *MSSE 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 interlocked iterative parts. The coding-tree partitioning aims at finding the best decomposition of a CTU of 64x64 pixels into CUs as illustrated in Figure 3. Then, for all CUs, Intra-mode prediction aims at finding the best mode to predict blocks from its neighbors.

2) *MSSE 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 for the two MSSEs of the RDO process: *Coding-tree partitioning* and *Intra-mode prediction*. The MCP is the energy obtained when the encoder is able to perfectly predict the best solution and thus only this solution is tested to encode the CTU. Therefore, the energy consumption of the search process is reduced to the energy consumption of the solution and so the MCP is the minimal energy consumption point that can be achieved for the highest encoding quality.

Table I summarizes the energy reduction opportunities for the different considered video resolutions. The results show that the search space is similar across all resolutions and that the largest energy reduction search space occurs at the *Coding-tree partitioning* with up to 78.1% of potential energy reduction whereas working on the *Intra-mode prediction* offers 30% 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*.

3) MSSE Approximation:

a) *Coarse Solution Predictor Design*: In this case, the coarse solution predictor aims to predict the coding-tree partitioning from video frame content. [19], [20] show the relationship between CU size and the corresponding variance properties. Based on this observation, they propose a variance-aware coding-tree prediction. The energy reduction technique used in this paper follows the same 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

TABLE I
ENERGY REDUCTION OPPORTUNITIES (IN J)

Res.	Energy for exhaustive search in J	Energy for MCP in J		Difference (in J)		Difference (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

TABLE II
FIRST SET OF PARAMETERS

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

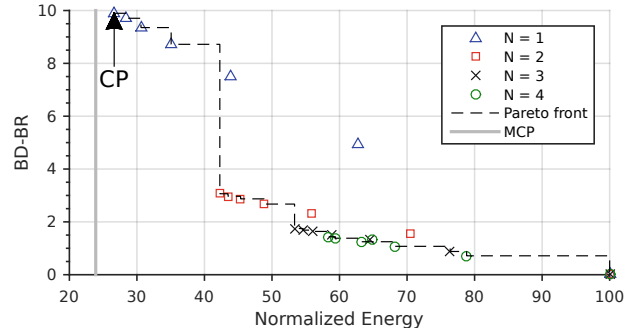


Fig. 4. Pareto in Rate-Energy space from the first set of parameters

frames of the GOF, the variance of each CU of each size are recursively compared to the thresholds to build the coding-tree partitioning.

b) *Approximation management*: The first parameter that impacts the encoding quality and energy consumption is the number of frames F in the GOF. Since applying the RDO process on the predicted depth map is the result of a coarse estimation, it is possible to improve the process by exploring more depths around the predicted optimum. The second parameter N defines the number of depth values tested around the prediction, this for each constrained CTU.

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 II summaries the first set of parameters used to explore the trade-off between energy consumption and quality. Bjøntegaard Delta Bit Rate (BD-BR) [21] 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 algorithms considering the same Peak Signal-to-Noise Ratio (PSNR). $F = 1$ represents an encoding without a constrained frame (anchor). The anchor encoding is used to normalize the energy consumption and compute BD-BRs.

c) *Quality & Cost Evaluation*: Figure 4 shows the Rate-Energy space of all the combinations of parameters defined in Table II. Identical markers correspond to different values of F for a fixed value of N . As shown in Figure 4, the coarse solution predictor is able to predict a solution close to the MCP in term of energy; the difference between C_{MCP} and C_{CP} is around 2% of energy. The results show that the relation between energy consumption and BD-BR seems close to linear compared to F for a fixed value of N . Nevertheless, a significant gap in term of BD-BR divides points between $N = 1$ and $N = 2$. This observation requires to refine N and to use non-integer values. To explore non-integer numbers of depths, CTUs in constrained frame are split in two

TABLE III
SECOND SET OF PARAMETERS

Parameter	Values
F	1, 2, 4, 8, 12, 32, 50
N	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

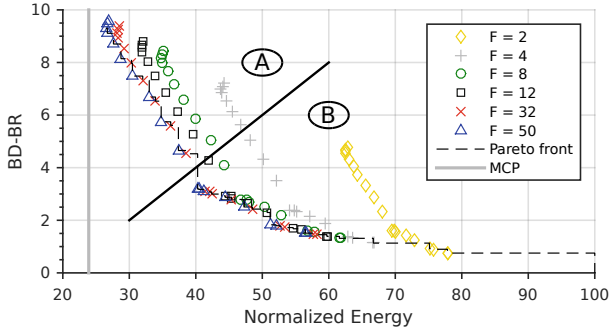


Fig. 5. Pareto in Rate-Energy space from the second set of parameters

categories: $(N - \lfloor N \rfloor) \times 100$ per cent of CTUs are encoded with $\lfloor N \rfloor$ depths and $(1 - (N - \lfloor N \rfloor)) \times 100$ per cent with $\lfloor N \rfloor$ depths.

