

Towards Greener Computing Systems For Video Compression

Tiago A. da Fonseca, *Member, IEEE*, and Ricardo L. de Queiroz, *Senior Member, IEEE*

Abstract—Over the past years, multimedia communication technologies have demanded higher computing power availability and, therefore, higher energy consumption. In order to meet the challenge to provide software-based video encoding solutions with reduced consumption, we adopted a software implementation of a state-of-the-art video encoding standard and optimized its implementation in the energy (E) sense. Thus, besides looking for the coding options which lead to the best fidelity in a rate-distortion (RD) sense, we constrain the video encoding process to fit within a certain energy budget i.e., an RDE optimization. We considered energy by integrating power measurements from the system power supply unit. We present an RDE -optimized framework which allows for software-based real-time video compression, meeting the desired targets of electrical consumption, hence, controlling carbon emissions. The system can be made adaptive, dynamically tracking changes in image contents and in energy demands. We show results of energy-constrained compression wherein one can save as much as 31% of the power consumption with small impact on RD performance.

Index Terms—Green computing, video codec, H.264/AVC, software implementation, tunable fidelity.

I. INTRODUCTION

HISTORICALLY, processor manufacturers have responded to the demand for more processing power primarily with faster processor speeds. Higher clock speeds imply in higher power consumption and heat [1]. Image and video processing are driving forces behind this computational power pursuit. The *state-of-the-art* video compression standard, H.264/AVC [2], [3], [4], is a computation-hungry application used throughout the industry. Nevertheless, energy usage and carbon emissions are a major concern today. Data centers are substantially strained by electricity costs and power dissipation is a major concern in portable, battery-operated devices [5], [6], [7]. Governments are providing incentives to save energy and to promote the use of renewable energy resources. Individuals, companies, and organizations move towards energy-efficient products as energy costs have grown to be a major factor. Saving energy has become a leading design constraint for computing devices through new energy-efficient architectures and algorithms [8].

As results of this new design trend, we observe the emergence of new energy efficiency technologies [9] which provide subsystems that are able to scale the processor frequency

and voltage in order to reduce the power demands.^{1,2} Apart from the scalability of voltage and clock (dynamic voltage and frequency scaling, or DVFS), CPU manufacturers can turn-off parts of the CPU which are not being used (power gating), resulting in further savings in energy consumption and lower heat dissipation. All these technologies allow for modern processors to correlate computation throughput with energy consumption.

Traditionally, complexity can be considered as a measure of the effort to accomplish certain computation tasks and can be accounted either as the amount of memory, or the time, or the number of operations it takes to perform some computation [10]. We propose to evaluate energy demand instead of complexity [11], since energy is a fundamental resource that can be directly mapped to operational costs, and we will show that complexity estimation is not always a reliable indicator of energy consumption.

The present work suggests new strategies in the direction of saving energy in real-time computation. We present a fidelity-energy (ΦE) optimization strategy to constrain the energy demanded by an application in a real-time scenario. In a video encoder, fidelity Φ can be evaluated in terms of the rate-distortion (RD) performance [12], [13]. Then, the optimized parameters are used to implement an RDE -optimized real-time encoding framework. We chose an open-source high-performance encoder, x264 [14], as the H.264/AVC software implementation due to its excellent encoding speed and good rate distortion (RD) performance. The proposed approach suits, for example, mobile communication systems where energy efficiency is still a major bottleneck [15]. The system can be made adaptive, dynamically tracking changes in image contents and in energy demands.

The present work is similar to another [16] in the aspect of optimizing a video encoder constrained to energy expenditure. However there are significant variations. There is also work [17] proposing a power-rate-distortion model for wireless video communications under energy constraints, and the dissimilarities to both works will be discussed in the next section.

Our framework allows for real-time software-based energy-constrained video coding. We provide a management module capable of delivering the user-demanded encoding speed while spending less energy and smoothly affecting the RD -performance. Part of the novelty of our approach is that we take a standard video encoder to achieve significant encoding

Tiago A. da Fonseca is with the Gama Faculty of the University of Brasilia, Brasilia, DF, 72.444-240, Brazil (e-mail: tiago@image.unb.br).

Ricardo L. de Queiroz is with the Computer Science Department, Universidade de Brasilia, Brasilia, DF, 70910-900, Brazil (e-mail: queiroz@ieee.org)

This work was supported by FINEP and by CNPq.

Digital Object Identifier: 10.14209/jcis.2015.5

¹AMD® Cool'n Quiet™: <http://www.amd.com/us/products/technologies/cool-n-quiet/Pages/cool-n-quiet.aspx>

²Intel EIST®: <http://www.intel.com/technology/product/demos/eist/demo.htm>

energy savings (up to 31% less energy) on SD and HD video (rather than CIF and QCIF), without resorting to frame-skipping or resolution changes. Further novelty is that we analyze the encoder within a global *RDE* trade-off, wherein encoding is performed in groups of frames and the energy is actually measured. Also, it can all be done within a closed-loop-adaptive framework. We have not found these features elsewhere.

The proposed encoding framework can be considered a true example of green computing where the same task is accomplished in the same hardware system with much less energy consumption, reducing the carbon footprint of video compression systems.

II. BACKGROUND ON H.264/AVC IMPLEMENTATION

The H.264/AVC is a hybrid video codec, i.e. along with a transform module, it has a prediction module, a differential stage and a feedback loop [12]. The H.264/AVC prediction module has techniques which can be categorized in two classes: temporal (“Inter-prediction”) and spatial (“Intra-prediction”) techniques. AVC brought significant advances in Inter-prediction in comparison to earlier video standards, which include the support for a wide range of block sizes (16×16 -pixels and smaller), multiple reference frames and refined motion vectors (quarter-sample resolution for the luminance component). In Intra-prediction, the predicted block can have different sizes (besides 16×16 -pixel size macroblock, blocks of 8×8 and 4×4 -pixel size are also allowed) and is formed based on planar extrapolation of previously encoded blocks in the same frame. The prediction residue is transformed and quantized through the use of integer transforms [18].

The data set composed by block size and Intra (spatial extrapolation) choice or Inter parameters, like motion vectors and reference frames, forms the “prediction mode” of a block. The encoder typically selects the prediction mode that minimizes the difference between the predicted block and the block to be encoded, constrained to a given bitrate.

In order to scale the encoder complexity, one may modify the prediction stage, which is one of the most computationally intensive steps in digital video encoding, as the numbers in Table I suggest. These results are for Platform 1 and x264 implementation (see Sec. III) set to High Profile [19]³. Similar tables can be verified in [20] and [21] for the reference software implementation.

TABLE I
X264 RELATIVE COMPUTATIONAL COMPLEXITY FOR ENCODING
“MOBILE” (CIF) AND “MOBCAL” (720P) SEQUENCES.

Coding Stage	Resolution	
	CIF	720p
Predictions	91.24%	90.42%
Encoding	6.07%	6.13%
Other Stages	2.69%	3.45%
Total	100.0%	

³We analyzed encoder executions using *gprof*, an open source profiler. Available at <http://www.gnu.org/software/binutils>.

There are many studies into managing H.264/AVC complexity. Some explore prediction techniques for reducing computations with small *RD* penalties [22], [23], [24]. Assuming a correlation between computations and demanded energy, reducing the computations can help in reducing the energy demands. A recent work provides substantial H.264/AVC complexity reduction [25] using the reference software as baseline. Nevertheless, much of the complexity scaling would not be perceived if the framework is implemented using faster algorithms, high-performance libraries and platform dependent resources [26], [27]. Other works [28], [29] developed complexity models. Their results are evaluated using the reference H.264/AVC software (which is not optimized in terms of encoding complexity) and are tested on low-resolution material. There are recent investigations on providing complexity scalability to a high-performance encoder [30] within a somewhat short range. Energy-awareness in video compression was first presented by Sachs et al [31], who propose a proprietary video encoder for general purpose processors that trade computational complexity for compression efficiency in order to minimize total system energy. As we mentioned, the present work is similar to the one by Shafique et. al. [16] in many aspects. Nevertheless, while there the focus is in the motion estimation (ME) stage of the video coder (varying the search patterns and the motion vector precision), we cover the whole prediction stage and its different parameters. Any change in pattern can be easily re-trained and we incorporate many other parameters such as number and types (I, P, or B) of reference frames and multi-threading. Furthermore, that work [16] uses lower-resolution content (the largest frame-size tested was CIF), focuses on a hardware implementation, and relies on energy consumption estimation. We, however, focus on real-time software-based standard-definition (SD) and high-definition (HD) video coding on general purpose computers and we use actual energy measurements. Additionally, our framework is adaptive to changes in video contents and power targets. He et. al. [17] proposed a power-rate-distortion model for wireless video communications under energy constraints. They analyze the encoding mechanism of typical video coding systems and developed a parametric video encoding architecture which is fully scalable in a computational sense, focusing only on DVFS and stock processors. The baseline video encoder was H.263 [32] applied to low-resolution (QCIF, i.e. 176×144 -pixel) frames of head and shoulder sequences and allowing for frame dropping.

