

# ESTIMATING THE HEVC DECODING ENERGY USING HIGH-LEVEL VIDEO FEATURES

*Christian Herglotz and André Kaup*

Multimedia Communications and Signal Processing  
Friedrich-Alexander University Erlangen-Nürnberg (FAU), Cauerstr. 7, 91058 Erlangen, Germany  
Email: { christian.herglotz, andre.kaup } @FAU.de

## ABSTRACT

This paper shows how the decoding energy of HEVC software decoders can be estimated when using high-level features of a coded bit stream. The investigated features comprise number of frames, resolution, bitrate, QP, and encoder configuration, where the proposed model reaches an average estimation error of 10%. While establishing this model, we closely investigated the influence of these high-level features on the decoding energy. Mathematical relations are derived that can, e.g., be exploited to control the decoding energy from the encoder side. To show the validity of our research, evaluations are performed on two different hardware devices and three different software solutions.

**Index Terms**— Video decoding, HEVC, energy estimation

## 1. INTRODUCTION

Smartphones and other portable devices have become an indispensable gadget during the past decade, allowing for phoning, messaging, or watching videos at will. A major challenge in the development of these devices is that they should provide high-quality content while using as little battery power as possible so as to maximize the device's operating time.

One of the most important applications for these gadgets is video streaming. More than half of the global mobile internet traffic constitutes video streaming [1], requiring an enormous amount of energy for transmitting, decoding, and displaying videos on the device. In this paper, we investigate the decoding process of an HEVC-coded bit stream, construct a simple estimation model, and explain in detail how high-level bit stream features influence the decoding energy. In this respect, this paper will give answers to the following questions:

- How does the decoding energy change with respect to modifications to high-level features?
- Is it possible to obtain valid energy estimates using high-level features?

In earlier work, investigations particularly aimed at modeling the power consumption of real-time streaming services for power-saving purposes. To this end, Li et al. [2] proposed a model that estimates the power consumption of an

H.264 decoder based on the features frame rate, frame size, and quantization. Later, Raoufi et al. [3] suggested using bit rate and the ratio of intra coded frames. Both models are built upon properties that describe the video bit stream at a high abstraction level.

For the state-of-the-art codec HEVC, Ren et al. [4] proposed estimating the decoding energy using hardware level descriptors such as instruction fetches, cache misses, and hardware interrupts. We suggested an even simpler method based only on the processing time of the decoder [5]. In another work, we proposed estimating the energy consumption using a set of bit stream features that describes the coded bit stream in detail and provides useful information about the energetic properties of the modeled implementation [6].

Although the latter three models provide valid and detailed energy estimates, their application is rather inconvenient. The former two models can only provide estimates after the execution of the decoding process and the latter model is based on a very large amount of parameters. Hence, in this paper, we investigate how high-level features as used by Li et al. and Raoufi et al. influence the decoding energy for HEVC decoding and how they can be exploited to obtain rough energy estimates. A possible application is to manipulate the energy consumption of the decoder during the encoding process by, e.g., changing the QP adaptively.

The paper is organized as follows: Section 2 presents the test setup that is used to determine the energy consumption. Afterwards, Section 3 introduces a number of typical high-level features and investigates their influence on the decoding energy. Then, based on these findings, a model that is capable of estimating the real decoding energy is derived in Section 4, followed by an evaluation of the estimation accuracy. Finally, Section 5 concludes the paper.

## 2. TEST SETUP

We begin by introducing the test setup as it is the basis for the observations explained in Section 3. The energy consumption  $E$  of a decoding process is measured using the setup described in [5]. The measured energies mainly comprise CPU processing and the energy consumption of the RAM. Background processes and the idle energy are not included. As a general

**Table 1.** Decoder configurations for the Pandaboard [7] and the Beagleboard [8]. Column RL denotes the runlevel of the operating system and column Optim. denotes the gcc compiler optimization flag. For the detailed evaluation of the high-level features, system (D) was used. For systems (A) to (F), the bit stream and the decoder software were loaded onto a RAM disk prior to execution.