Table III resumes the second set of parameters used to explore the trade-off between energy consumption and quality. Figure 5 shows the Rate-Energy space of all the combinations of parameters defined in Table III. The Figure 5 shows that Pareto front have an inflection point (illustrated by the black line in Figure 5). This inflection point split the Pareto front into two parts (A and B). In part A, a normalized energy reduction of up to 60% has a strong impact on the quality. The points of the Pareto front are generated with a high value of F and a low value of N . In the other hand, in part B, the quality degradation increases slowly when the consumed energy is reduced. The configurations are obtained with $F = 2$ and a high value of N . To conclude, the encoder has to play on both F and N parameters respectively the size of the GOF and the number of explored depths to control the energy consumption of the HEVC encoder with a acceptable quality degradation.

V. CONCLUSION

In this paper, we propose a new methodology to exploit the computation-skip concept by using a SSSR technique to explore the trade-offs between degradation and cost reduction in MSSE applications. The methodology is applied on HEVC Intra video encoder. By applying SSSR to this use case, energy reductions from 20% to 70% are explored through Pareto in Rate-Energy space. Future work will consider Intra prediction process as second MSSE and combined the approximations associated to two MSSEs of HEVC Intra encoder.

ACKNOWLEDGMENTS

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

REFERENCES

- [1] L. Benini and G. De Micheli, *Dynamic Power Management*. Boston, MA: Springer US, 1998.
- [2] V. Camus, J. Schlachter, and C. Enz, "A low-power carry cut-back approximate adder with fixed-point implementation and floating-point precision." ACM Press, 2016, pp. 1–6.
- [3] J. Ludwig, S. Nawab, and A. Chandrakasan, "Low-power digital filtering using approximate processing," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 3, pp. 395–400, 1996.
- [4] S. Sidiroglou-Douskos, S. Misailovic, H. Hoffmann, and M. Rinard, "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, 2011, pp. 124–134.
- [5] J. Sorber, A. Kostadinov, M. Garber, Brennan, E. D. Berger, and M. D. Corner, "Eon: A Language and Runtime System for Perpetual Systems," *Proceedings of the 5th international conference on Embedded networked sensor systems*, vol. 400, p. 600, 2007.
- [6] A. Sampson, W. Dietl, E. Fortuna, D. Gnanaprasam, L. Ceze, and D. Grossman, "EnerJ: Approximate data types for safe and general low-power computation," in *ACM SIGPLAN Notices*, vol. 46. ACM, 2011, pp. 164–174.
- [7] M. Carbin, S. Misailovic, and M. C. Rinard, "Verifying quantitative reliability for programs that execute on unreliable hardware." ACM Press, 2013, pp. 33–52.
- [8] E. Nogue, D. Menard, and M. Pelcat, "Algorithmic-level Approximate Computing Applied to Energy Efficient HEVC Decoding," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2016.
- [9] F. De Dinechin and A. Tisserand, "Multipartite table methods," *IEEE Transactions on Computers*, vol. 54, no. 3, pp. 319–330, 2005.
- [10] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Transactions on Electronic Computers*, no. 3, pp. 330–334, 1959.
- [11] J. Bonnot, E. Nogue, and D. Menard, "New non-uniform segmentation technique for software function evaluation," in *ASAP*. London, United Kingdom: IEEE, Jul. 2016, pp. 131–138.
- [12] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
- [13] A. Koivula, M. Viitanen, A. Lemmetti, J. Vanne, and T. D. Hmlinen, "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, 2015, pp. 1250–1254.
- [14] A. Koivula, M. Viitanen, J. Vanne, T. D. Hamalainen, and L. Fasnacht, "Parallelization of Kvazaar HEVC intra encoder for multi-core processors," in *Signal Processing Systems (SiPS), 2015 IEEE Workshop on*. IEEE, 2015, pp. 1–6.
- [15] M. Viitanen, A. Koivula, A. Lemmetti, J. Vanne, and T. D. Hamalainen, "Kvazaar HEVC encoder for efficient intra coding," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 1662–1665.
- [16] F. Bossen, "Common HM test conditions and software reference configurations," in *JCTVC-L1100*, Geneva, Switzerland, 2013.
- [17] D. Hackenberg, R. Schone, T. Ilsche, D. Molka, J. Schuchart, and R. Geyer, "An Energy Efficiency Feature Survey of the Intel Haswell Processor," in *IPDPSW*. IEEE, May 2015, pp. 896–904.
- [18] R. Efraim, N. Alon, R. Doron, A. Avinash, and W. Eliezer, "Power-Management Architecture of the Intel Microarchitecture Code-Named Sandy Bridge," *IEEE Computer Society*, vol. 32, no. 2, pp. 20–27, Apr. 2012.
- [19] M. U. K. Khan, M. Shafique, and J. Henkel, "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, 2013, pp. 1578–1582.
- [20] M. Shafique and J. Henkel, "Low power design of the next-generation high efficiency video coding," in *Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific*. IEEE, 2014, pp. 274–281.
- [21] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003.