III. OUR H.264/AVC TEST SYSTEMS

A software-based video solution implies platform-dependent results. Nevertheless, the collected data suggests that, even for different processors and underlying hardware for different PCs, the power profile can be well characterized to reduce consumption in the mean power sense for a group of frames. Analyzing hardware implementations is beyond the scope of this paper and we use two systems as our test platforms (PCs): Platform 1 has an Intel[®] Core i7 CPU 950 processor in an Asus[®] P6X58D-E motherboard, while Platform 2 has an AMD[®] Phenom II X6 1055T processor in an Asus[®]

M4A78LT-M motherboard. Both systems have 8GB RAM DDR3, a solid-state disk Corsair® CSSD-F115GB2-A and no monitors are attached.

Both platforms run LINUX Operating System (Debian 2.6.32) in multi-user mode and the coding processes run at maximum priority, set to real-time scheduling. All unnecessary processes are made inactive and we assume that only one user requests the coding of video frames.

The reference H.264/AVC standard implementation, also known as JM⁴, tries to provide the most complete encoder/decoder implementation. The Intel® Performance Primitives (IPP) library [27] has a proprietary implementation of the H.264/AVC video codec built upon its high performance primitives. Even though we can control complexity within an IPP implementation [33], we feel that x264, an open source H.264/AVC standard-compliant implementation [14] is better suited for the present work, yielding better performance. Hence, we opted for only using x264 in our tests. x264 uses assembly-optimized routines for the most complexity-intensive operations [14] and explores “early stop” tests during rate-distortion optimization, yielding a 50-times speed-up over JM without significantly sacrificing RD performance. We ran x264 in H.264/AVC High profile: 64×64-pixel motion-estimation window, 5 reference-frames, refined RD-optimization in all macroblock predictions, quarter-pixel-precision motion vectors, uneven multi-hexagon search, 8×8 integer transform and CABAC entropy coder.

IV. POWER AND ENERGY IN COMPUTING SYSTEMS

In the scope of computing, work is related to activities associated with running programs (the microprocessor instructions involved in certain computation), power (P) is the rate at which the computer consumes electrical energy while performing these activities, and E is the accumulated electrical energy demanded by the computer during a certain time interval. Complexity [11] can be expressed as the number of iterations of an algorithm, or as the amount of memory or even the time necessary to execute it.

The distinction among energy, power and complexity is important because optimizing for one does not always ensure the others will be optimized. For example, an application can be implemented using specific instructions provided by the execution platform. This can raise the instantaneous power demand, but should reduce the execution time, perhaps bringing energy savings. So, in this example, compared to not using the specific instructions, the second implementation would have the same complexity, higher power, but reduced energy. This could be an issue for a mobile battery-operated platform. For a high-performance server, the temperature profile is an issue, so that power surges should be avoided [34], [35]. Power consumption can be addressed at different levels.

1) *Addressing power consumption at the device level:* CMOS technology prevails in modern electronic devices [1] and is usually profiled according to two power models: static and dynamic. The static (leakage) power profile is composed by the leakage currents that occur while keeping circuits

polarized, regardless of clock rates and usage. This static power is mainly determined by the type of transistors and the fabrication process technology. Reduction of the static power requires changes at the low-level system design.

The dynamic power profile is created by circuit activity (transistors switching, memory components varying their states etc.) and depends on the usage. It has two sources: short-circuit current and switched capacitance. The short-circuit current causes only 10-15% of total power consumption and there is no effective way to reduce it without compromising the performance [36]. Switched capacitance is the primary source of dynamic power consumption described as

$$P_{dynamic} \propto aC_{phys}.V^2f, \quad (1)$$

where C_{phys} is the physical capacitance, V is voltage, f is the clock frequency and a is an activity factor. In order to change physical capacitance, changes in low-level system design and fabrication methodologies are required. The combined reduction of f and V is achieved with widely-adopted DVFS, which intentionally down-scales the CPU performance, when it is not fully demanded. DVFS should ideally change dynamic power dissipation in a cubic factor because dynamic power is quadratically affected by voltage and is linearly affected by clock frequency [9].

2) *Addressing power consumption at the infra-structure level:* Studies show that the main part of the energy consumed by a server is drawn by the CPU, followed by the memory and by losses due to the power supply unit (PSU) inefficiency [37], [38]. Nowadays, the systems can dynamically enable low-power CPU modes, saving resources. Current desktop and server CPUs can consume less than 30% of their peak power at low-activity modes, leading to a dynamic power range of up to 70% of peak power [39]. In contrast, dynamic power ranges of all other server’s components are much narrower: less than 50% for DRAM, and 25% for disk drives [40]. The reason is that many components cannot be partially switched off and may have current surges while transitioning from inactivity.

3) *Addressing power consumption at the application level:* The application software can also allow for power reduction using compiler tools such as statistical optimizations and dynamic compilation [36]. Holistic approaches give applications a large role in power management decisions. Some works adopted an “architecture centric” view of applications that allows for some high-level transformations capable of reducing the system power demand [41]. Sachs et al. [31] explored a different adaptation method which involves trading the accuracy of computations for reduced energy consumption in video encoding.

The energy consumption of a computing device is not only determined by the efficiency of its physical devices, but it is also dependent on resource management and on applications usage patterns [37], [34], [42], [5], [43], [44], [45], [46].

V. ENERGY VS. COMPLEXITY

A. Saving Energy in a PC-based platform

We first define idle and full-power states. In idle state only the basic operations are executed and the scheduler keeps the

⁴“JM,” Available: <http://iphome.hhi.de/suehring/tm1>

processor “sleeping” almost at all times. In full-power state the processor carries intensive operations and the scheduler never allows the processor to “sleep”.

Because of energy management techniques like DVFS, when in idle state, our Platforms 1 and 2 demand 105W and 80W, respectively. When the computation workload increases, the power demand also increases. When in full-power state, our platforms 1 and 2 drive 240W and 180W, respectively, of active power to provide the currents to feed the increased gate switching, and to keep up with higher access rates to memory, hard-disks, buses and other components.

We consider a real-time *clocked* video coding scenario, where frames are periodically made available to be encoded at a given rate f_a , e.g. at 30 Hz, or 30 frames/second (fps). Hence, we have $T_a = 1/f_a$ seconds to encode each frame, and that is the period that governs the compression system. If we use only T_p seconds to encode each frame, in the remaining time ($T_i = T_a - T_p$) the processor may go idle. If we let P_i and P_{fp} be the power demanded in idle and full-power states, respectively, such a power profile can be illustrated as in Fig. 1(a). It is also useful to define the processing (or encoding) speed as $f_p = 1/T_p$, which indicates the speed (in fps) the encoder would be capable of encoding frames if they are available at once, say off-line. What should be clear from Fig. 1(a) is that we can save energy consumption if we reduce T_p , i.e. if we increase the encoder speed f_p . In this way, the sooner the encoder is done encoding a frame, the longer the processor goes idle (higher T_i).

In this binary utilization model, in which the processor is either fully idle or fully busy, one can save energy by increasing the encoding speed, i.e. reducing T_p , as in Fig. 1(b). An increase in encoding speed is typically obtained at the expense of RD performance. While the profile in Fig. 1(b) would demand less energy than the one in Fig. 1(a), one could also use dynamic frequency/voltage scaling to slow down the processor and do the same task as in Fig. 1(b) but at a lower pace [47]. In the case depicted in Fig. 1(c) the processor would run longer using less power. Here, we are not examining this case, but rather considering the energy savings provided in Fig. 1(b) by increasing the encoding speed.

B. Measuring energy

Energy consumption is here measured in two ways. The computer is connected by itself (no monitor or other peripherals) to a wattmeter and from there to the local power supply. We can read the energy consumption from the wattmeter on another computer at every second, through a USB port, as shown in Fig. 2. This is sufficient for steady state tests.

However, in order to investigate the energy consumption behavior at very fast cycles (e.g. 30 Hz or 60 Hz video), which are comparable to the voltage cycles of the energy provided by our local power company (60 Hz), we resorted to oscillography. For these tests we used an Elspec G4500 BlackBox and a California Instruments 5001ix sinusoidal power supply, as illustrated in Fig. 3.