ID	Hardware	RL	Cores	Software	Optim.
(A)	Panda	1	Single	libde265 [9]	-o3
(B)	Panda	1	Single	FFmpeg [10]	-
(C)	Panda	1	Dual	FFmpeg	-
<b>(D)</b>	<b>Panda</b>	<b>1</b>	<b>Single</b>	<b>HM-13.0 [11]</b>	<b>-o3</b>
(E)	Panda	2	Single	HM-13.0	-o3
(F)	Panda	1	Single	HM-13.0	-o0
(G)	Beagle	1	Single	HM-13.0	-o3

**Table 2.** Properties of evaluated sequences. The sequences are taken from the HEVC test set and are encoded with HM-13.0 using the standard configurations intra, lowdelay\_P, lowdelay, and randomaccess. QP is set to 10, 32, and 45.

Name	Class	Resolution	No. frames
PeopleOnStreet	A	2560 × 1600	16
Traffic	A	2560 × 1600	16
Kimono	B	1920 × 1080	16
RaceHorses	C	832 × 480	16
BasketballPass	D	416 × 240	40
BlowingBubbles	D	416 × 240	40
BQSquare	D	416 × 240	40
RaceHorses	D	416 × 240	40
vidyo3	E	1280 × 720	16
SlideEditing	F	1280 × 720	30

feature of all measurements, the reconstructed videos are discarded, i.e. sent to the null device (/dev/null). We show that the results presented in this paper are valid in general by performing the energy modeling on multiple decoding systems consisting of different software and hardware configurations as listed in Table 1. As testing the single features for all decoding systems is a highly time-consuming task, we decided to perform the detailed feature analysis explained in Section 3 only for decoding system (D).

As input videos we take sequences from the HEVC test set that are encoded using the HM-13.0 reference software [11] with different parameter sets. Particularly, the four different encoder configurations (intra, lowdelay\_P, lowdelay, and randomaccess) are tested, each using various quantization parameter (QP) values to obtain different compression levels. As the main evaluation test set we chose 10 sequences as listed in Table 2 using the four encoder configurations and three different QPs, resulting in 120 bit streams. Moreover, further streams are encoded to analyze the various high-level features in detail. These will be introduced in the next section in conjunction with the high-level feature analysis.

**Table 3.** Resolutions to test the dependency between decoding energy and number of pixels per frame. The bold resolutions denote the original resolutions of the two input sequences.

BlowingBubbles	PeopleOnStreet
<b>416 × 240</b>	320 × 200
832 × 480	512 × 320
1248 × 720	640 × 400
1664 × 960	1280 × 800
2080 × 1200	<b>2560 × 1600</b>
2496 × 1440	-

### 3. HIGH-LEVEL FEATURES

The high-level features we consider represent major properties of the coded bit stream. Furthermore, they are easy to determine such that no complex analysis of the bit stream is required. We investigate resolution, number of frames, encoder configuration, QP, and bit stream file size. They are inspired by the parameters suggested in [2] and [3]. As we aim to analyze the energy consumption instead of the power, some of these parameters are modified to fit our purposes.

#### 3.1. Resolution

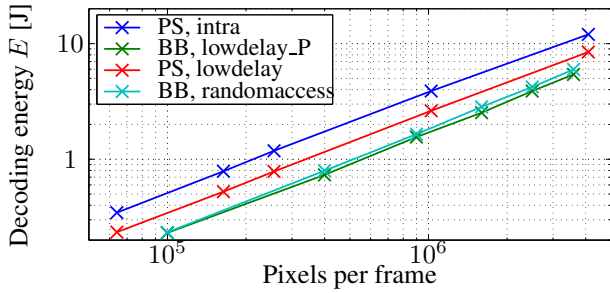
To begin with, we investigate the relation between decoding energy and image resolution. Intuitively, the work to decode a sequence grows linearly with the number of pixels to be decoded, which is experimentally proven in this subsection.

We investigate two input sequences in detail (see Table 3). As both raw sequences are only provided with a fixed resolution, we downsample and upsample them, respectively, using bi-cubic interpolation. In doing so, we ensure that the content has little impact on the decoding energy. The resulting resolutions are listed in Table 3, where the original resolutions are printed in bold.