Time measurements can be disturbed by the OS scheduler in real-time systems. We used a 250 Hz scheduler frequency and

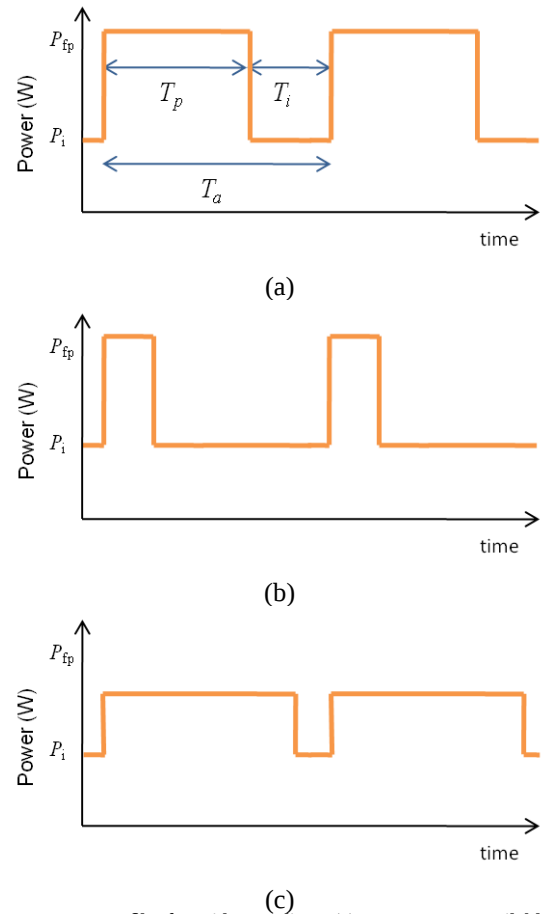


Fig. 1. Power profile for video coding. (a) Frames are available in T_a intervals. The frame is encoded in T_p seconds and the processor returns to idle state for $T_i = T_a - T_p$ seconds until a new frame arrives. (b) Profile for reduced consumption by making the encoder faster. (c) Profile for reduced consumption by making the processor less consuming and slower.

we can expect experimental errors of ± 2 ms. Considering the encoding speeds provided by our platforms, which can allow the encoding of SD and 720p video sequences of up to 250 fps, the measurement of short time intervals used to encode a video frame can be compromised. One way to overcome the scheduler-induced variances is the grouping of frames in GOPs (Group of Pictures).

The GOP grouping of frames can also affect the demanded power waveforms. To illustrate this, we monitored our test platform, while compressing 300 high-definition (720p 30-Hz) frames in real time, at different GOP sizes. As the processor is faster than necessary to guarantee real-time coding, the processor can “sleep” from the time it is done compressing a GOP until the next GOP is available for compression. The power waveform is registered by measuring the demanded power according to Fig. 3. The results from oscillography are presented in Fig. 4⁵

The waveforms show distinctive GOP grouping signatures. The rapid processor switching between idle and busy states in Fig. 4(a) is represented by an irregular sequence of peaks and

⁵For Figs. 3 to 6, we coded the head and shoulder sequence “Seq.06”. In this sequence, there is a person seated behind the table in front of a detailed background (high-frequency content background).

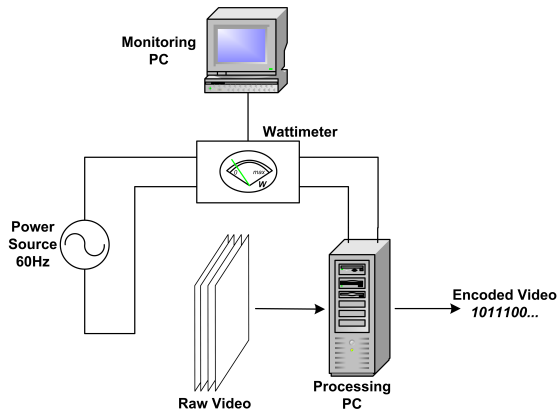


Fig. 2. On-line measurement setup.

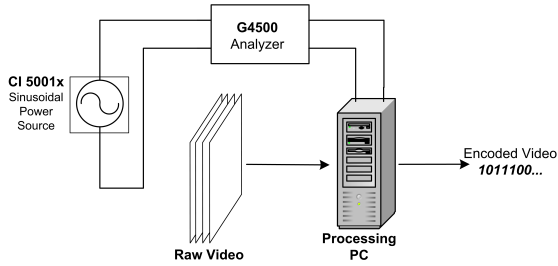


Fig. 3. Off-line measurement setup.

valleys. The plot is a zoom of the process of compressing 24 frames. When measuring power at the PC's PSU, the “sleep” moments are not well determined, as the processor does not remain in the “idle” state for a long period. This is the result of various factors: a filtering effect at the PSU related to the AC-DC conversion; the processor scaling due to DVFS; and the ACPI activity over other PC components [1], [9], [38]. As the GOP size is increased (Figs. 4(b) and (c)), the waveforms approach the model of Fig. 1(a). Basically, the GOP grouping reduces the processor state oscillation and the waveform frequency, giving the energy efficiency embedded to the PC enough time to put the processor (and other subsystems) in the “idle” state. We chose to use a 50-frame GOP to conform the waveform to the model from Fig. 1 and also to avoid OS scheduling jitter in the time measurements required by our framework. The oscillography of such a setup is presented in Fig. 5. Furthermore, the overall energy consumption is 3% lower in a GOP of 50 than it is for a single-frame GOP.

C. Complexity Issues

Computers are very complex systems, where many simultaneous events are treated by the CPU while it interacts with the user and with all the peripherals. Most applications are multi-threaded to guarantee the proper handling of all events. Complexity evaluation in a single task situation is not very precise. Nevertheless, complexity is still useful to perform comparisons of memory and time requirements [37], [9].

The precision of complexity estimation in terms of operations and time measurements is disturbed by the variances induced by computing speedup techniques, caching, compiling optimizations and the availability of multi-core CPUs. All

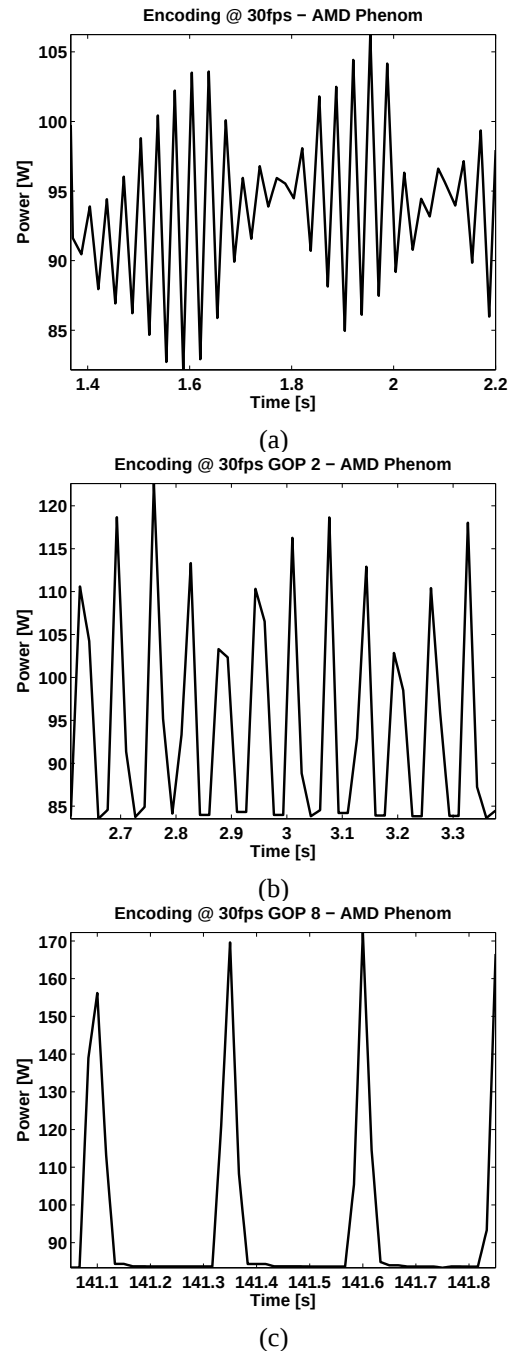


Fig. 4. Power waveforms for the encoding of 24 720p-video frames made available (and compressed) at 30 Hz and grouped in different GOPs configurations: (a) 1-frame GOP; (b) 2-frame GOP and (c) 8-frame GOP. As the GOP size is increased, the waveform tends to the Fig. 1(a) model.

these enhancement techniques, albeit improving performance, are sources of high unpredictability in time measurements, which, in turn, are also affected by the operating system activities and the concurrency of other executing applications. More variance is induced by DVFS and ACPI [36]. Therefore, the accounting of computing effort only in terms of the number of computations is imprecise and can be considered unsuitable in evaluating critical real-time applications.