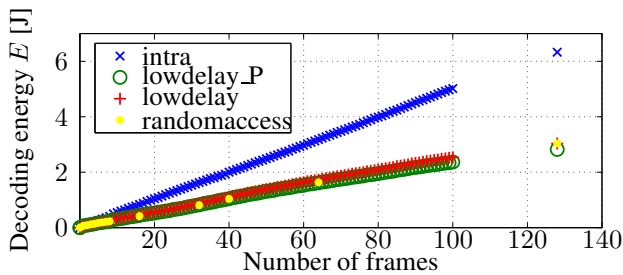
The video bit streams for this test are coded with a fixed number of 8 frames, a fixed QP of 32, and the four encoder configurations intra, lowdelay\_P, lowdelay, and randomaccess, resulting in 44 tested bit streams. The results for four representative cases are depicted in Figure 1, where we plot the decoding energy over the number of pixels per frame, which represents the resolution in a scalar value. The other curves are dropped for clarity, but show similar characteristics. The lines in the plot indicate that there is a linear relationship between number of pixels and energy, as expected.

#### 3.2. Number and Type of Frames

In a further survey we test linearity for the number of frames. Therefore, we encode the sequence BlowingBubbles with different numbers of frames ranging from 1 to 100 (fixed QP of 32, all encoder configurations). The decoding energies for these sequences, depending on the number of frames, are depicted in Figure 2. We can see that the relation is highly linear and that intra coded frames require about twice as much decoding energy as inter coded frames. In this plot we can



**Fig. 1.** Decoding energy over number of pixels per frame, both in logarithmic scale for the BlowingBubbles (BB) and the PeopleOnStreet (PS) sequence. The lines are drawn to highlight the linear relationship.



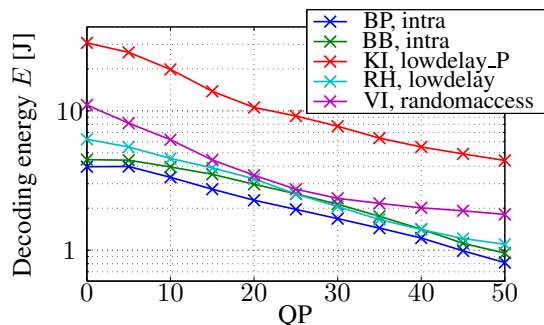
**Fig. 2.** Decoding energy over number of frames for test sequence BlowingBubbles. The QP and the resolution are fixed.

also see that the inter configurations lowdelay\_P, lowdelay, and randomaccess require a comparable amount of decoding energy.

### 3.3. Quantization Parameter (QP)

In order to obtain the influence of the QP on the energy, we measure the decoding of five input sequences (Basketball-Pass (BP), BlowingBubbles (BB), Kimono (KI), RaceHorses (RH), and vidyo3 (VI)) with all configurations and QPs ranging from 0 to 50 in increments of 5. A representative selection of the resulting curves can be found in Figure 3.

We choose a logarithmic scale for the y-axis to enhance visibility. The slope of the curves for the other bit streams are similar and differ mainly in the vertical offset. We can see



**Fig. 3.** Relation between energy and QP for different input sequences and encoder configurations. The energy is plotted in logarithmic scale.

that, as expected, the decoding energy drops with an increasing QP for all configurations.

As an interesting observation, the decoding of the BlowingBubbles sequence consumes on average 20% more energy than the decoding of the BasketballPass sequence (green curve “BB, intra” and blue curve “BP, intra”, respectively), although all the encoding parameters (range of QP, frame number, resolution, and encoder configuration) are exactly the same for both sequences. We assume that this behavior is caused by the differing video content.

To obtain the relation between QP and energy expressed in numbers, we further investigate suitable curve fittings. As an absolute relation cannot be determined due to video content dependency, the influence of a QP change on the relative decoding energy was analyzed. Therefore, we find that an exponential relationship is well suited to estimate the decoding energy

$$E_{\text{dec}} \approx \zeta \cdot e^{-\xi \cdot \text{QP}}, \quad (1)$$

where  $\xi$  is the slope of the curve and  $\zeta$  the vertical displacement that depends on resolution, number of frames, encoder configuration, and video content. This equation means that QP variations at low image qualities has a lower absolute influence on the decoding energy than QP variations at high qualities.