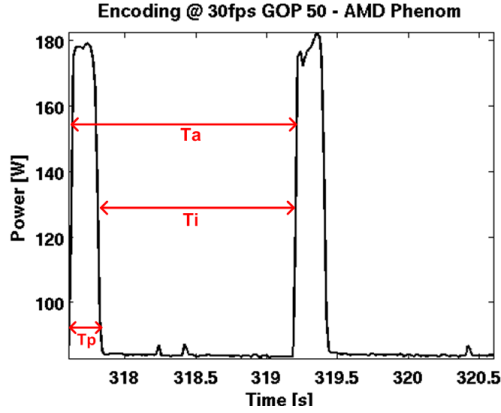


Fig. 5. Power waveforms for the encoding of 100 720p-frames recorded and compressed at 30 Hz and grouped in a 50-frame GOP. In red, we highlight the time intervals of interest for the Fig. 1(a) power model: T_a , T_p and T_i .

D. Energy as a computational effort measure

We just argued that complexity measurements based on estimates of operations can be very imprecise. Furthermore, we also argued that energy consumption is completely defined by f_p for a binary utilization model. However, while there are some applications where the correlation between the processing frequency f_p and power can be linear, for more complex tasks that relationship is not so well behaved, as illustrated in Fig. 6. This figure shows typical results relating f_p and P for a video coding task, where we compute both the power demand and speed. Note that the curve is not very linear (as expected in a logarithmic scale plot) and there are dispersed points. The reason is because the real power profile is never as well behaved as in Fig. 1, which does not account for imperfections and oscillations caused by the many hardware nuances involved. f_p cannot be easily measured with small GOP sizes as in Fig. 4. Because of that, we decided to measure real energy/power demand rather than estimating it in any way.

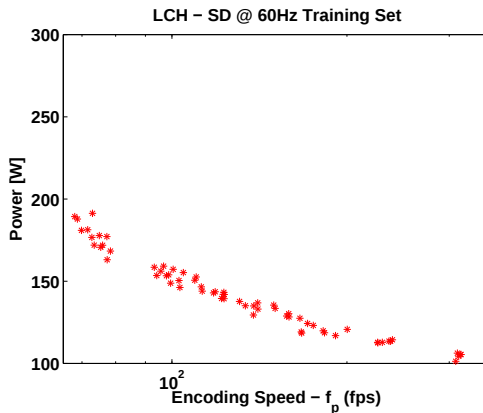


Fig. 6. Correlation of demanded power and compression speed (f_p).

VI. ENERGY-AWARE OPTIMIZATION

A. RDE optimization

Typical optimization tasks deal with cost functions or success measures. Let a software encoder execute its job for which

we can somehow measure its cost. For signal compression, the cost measure can be a measure of quality, like distortion (D) or the bit-rate (R) or a combination of both. The compression is assumed parameterized, i.e., one has the freedom to choose the values of N parameters $\{P_i\}_{i=1,\dots,N}$. Let $\mathbf{P}$ be the vector with all P_i . The encoder runs on a given set of data Z that may be different at every instantiation. For every choice of $\mathbf{P}$ and Z , we can have a measure C of the encoder cost. In essence, we can have a mapping

$$C = f(\mathbf{P}, Z).$$

Another attribute we can derive from each instantiation is the effort taken to execute the encoding task, which can be measured as demanded energy $E = g(\mathbf{P}, Z)$. It is expected that some parameters like number of iterations, data sizes, etc. would influence the demanded energy while some others would not. The central idea in this paper derives from the fact that the correlation of E and C is different for different parameters. We will use this to find points that minimize the energy consumption. The idea is illustrated in Fig. 7, which depicts a cloud of points in the cost-energy space of all achievable $\mathbf{P}$ at a given system and input data. Along with the cloud, the figure highlights a subset after optimization, the lower convex hull (LCH) of all points, represented by green square-points. Points that lie on the LCH represent instantiations that yield the lowest energy for a given cost, and is where we would like to operate. Another subset in the illustration is composed by points traversed as we increase one parameter, with all the remaining fixed, which are illustrated with red stars. Changing one parameter may lead to a sub-optimal set, away from the LCH.

Out of the many definitions of the LCH, one easy solution that leads to a slightly non-convex set is to include a point in the LCH such that no other point has simultaneously lower C and lower E than it. Hence, the algorithm to find the LCH points, in this case, is rather simple. We make a list of LCH points (initially empty). A new candidate point P to the LCH has to be compared to all the points in the LCH list. If no point in the list has simultaneously lower C and lower E , the candidate point is inserted in the LCH list. Before the point is inserted in the list, we also need to be check if any point in the LCH needs to be removed because of the new one, i.e. if it has simultaneously lower C and lower E . We repeat the process for all points in the cloud.

Despite the easier explanation using a scalar cost, in video coding, the mapping is conveniently addressed by a multidimensional variable as $\mathbf{C} = [R, D]$. Hence, $\mathbf{C} = f(\mathbf{P}, Z)$.

$\mathbf{P}$ and Z are mapped to R , D and E , adding the energy dimension to the usual RD optimization problem. We measure active power from which we can derive accumulated energy consumption. We want to find the parameters that allow us to operate on the LCH in RDE space. In this manner, we can be assured that no configuration would yield lower energy consumption for a given cost value. Conversely, we can assure that, for a given energy consumption level, no other configuration would achieve better RD performance. Figure 8 illustrates the LCH in RDE space.

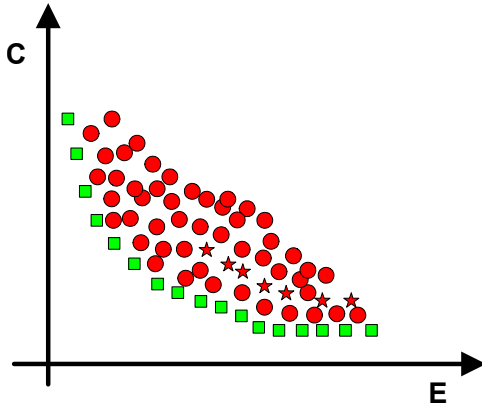


Fig. 7. Cloud of points in the energy vs. cost space. The LCH points are indicated by the green squares. A suboptimal path is achieved, for example, by varying just one parameter is illustrated by the red stars.

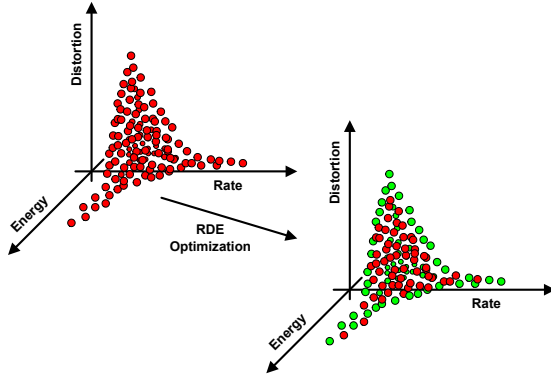


Fig. 8. Illustration on the set of *RDE* points that compose the Pareto front. The visible green points belong to the lower convex hull; some points are hidden due to the viewpoint.

Our approach is to use training data sets. Let $\{\mathbf{P}_k\}$ be the set of all parameter choices, ordered in some fashion. Let also $\mathbf{P}_k$ have elements $P_{k,n}$. If we use a representative data set $\hat{Z}$, we can span $\{\mathbf{P}_k\}$, computing E , R and D for each choice and identifying the points that belong to the LCH of $E \times R \times D$. If the n -th point belongs to the LCH, we record $\mathbf{Q}_n = [E_n, R_n, D_n, \mathbf{P}_n]$, which contains the optimal points for the set $\hat{Z}$, but which are also assumed good enough for other data. The off-line training algorithm is:

- 1) Input a representative data set $\hat{Z}$ and create an empty list Q .
- 2) For all k , compute $E_k = g(\mathbf{P}_k, \hat{Z})$ and $[R_k, D_k] = f(\mathbf{P}_k, \hat{Z})$. If point belongs to LCH, record $\mathbf{Q}_k = [E_k, R_k, D_k, \mathbf{P}_k]$ into Q .
- 3) Output a list Q of points in the LCH.

After finding the N_q points which belong to LCH, we sort Q in an ascending order of energy, i.e. $\{E_i\}$ in Q in non-decreasing. When running on-line, the parameter finding algorithm is as follows. Initially, consider a target bit-rate R^r (channel constraint) and a desired energy target E^r . Then:

- 1) Input a list Q of points in the LCH, the energy target E^r and the rate target R^r . Create an empty list L .
- 2) Span Q , for $k = 1, \dots, N_q$. If $|R_k - R^r| < \epsilon$ insert $\mathbf{Q}_k$

into L .

- 3) Count N_l , the number of items in L . Note that the items in L are still in ascending order of energy and all parameters are supposed to achieve similar bit-rate.
- 4) Span L , for $k = 1, \dots, N_l$, until $E_k \leq E^r \leq E_{k+1}$, then stop.
- 5) Find $\mathbf{P}'$ as a proportional interpolation of $\mathbf{P}_k$ and $\mathbf{P}_{k+1}$ in L .
- 6) Output parameter vector $\mathbf{P}'$.

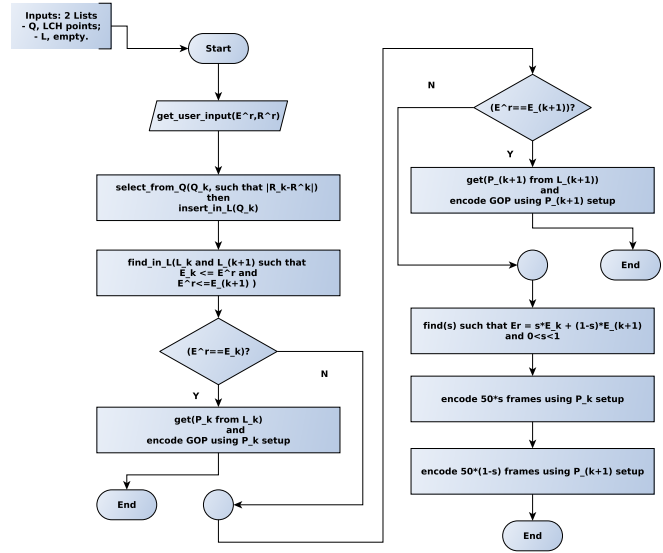


Fig. 9. Framework for multidimensional parameters interpolation used in energy-control.

Parameter set $\mathbf{P}'$ is then used to compress data set Z . Fig. 9 presents our interpolation approach to encode a GOP. We used energy targets E^r constrained to a bitrate R^r , but it is trivial to replace it with a distortion target D^r . Of course, many parameters do not assume continuous values and some action has to be taken to properly assign them. For example, the m -th parameter may use the value from $P_{k,m}$ if $E^r - E_k < E_{k+1} - E^r$, or, otherwise, the value from $P_{k+1,m}$.

If feedback control is turned on, one can monitor the system energy consumption and continuously adjust the parameters. If the energy consumption is not as predicted, it is because of discrepancies between Z and $\hat{Z}$, so that $\hat{Z}$ is not as representative as one would assume. Such a mismatch may also depend upon the non-linear mapping g . One solution is to start with a target E^r and to periodically measure the energy $E(n)$. We then adapt the parameters in order to control the energy expenditure (or cost). Assume that at any given instant n , $\mathbf{P}'$ is taken somewhere as an interpolation of $\mathbf{P}_j$ and $\mathbf{P}_{j+1}$. If $E(n) < E^r$ one should move $\mathbf{P}'$ towards $\mathbf{P}_{j+1}$ or even $\mathbf{P}_{j+2}$. Conversely, if $E(n) > E^r$ one should move in the opposite direction, i.e. towards $\mathbf{P}_j$ or even $\mathbf{P}_{j-1}$.

The control loop enjoys all the properties of trivial adaptive systems and there are many techniques to choose adaptation steps and to deal with convergence issues [48].

B. Practical approach

We use x264 as our H.264/AVC encoder, with $\mathbf{P}$ being the aggregation of the following parameters: the number of B-frames ($\#B$), the number of references frames ($\#Refs$), the motion vectors precision (MVP) used in motion compensation, the mode decision technique (MD), the quantization parameter (QP) and the number of encoding threads ($\#Thrds$). Hence,

$$\mathbf{P} = \{\#B, \#Refs, MVP, MD, QP, \#Thrds\}.$$

The first step to optimize the H.264/AVC in the E -sense is to determine a representative training set $\hat{Z}$ from where we will derive the encoder Pareto front. To build $\hat{Z}$, we opted to use standard definition (SD, 704×576-pixels) video sequences recorded at 60Hz and high definition (720p, 1280×720-pixels) video sequences recorded at 30Hz and at 50Hz. The SD training sequence was obtained by concatenating the sequences “Harbour”, “Crew” and “Soccer”. The 50Hz HD training set is composed by sequences “Parkrun,” “Stockholm” and “Tractor”. The 30Hz HD training set is composed by videoconference sequences 5, 6 and 17.

For each resolution, we encode the training set and, for each encoder instantiation, we record the bitrate, the resulting distortion and the demanded electric energy. We also measure encoding speed in order to allow for real-time compression. Those values of $\mathbf{P}_k$ which are not capable of delivering $f_p \geq f_a$ are disregarded, in such a way that the optimized codec will only accept setups which allows for real-time encoding.

The maximum number of references frames ($\#Refs$) and of threads ($\#Thrds$) were set to 5 and 8, respectively. The maximum number of B-frames ($\#B$) is restricted by x264 which bounds the maximum number of B-frames between P-frames to the number of reference frames. The other $\mathbf{P}$ components (QP, MD and MVP) are freely varied in their ranges. In summary, in the training stage⁶, we focused on finding the fastest settings leading to lower energy demand, assuring that $f_p > f_a$, by varying the motion vectors precision, the mode decision technique (the level of optimization effort in RDO), the QP, the number of reference frames and the number of B-frames.

The simulations results delivered an RDE -point cloud from which we derive the LCH. Once the LCH for the representative sequences is found, we derived look-up tables from where we can adaptively control the encoder energy demands. These tables are inserted in the energy controller framework, whose diagram is depicted in Fig. 10. The closed-loop controller in Fig. 10 manages the desktop computer power profile as discussed in Section V-A. It measures the actual encoding energy and adjusts settings. The central idea is to scale the ratio T_p/T_a , in order to adjust the demanded energy to the desired target. The closed-loop adjusts the codec to different P_{fp} and P_i levels and guarantees the target energy. If the encoder is spending more energy than it should be, the control module adjusts the encoding parameters to a

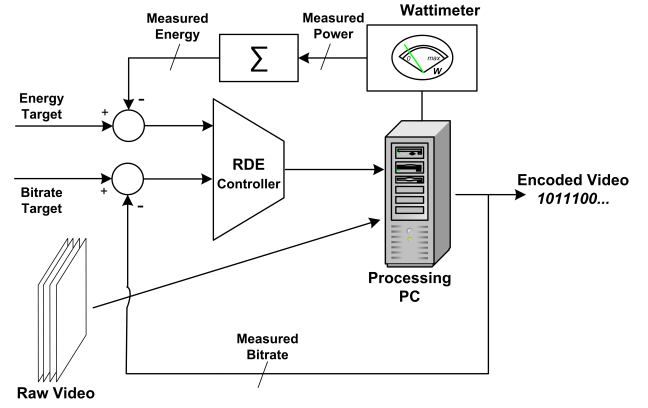


Fig. 10. Energy controller scheme. The closed-loop framework guarantees to follow the energy target required by the user. Deviations from the requested value are minimized by the framework which adjust the encoder settings in order to vary its energy demand.

less energy/power demanding setup which, in turn, yields inferior RD -performance. If there is any surplus, the encoder is allowed to use parameters which are more energy/power demanding, but also yield better performance in terms of RD .

The resulting parameters are platform dependent but the method is not, just requiring retraining once for each platform, which is not excessively complex in light of a continuous real-time operation.

An important issue is the human sensitivity to variations in quality over time. Such variations can be made smooth enough not to cause impairing. We expect higher variations, perhaps visible, at lower bit-rates when tracking large energy savings. Of course, there may be curious situations which would cause rapid oscillating behavior in quality control and cause noticeable flickering. However, our one-measurement-per-second setup in Fig. 2 only provides for very slow transitions and we have not observed any impairment.