Curve fitting this formula for each of the configurations we found that the parameter  $\zeta$  is highly variable, in contrast to  $\xi$  which showed a rather constant value ranging from 0.025 to 0.053, where most values occurred close to the mean of 0.04. Translating this value reveals that when increasing the QP by approximately 17, the decoding energy can be halved.

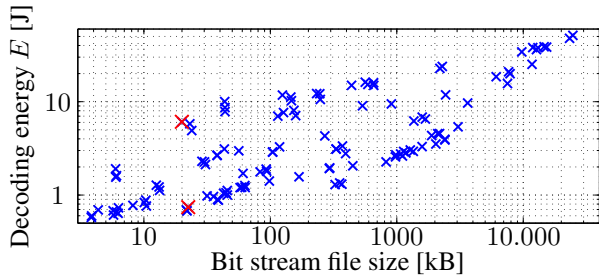
### 3.4. Bit Stream File Size

As shown in the subsection above, the decoding energy strongly depends on the content of the sequence. A very simple parameter that is easy to determine and that depends on the content is the file size of the bit stream. The more complex the content of the sequence, the larger the file will be. In this subsection, we investigate how the file size and thus the video content influence the decoding energy.

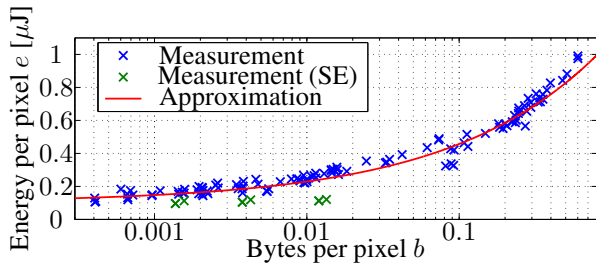
Therefore, Figure 4 plots the decoding energy over the file size for all tested bit streams. We can see that there is a relation, but that the markers are widely spread such that the energies may even vary by one order of magnitude for a fixed file size (as indicated by the two red markers).

We have seen in Subsections 3.1 and 3.2 that there is a strong linear correlation between decoding energy and resolution as well as frame number. Hence, we decided to investigate whether per-pixel values show a higher correlation. Therefore, we divide the measured decoding energies and the bit stream file sizes by their corresponding frame number and resolution. We obtain the bytes per pixel  $b$  and the energy per pixel  $e$

$$b = \left( \frac{B}{R \cdot N} \right) \quad \text{and} \quad e = \left( \frac{E}{R \cdot N} \right), \quad (2)$$



**Fig. 4.** Bit stream file sizes and measured decoding energies for the evaluation bit stream set. The two red markers indicate the variation the decoding energy can have for a fixed bit stream file size.



**Fig. 5.** Relation between bytes and decoding energy per pixel for the bit streams of the evaluation test set (blue and green markers) and approximation (red). The green markers depict the values for inter-coded frames from the sequence SlideEditing (SE). Due to the 4:2:0 color format, a raw 8-bit sequence would feature  $b = 1.5$  bytes per pixel.

where  $B$  is the bit stream file size in bytes,  $E$  the measured decoding energy,  $R = \text{width} \cdot \text{height}$  the resolution in number of pixels per frame, and  $N$  the number of frames. The result is plotted in Figure 5.

This diagram shows the mean decoding energy per pixel  $e$  over the mean bytes per pixel  $b$  for each bit stream (blue and green markers). We can see that the variation has decreased significantly and that this distribution can even be well approximated by a curve (red line). The proposed approximation will be further used and discussed in the next section.

To summarize, we have seen that a relation between decoding energy and a selected set of high-level features is inherent to HEVC software decoding. In the next section, we derive an energy model based on these observations and show the estimation accuracy for a larger set of test cases.