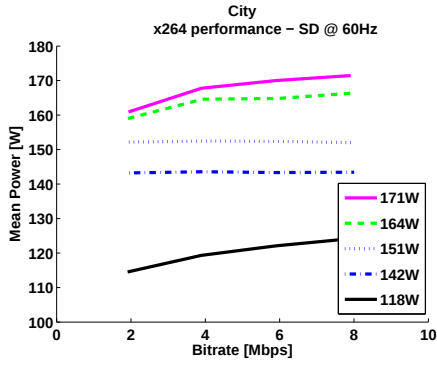
VII. RESULTS

At every sequence that is compressed we obtain an RDE triplet. In order to display results in 2D, we can use the RD plots as in Fig. 11(b), one curve for each energy (power) level. It is important to note that not all points in a curve indicate the same power consumption. We simply labeled the curve by its average as shown Fig. 11(a), which indicates the actual power consumption as the controller tracks the demanded energy target for various bit-rates.

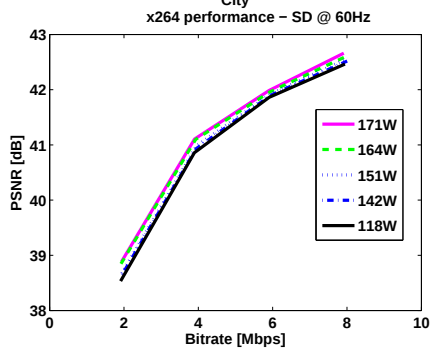
RD curves for encoding an SD sequence at different power levels are shown in Fig. 11(b) and Fig. 12. Similar plots are shown in Fig. 13 and 14 for 720p sequences at 50 Hz and 30Hz, respectively. The controller acts by forcing the energy demand to comply to the available budget. The higher baseline speed, required to handle 50Hz and 60Hz sequences, demands increased power compared to the compression of videoconferencing sequences, recorded at 30Hz.

The curves in Figs. 11 to 14 are close to each other. In order to compare them, it is convenient to analyze averaged PSNR differences between two RD curves as described in

⁶This stage is done once for each processor. Its derived parameters are used in a closed-control framework, which tries to cope with any deviation from expected reference levels. As the system is trained once, we opted to not account the total energy spent in this stage.

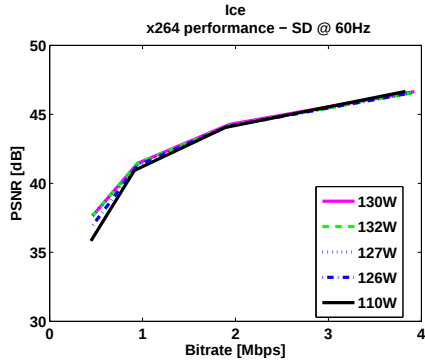


(a)

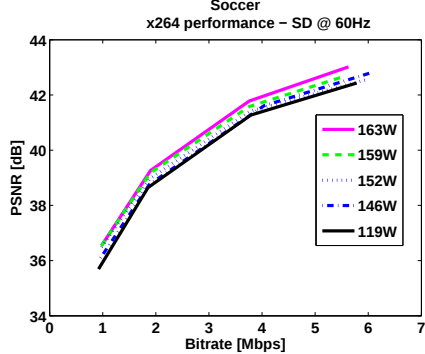


(b)

Fig. 11. Energy scaling for compressing SD sequence “City”. A range of 10% of deviation is allowed for both bitrate and power. (a) Actual demanded power for various bit-rates and several target power demands. (b) RD curves for real-time compression.



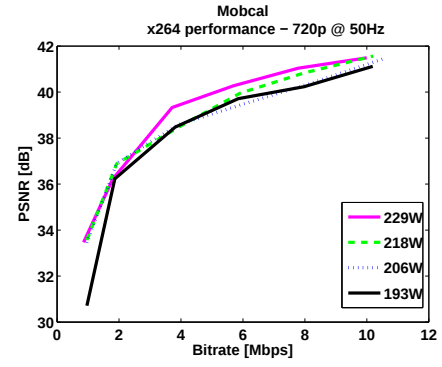
(a)



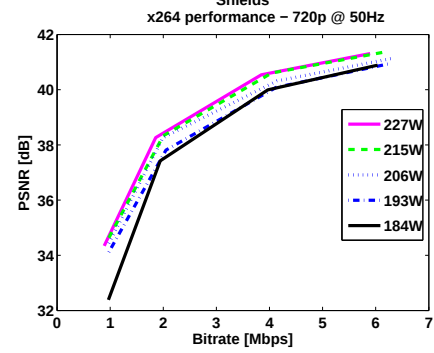
(b)

Fig. 12. RD-curves for sequences (a) “Ice” and (b) “Soccer” encoded at different averaged power levels.

[49]. For each sequence, each *RD*-curve is compared to

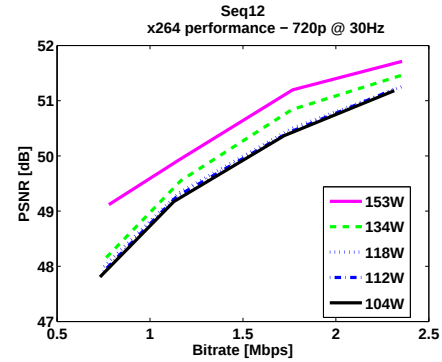


(a)

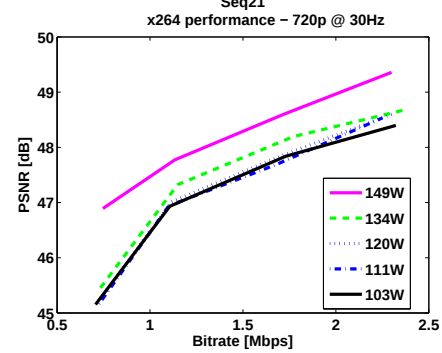


(b)

Fig. 13. RD-curves for sequences (a) “Mobcal” and (b) “Shields” encoded at different averaged power levels. These sequences were trained and evaluated in the Intel® Core™ i7-powered PC.



(a)



(b)

Fig. 14. RD-curves for videoconference sequences (a) “Seq12” and (b) “Seq21” encoded at different averaged power levels.

the best *RD*-performance setup, which, in turn, has the highest averaged power expenditure. Power expenditure is

also presented in relative numbers. The averaged results are illustrated in Fig. 15(a) for SD video sequences. The general behavior suggests that, as we reduce the available power (and energy) used to encode a video sequence, the performance penalties increase. In Fig. 15(b) the results are shown for 720p sequences.

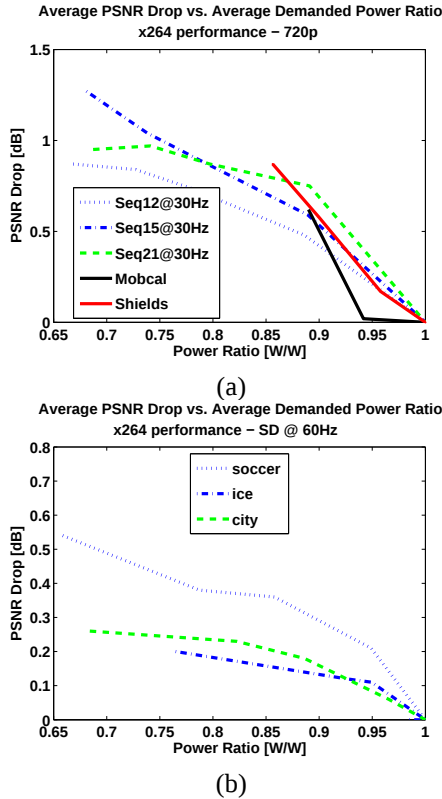


Fig. 15. PSNR drop vs. mean power ratio for (a) SD and (b) 720p video sequences. Video quality increases as we increase the power budget. A energy ratio of 1.0W/W represents the case of best *RD*-performance for real-time coding.

The main result is an energy-controlled framework which allows the user to choose the desired energy budget while real-time encoding HD and SD⁷ video sequences. As expected, the *RD*-performance tends to be penalized as the encoding speed is raised. However, the curves are close to each other and the worst case is represented by high-motion high-frequency (50Hz) detailed sequences ("Shields" and "Mobcal"). For less demanding video sequences, like those in 30 Hz video-conferencing ("Seq15" and "Seq21")⁸, PSNR reduction is less than 1.3dB on average while providing up to 31% of mean power and energy savings. The SD results, besides the increased baseline compression speed for real-time coding (60Hz), delivered lower PSNR drops (less than 0.6dB) for similar energy savings, even for very detailed video sequences. Better training sets may also lead to better results.

⁷"Soccer" is present in the training and in the evaluation steps; however, the frames used to evaluate the encoder are from a different set from those used to build the training sequence.

⁸"Seq15" and "Seq21" are scenes where there is a couple of speakers on a table: the background is plain on "Seq15" and is detailed on "Seq21".

VIII. CONCLUSIONS

We proposed an energy-optimized framework for an H.264/AVC software implementation that allows for real-time coding. Rather than using all prediction tools, we can optimally choose a subset of them, constrained by an energy budget. We have trained and adjusted parameters in order to yield the best *RD*-performance within a given power consumption budget. We also inserted a control module capable of continuously adjusting the encoder speed and throttling the energy expenditure. Our tests have shown that the *RD* performance is smoothly affected by the framework, which does not make use of frame-skipping or resolution change. Nevertheless, it provides significant encoding complexity scalability. In essence, we can perform the requested task (H.264/AVC encoding) using the requested computing system (software and hardware) using up to 31% less energy! Our framework can be readily used to build PC-based video encoder appliances that can adjust themselves to the available *RDE* conditions without the need of changing the decoder implementation. Eventual changes in image contents and in energy demands can be dynamically tracked by the adaptive control system.

This is a true example of green computing where the same task is accomplished in the same hardware system with much less energy consumption, incurring in only small *RD* performance penalties.

Algorithms and implementation of the upcoming HEVC (High Efficiency Video Coding) [50] are not mature enough for tests yet. Nevertheless, the concepts here discussed apply as well to HEVC.

REFERENCES

- [1] A. Sedra and K. Smith, *Microelectronic circuits*. Oxford University Press, USA, 1998, vol. 1.
- [2] JVT of ISO/IEC MPEG and ITU-T VCEG, "Advanced video coding for generic audiovisual services, Tech. Rep. 14496-10:2005, March 2005.
- [3] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*. John Wiley & Sons Ltd, 2003. [Online]. Available: <http://dx.doi.org/10.1002/0470869615>. doi: 10.1002/0470869615.
- [4] R. L. de Queiroz, R. S. Ortis, A. Zaghetto, and T. A. Fonseca, "Fringe benefits of the H.264/AVC," in *Proc. International Telecommunication Symposium*, 2006, pp. 208–212.
- [5] G. Procaccianti, A. Vetro', L. Ardito, and M. Morisio, "Profiling power consumption on desktop computer systems," *Information and Communication on Technology for the Fight against Global Warming*, pp. 110–123, 2011, [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23447-7_11, doi: 10.1007/978-3-642-23447-7_11.
- [6] P. Somavat, S. Jadhav, and V. Namboodiri, "Accounting for the energy consumption of personal computing including portable devices," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*. ACM, 2010, pp. 141–149, [Online]. Available: <http://dx.doi.org/10.1145/1791314.1791337>, doi: 10.1145/1791314.1791337.
- [7] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proceedings of the 2010 USENIX conference on USENIX annual technical conference*. USENIX Association, 2010, pp. 21–21.
- [8] S. Albers, "Energy-efficient algorithms," *Communications of the ACM*, vol. 53, no. 5, pp. 86–96, 2010, [Online]. Available: <http://dx.doi.org/10.1145/1735223.1735245>, doi: 10.1145/1735223.1735245.
- [9] A. Beloglazov, R. Buyya, Y. Lee, A. Zomaya, et al., "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in Computers*, vol. 82, no. 2, pp. 47–111, 2011.
- [10] L. Fortnow and S. Homer, "A short history of computational complexity," *Bulletin of the EATCS*, vol. 80, pp. 95–133, 2003.

- [11] J. Hartmanis and R. Stearns, "On the computational complexity of algorithms," *Transactions of the American Mathematical Society*, vol. 117, no. 5, pp. 285–306, 1965.
- [12] K. Sayood, *Introduction to Data Compression*. Morgan Kaufmann Publishers, 2000.
- [13] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, Nov. 1998, [Online]. Available: <http://dx.doi.org/10.1109/79.733497>, doi: 10.1109/79.733497.
- [14] L. Merritt and R. Vanam, "Improved rate control and motion estimation for H.264 encoder," in *Proc. IEEE International Conference on Image Processing*, vol. 5, 2007, [Online]. Available: <http://dx.doi.org/10.1109/ICIP.2007.4379827>, doi: 10.1109/ICIP.2007.4379827.
- [15] O. Silven and K. Jyrkkä, "Observations on power-efficiency trends in mobile communication devices," *EURASIP Journal on Embedded Systems*, vol. 2007, no. 1, pp. 17–27, 2007, [Online]. Available: <http://dx.doi.org/10.1155/2007/56976>, doi: 10.1155/2007/56976.
- [16] M. Shafique, L. Bauer, and J. Henkel, "enbudget: A run-time adaptive predictive energy-budgeting scheme for energy-aware motion estimation in h. 264/mpeg-4 avc video encoder," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010*. IEEE, 2010, pp. 1725–1730, [Online]. Available: <http://dx.doi.org/10.1109/DATE.2010.5457093>, doi: 10.1109/DATE.2010.5457093.
- [17] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 5, pp. 645–658, May 2005, [Online]. Available: <http://dx.doi.org/10.1109/TCSVT.2005.846433>, doi: 10.1109/TCSVT.2005.846433.
- [18] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, vol. 4, pp. 7–28, Jan-Mar 2004, [Online]. Available: <http://dx.doi.org/10.1109/MCAS.2004.1286980>, doi: 10.1109/MCAS.2004.1286980.
- [19] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," *Proc. SPIE Conference on Applications of Digital Image Processing XXVII*, Aug. 2004, [Online]. Available: <http://dx.doi.org/10.1117/12.564457>, doi: 10.1117/12.564457.
- [20] T. da Fonseca and R. de Queiroz, "Complexity reduction techniques for the compression of high-definition video," *Journal of Communications and Information Systems*, vol. 24, no. 1, 2009.
- [21] Y.-Y. Huang, B.-Y. Hsieh, S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 4, pp. 507–522, Apr. 2006, [Online]. Available: <http://dx.doi.org/10.1109/TCSVT.2006.872783>, doi: 10.1109/TCSVT.2006.872783.
- [22] Z. Chen, P. Zhou, and Y. He, "Hybrid unsymmetrical-cross multi-hexagon-grid search strategy for integer pel motion estimation in H.264," *Proc. Picture Coding Symposium*, Apr. 2003.
- [23] B. Kim, S.-K. Song, and C.-S. Cho, "Efficient inter-mode decision based on contextual prediction for the P-slice in H.264/AVC video coding," *Proc. IEEE International Conference on Image Processing*, pp. 1333–1336, September 2006, [Online]. Available: <http://dx.doi.org/10.1109/ICIP.2006.312664>, doi: 10.1109/10.1109/ICIP.2006.312664.
- [24] C. S. Kannangara, Y. Zhao, I. E. Richardson, and M. Bystrom, "Complexity control of H.264 based on a Bayesian framework," *Proc. Picture Coding Symposium*, Nov. 2007.
- [25] M. Moelcke and R. Seara, "Sorting rates in video encoding process for complexity reduction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 1, pp. 88–101, 2010, [Online]. Available: <http://dx.doi.org/10.1109/TCSVT.2009.2029022>, doi: 10.1109/TCSVT.2009.2029022.
- [26] K. Park, N. Singhal, M. H. Lee, and S. Cho, "Efficient design and implementation of visual computing algorithms on the GPU," *Proc. IEEE Intl. Conf. on Image Processing*, pp. 2321–2324, Nov. 2009, [Online]. Available: <http://dx.doi.org/10.1109/ICIP.2009.5414207>, doi: 10.1109/10.1109/ICIP.2009.5414207.
- [27] Intel, "Intel Integrated Performance Primitives," <http://software.intel.com/en-us/intel-ipp/>.
- [28] C. Kannangara, I. Richardson, M. Bystrom, and Y. Zhao, "Complexity control of H.264/AVC based on mode-conditional cost probability distributions," *IEEE Transactions on Multimedia*, vol. 11, no. 3, pp. 433–442, 2009, [Online]. Available: <http://dx.doi.org/10.1109/TMM.2009.2012937>, doi: 10.1109/TMM.2009.2012937.
- [29] L. Su, Y. Lu, F. Wu, S. Li, and W. Gao, "Complexity-constrained H.264 video encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 4, pp. 477–490, 2009, [Online]. Available: <http://dx.doi.org/10.1109/TCSVT.2009.2014017>, doi: 10.1109/TCSVT.2009.2014017.
- [30] M. Chien, J. Huang, and P. Chang, "Complexity control for H.264 video encoding over power-scalable embedded systems," in *Proc. 13th IEEE International Symposium on Consumer Electronics, ISCE2009*, 2009, pp. 221–224, [Online]. Available: <http://dx.doi.org/10.1109/ISCE.2009.5157007>, doi: 10.1109/ISCE.2009.5157007.
- [31] D. Sachs, S. Adve, and D. Jones, "Cross-layer adaptive video coding to reduce energy on general-purpose processors," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 3. IEEE, 2003, pp. III–109, [Online]. Available: <http://dx.doi.org/10.1109/ICIP.2003.1247193>, doi: 10.1109/ICIP.2003.1247193.
- [32] ITU-T, "ITU-t recommendation h.263, video coding for low bit rate communication," Tech. Rep., Nov. 2000.
- [33] T. A. da Fonseca, R. L. de Queiroz, and D. Mukherjee, "Complexity-scalable h. 264/avc in an IPP-based video encoder," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 2885–2888, [Online]. Available: <http://dx.doi.org/10.1109/ICIP.2010.5651898>, doi: 10.1109/ICIP.2010.5651898.
- [34] Q. Tang, S. Gupta, D. Stanzone, and P. Cayton, "Thermal-aware task scheduling to minimize energy usage of blade server based datacenters," in *Dependable, Autonomic and Secure Computing, 2nd IEEE International Symposium on*. IEEE, 2006, pp. 195–202, [Online]. Available: <http://dx.doi.org/10.1109/DASC.2006.47>, doi: 10.1109/DASC.2006.47.
- [35] W. Huang, M. Allen-Ware, J. Carter, M. Stan, K. Skadron, and E. Cheng, "Temperature-aware architecture: Lessons and opportunities," *IEEE Micro*, vol. 31, no. 3, pp. 82–86, 2011, [Online]. Available: <http://dx.doi.org/10.1109/MM.2011.60>, doi: 10.1109/MM.2011.60.
- [36] V. Venkatachalam and M. Franz, "Power reduction techniques for micro-processor systems," *ACM computing surveys*, vol. 37, no. 3, pp. 195–237, 2005, [Online]. Available: <http://dx.doi.org/10.1145/1108956.1108957>, doi: 10.1145/1108956.1108957.
- [37] S. Song, R. Ge, X. Feng, and K. Cameron, "Energy profiling and analysis of the hpc challenge benchmarks," *International Journal of High Performance Computing Applications*, vol. 23, no. 3, pp. 265–276, 2009, [Online]. Available: <http://dx.doi.org/10.1177/1094342009106193>, doi: 10.1177/1094342009106193.
- [38] A. Mahesri and V. Vardhan, "Power consumption breakdown on a modern laptop," *Power-aware computer systems*, pp. 165–180, 2005, [Online]. Available: http://dx.doi.org/10.1007/11574859_12, doi: 10.1007/11574859_12.
- [39] L. Barroso and U. Holzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, 2007, [Online]. Available: <http://dx.doi.org/10.1109/MC.2007.443>, doi: 10.1109/MC.2007.443.
- [40] X. Fan, W. Weber, and L. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 13–23, 2007, [Online]. Available: <http://dx.doi.org/10.1145/1273440.1250665>, doi: 10.1145/1273440.1250665.
- [41] T. Tan, A. Raghunathan, and N. Jha, "Software architectural transformations: A new approach to low energy embedded software," in *Proceedings of the conference on Design, Automation and Test in Europe-Volume 1*. IEEE Computer Society, 2003, p. 11046, [Online]. Available: <http://dx.doi.org/10.1109/DATE.2003.1253742>, doi: 10.1109/DATE.2003.1253742.
- [42] J. Flinn and M. Satyanarayanan, "Managing battery lifetime with energy-aware adaptation," *ACM Transactions on Computer Systems (TOCS)*, vol. 22, no. 2, pp. 137–179, 2004, [Online]. Available: <http://dx.doi.org/10.1145/986533.986534>, doi: 10.1145/986533.986534.
- [43] H. Javaid, M. Shafique, S. Parameswaran, and J. Henkel, "Low-power adaptive pipelined mpsoes for multimedia: an h. 264 video encoder case study," in *Proceedings of the 48th Design Automation Conference*. ACM, 2011, pp. 1032–1037, [Online]. Available: <http://dx.doi.org/10.1145/2024724.2024951>, doi: 10.1145/2024724.2024951.