#### 4. HIGH-LEVEL ENERGY MODEL

Based on the observations presented above, we propose the following model to estimate the decoder's energy consumption

$$\hat{E} = R \cdot N \cdot (\alpha + \beta \cdot b^\gamma). \quad (3)$$

Corresponding to Sections 3.1 and 3.2, the resolution  $R$  and the frame number  $N$  are considered as linear terms.  $\alpha$ ,  $\beta$ , and  $\gamma$  are specific variables of the decoding system and, in conjunction with the bytes per pixel  $b$ , describe the red curve

**Table 4.** System parameters fitted for the decoding systems. The parameter  $C$  is only required for system (G).

ID	$\alpha$	$\beta$	$\gamma$	$C$
(A)	$3.55 \cdot 10e^{-8}$	$6.46 \cdot 10e^{-7}$	0.424	0
(B)	$6.57 \cdot 10e^{-8}$	$5.76 \cdot 10e^{-7}$	0.499	0
(C)	$6.73 \cdot 10e^{-8}$	$5.67 \cdot 10e^{-7}$	0.493	0
(D)	$1.01 \cdot 10e^{-7}$	$9.87 \cdot 10e^{-7}$	0.443	0
(E)	$1.58 \cdot 10e^{-7}$	$1.57 \cdot 10e^{-6}$	0.448	0
(F)	$1.68 \cdot 10e^{-7}$	$3.54 \cdot 10e^{-6}$	0.370	0
(G)	$2.16 \cdot 10e^{-8}$	$2.76 \cdot 10e^{-7}$	0.763	0.132

shown in Figure 5. Furthermore, as we found in further tests that including the QP or the encoder configuration into the formula does not increase the estimation accuracy significantly, we conclude that both these parameters are well represented by the bytes per pixel  $b$  and disregard them in the following.

#### 4.1. Model Accuracy

To match this model to our investigated systems shown in Section 2, we calculate least-squares fits using the evaluation bit stream set (Table 2). The resulting parameter values are summarized in Table 4.

During the evaluation, we found that system (G) has a special property. Using the model proposed above returns very poor estimation results in comparison to the other systems. Investigating closely the measured decoding energies revealed that the model needs to be modified: For each bit stream, a constant  $C$  has to be added:

$$\hat{E}_{(G)} = C + R \cdot N \cdot (\alpha + \beta \cdot b^\gamma). \quad (4)$$

This observation will be discussed in the next subsection.

To show the estimation precision between the measured decoding energy  $E$  and the estimated decoding energy  $\hat{E}$  for a single bit stream, we calculate the estimation error for each decoding system  $X$  and bit stream  $m$  by

$$\varepsilon_{X,m} = \frac{\hat{E} - E}{E} \Big|_{X,m}. \quad (5)$$

Investigating the results for the different raw sequences, it is striking that especially the bit streams encoded from the sequence SlideEditing (SE) are poorly estimated. These sequences correspond to the green markers in Figure 5 and are located significantly below the red approximation curve. As this observation is true for all decoding systems, we conclude that special video content like, e.g. screen content or static scenes, should be estimated using different parameter values.

We demonstrate the overall estimation accuracy by calculating the mean absolute estimation error for each decoding system

$$\bar{\varepsilon}_X = \frac{1}{M} \sum_{m=1}^M |\varepsilon_{X,m}|. \quad (6)$$

**Table 5.** Mean absolute estimation errors for each decoding system. We show the estimation error including and excluding the SlideEditing (SE) sequences. The right column shows the estimation errors from the simple model in [6] for systems (A) to (D), the others were not evaluated.

ID	$\bar{\varepsilon}$ incl. (SE)	$\bar{\varepsilon}$ excl. (SE)	$\bar{\varepsilon}$ [6]
(A)	19.23%	10.95%	5.58%
(B)	14.76%	9.82%	6.51%
(C)	15.18%	9.96%	5.83%
(D)	14.51%	9.36%	4.47%
(E)	14.12%	9.29%	–
(F)	23.55%	13.82%	–
(G)	12.22%	10.27%	–

The results are shown in Table 5, where we distinguish between the mean error with and without the SlideEditing sequences. Furthermore, we give the estimation errors of the more sophisticated model using 20 parameters presented in [6] for comparison. We can see that, disregarding screen content videos, the proposed model reaches mean estimation errors of about 10%. This is only about twice as high as the sophisticated model [6] which is remarkable as the model is only based on three instead of 20 parameters.