- [44] H. Javaid, M. Shafique, J. Henkel, and S. Parameswaran, "System-level application-aware dynamic power management in adaptive pipelined mpsoes for multimedia," in *Proceedings of the International Conference on Computer-Aided Design*. IEEE Press, 2011, pp. 616–623, [Online]. Available: <http://dx.doi.org/10.1109/ICCAD.2011.6105394>, doi: 10.1109/ICCAD.2011.6105394.
- [45] M. Shafique, B. Zatt, F. L. Walter, S. Bampi, and J. Henkel, "Adaptive power management of on-chip video memory for multiview video coding," in *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012, pp. 866–875, [Online]. Available: <http://dx.doi.org/10.1145/2228360.2228516>, doi: 10.1145/2228360.2228516.
- [46] A. Mirtar, S. Dey, and A. Raghunathan, "Adaptation of video encoding to address dynamic thermal management effects," in *Green Computing Conference (IGCC), 2012 International*. IEEE, 2012, pp. 1–10, [Online]. Available: <http://dx.doi.org/10.1109/IGCC.2012.6322294>, doi: 10.1109/IGCC.2012.6322294.
- [47] E. Larsson and O. Gustafsson, "The impact of dynamic voltage and frequency scaling on multicore dsp algorithm design [exploratory dsp]," *Signal Processing Magazine, IEEE*, vol. 28, no. 3, pp. 127–144, 2011, [Online]. Available: <http://dx.doi.org/10.1109/MSP.2011.940410>, doi: 10.1109/MSP.2011.940410.
- [48] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-Constrained Coder Control and Comparison of Video Coding Standards," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 688–703, July 2003.
- [49] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *VCEG-M33*, April 2001.
- [50] G. Sullivan and J. Ohm, "Recent developments in standardization of high efficiency video coding (HEVC)," *SPIE Applications of Digital Image Processing XXXIII, Proc. SPIE*, vol. 7798, 2010, [Online]. Available: <http://dx.doi.org/10.1117/12.863486>, doi: 10.1117/12.863486.



Ricardo L. de Queiroz received the Engineer degree from Universidade de Brasília, Brazil, in 1987, the M.Sc. degree from Universidade Estadual de Campinas, Brazil, in 1990, and the Ph.D. degree from The University of Texas at Arlington, in 1994, all in Electrical Engineering.

In 1990-1991, he was with the DSP research group at Universidade de Brasília, as a research associate. He joined Xerox Corp. in 1994, where he was a member of the research staff until 2002.

In 2000-2001 he was also an Adjunct Faculty at the Rochester Institute of Technology. He joined the Electrical Engineering Department at Universidade de Brasília in 2003. In 2010, he became a Full Professor at the Computer Science Department at Universidade de Brasília. Dr. de Queiroz has published over 160 articles in Journals and conferences and contributed chapters to books as well. He also holds 46 issued patents. According to Google Scholar, his work has been cited over 3100 times. He is an elected member of the IEEE Signal Processing Society's Multimedia Signal Processing (MMSP) Technical Committee and a former member of the Image, Video and Multidimensional Signal Processing (IVMSP) Technical Committee. He is a past editor for the EURASIP Journal on Image and Video Processing, IEEE Signal Processing Letters, IEEE Transactions on Image Processing, and IEEE Transactions on Circuits and Systems for Video Technology. He has been appointed an IEEE Signal Processing Society Distinguished Lecturer for the 2011-2012 term.

Dr. de Queiroz has been actively involved with the Rochester chapter of the IEEE Signal Processing Society, where he served as Chair and organized the Western New York Image Processing Workshop since its inception until 2001. He is now helping organizing IEEE SPS Chapters in Brazil and just founded the Brasília IEEE SPS Chapter. He was the General Chair of ISCAS'2011, and MMSP'2009, and is the General Chair of SBrT'2012. He was also part of the organizing committee of ICIP'2002. His research interests include image and video compression, multirate signal processing, and color imaging. Dr. de Queiroz is a Senior Member of IEEE, a member of the Brazilian Telecommunications Society and of the Brazilian Society of Television Engineers.



Tiago A. da Fonseca received the Eng., M.Sc. and D.Sc. degrees from the Dept. of Electrical Engineering at Universidade de Brasília, Brazil, in 2006, 2008 and 2012, respectively.

In 2009-2015, he was technologist at Centro de Pesquisa e Desenvolvimento para a Segurança das Comunicações, Gabinete de Segurança Institucional da Presidência da República (CEPESC/GSI/PR).

He is currently an Adjunct Professor with the Gama Faculty of the University of Brasília. His main research interests are in image processing, video

processing and scalable video coding.