#### 4.2. Interpretation & Constraints

The system specific variable  $\alpha$  can be interpreted as the basic energy needed to reconstruct one pixel of a sequence, which includes initializing and saving the pixel data to the memory.  $\beta$  and  $\gamma$  describe how the energy increases when a higher number of bits is spent per pixel, which mainly reflects the decoding complexity. In this manner,  $b$  summarizes information about the input data complexity (the video content) and the chosen QP value, which both have a major impact on the bit stream file size.

The constant  $C$  introduced in (4) can be explained by the RAM usage: For system (G), the HM-decoder software needs to be loaded from the flash memory to the RAM. For the other systems, this is not required as the software is stored on a RAM disk beforehand. Hence, this process is not part of the measurement such that the constant  $C$  is not required.

Another constraint can be identified when analyzing short sequences. We found that the estimation error for sequences containing only four frames rises up to 28.91% when excluding the SlideEditing sequences (system (D)). Hence, the proposed model is only applicable for sequences containing at least 16 frames (cf. Table 2).

### 5. CONCLUSIONS

In this paper, we have shown the relation between high-level features and the decoding energy of software decoders for HEVC coded bit streams. Features that influence the decoding energy most are the number of frames, the resolution, the QP, the encoder configuration, and the content of the input

sequence. As the impact of the latter three features is hard to determine, we introduced the feature “bytes per pixel” which is highly suitable to represent these feature’s impact on the decoding energy. We showed that this model returns valid results for sequences containing at least 16 frames, where the input sequence shows no special content (e.g., screen content). An average estimation error of approximately 10% can be reached, which is only about twice as high as for more sophisticated modeling approaches [6]. In contrast, the proposed model only requires three instead of 20 or more features.

### Acknowledgment

This work was financed by the Research Training Group 1773 “Heterogeneous Image Systems”, funded by the German Research Foundation (DFG). Thanks to Benjamin Pfundt for valuable discussions on memory management.

### REFERENCES

- [1] Cisco, “Cisco visual networking index: Forecast and methodology, 2013-2018,” [http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white\\_paper\\_c11-481360.pdf](http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf), June 2014.
- [2] X. Li, Z. Ma, and F. C. A. Fernandes, “Modeling power consumption for video decoding on mobile platform and its application to power-rate constrained streaming,” in *Proc. VCIP*, San Diego, Nov 2012.
- [3] P. Raoufi and J. Peters, “Energy-efficient wireless video streaming with H.264 coding,” in *Proc. ICMEW*, July 2013, pp. 1–6.
- [4] R. Ren, E. Juarez, C. Sanz, M. Raulet, and F. Pescador, “On-line energy estimation model of an RVC-CAL HEVC decoder,” in *Proc. ICCE*, Jan 2014, pp. 63–64.
- [5] C. Herglotz, E. Walencik, and A. Kaup, “Estimating the HEVC decoding energy using the decoder processing time,” in *Proc. ISCAS*, May 2015, pp. 513–516.
- [6] C. Herglotz, D. Springer, and A. Kaup, “Modeling the energy consumption of HEVC P- and B-frame decoding,” in *Proc. ICIP*, Oct 2014, pp. 3661 – 3665.
- [7] “pandaboard.org,” <http://pandaboard.org/>, 2015, accessed 2015-02-12.
- [8] “beagleboard.org,” <http://beagleboard.org/>, 2015, accessed 2015-02-12.
- [9] “libde265 HEVC,” <http://www.libde265.org/>, 2015, accessed 2015-02-12.
- [10] “FFmpeg multimedia framework,” <http://ffmpeg.org/>, 2015, accessed 2015-02-12.
- [11] ITU/ISO/IEC, “HEVC Test Model HM13.0,” <https://hevc.hhi.fraunhofer.de/>, accessed 2015-02-12.