MODELLING POWER CONSUMPTION OF A H.263 VIDEO ENCODER

Xiaoan Lu, Thierry Fernaine, Yao Wang

Electrical and Computer Engineering, Polytechnic University, Brooklyn, New York

ABSTRACT

As video transmission is integrated into wireless communication systems, the theory of power control should also be expanded to consider both signal processing power and transmission power when designing new algorithms, since video coding consumes a significant portion of power. To better understand the interaction between signal processing and transmission, it helps to develop power consumption models for video coding. The goal of this work is to model the power consumption of a H.263 video encoder, in which motion estimation is the most computation intensive component. Different models for algorithms using 1) full search motion estimation; 2) fast algorithm using spiral order motion estimation are presented for a software H.263 encoder. We observe that one set of model parameters fits all test sequences for full search, whereas the model parameters are sequence specific for the fast algorithm.

1. INTRODUCTION

As the power-consuming video services are integrated into wireless networks, the theory of optimum power control should be expanded to include signal processing power at the mobiles. Previous research [1, 2] have shown that for the sake of reducing the total power consumption, we can either spend 1) less power in signal processing, more power in transmission; or 2) more power in signal processing, less power in transmission for a given quality at the receiver. However, how to allocate power between the signal processing and transmission depends on the video encoder, the transmitter and channel conditions. Ideally we would like to allocate between components to minimize the total power consumption. This requires accurate models of end-to-end distortion D_s and power consumption P_s for the underlying video encoder and transmission environment.

The performance of a video coder is controlled by the bit rate R_s (kbps), and adaptable parameters controlling the source coder complexity, denoted by γ, which in turn may also control the trade-off between power consumption, coding efficiency and error resilience. For example, for a H.263 video coder [3], the complexity parameter γ could be the INTER rate. If more macroblocks are encoded in the INTER-mode, more motion estimation is conducted, hence more

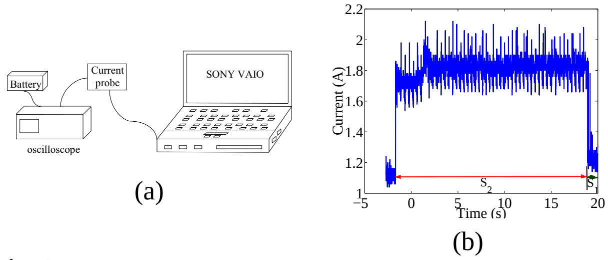

Fig. 1. Verification of source power consumption model. (a) Measurement set-up; (b) waveform of the current variation when running a software H.263 encoder.

computation is required. Increasing the INTER rate will also improve the coding efficiency at the expense of the source coder error resilience. To understand the interaction of parameters in the overall system power consumption minimization, besides developing the distortion model D_s, it is important to develop models of video coding power consumption. However, while the distortion model has been under active research [4, 5, 6], there is not much work toward modelling P_s. In our previous work [1], a simple model is proposed for a H.263 encoder with an INTRA-update scheme. In this paper we study more thoroughly the power consumption of the encoder. We first measure the power consumption of a H.263 software encoder running on a laptop and a wireless LAN (WLAN) adapter transmitting a big file, to verify that the power consumption by video encoding is in the same order of magnitude as radio transmission. Afterwards, we run the H.263 encoder with different motion estimation algorithms. Thirteen video sequences with different video patterns are test to reveal the impact of video content on the power consumption model.

The paper is organized as follows. In Section 2 we introduce the set-up of our measurement and verify that the power consumption by the H.263 encoder and WLAN transmission are comparable. In Section 3, power consumption models are proposed for different motion estimation algorithms based on measurement taken for different sequences. Section 4 concludes this paper.

2. MEASUREMENT SET-UP

We measure the power consumption of a public-domain software H.263 video coder [7] and a WLAN card transmitting data on a SONY VAIO PCG-F430 laptop computer with a 450 MHz Pentium III microprocessor. The INTRA update scheme forces a macroblock to be coded in the INTRA-mode after every $T-1$ INTER macroblocks. The INTER rate is defined as $\gamma = \frac{T-1}{T}$.

The circuit diagram is illustrated in Fig. 1(a). A Tektronix A622 current probe is used to measure the current (amps) while the voltage is held constant at $V_c = 14.8$ volts. To eliminate the effect of the power consumption by programs running at the background by the operating system, we first measure the current consumption $I_{idle}(t)$ at idle period $\mathcal{S}_1$ when no other task is running. The average

$$\bar{I}_{idle} = \int_{\mathcal{S}_1} I_{idle}(t)\mathrm{d}t / \int_{\mathcal{S}_1} \mathrm{d}t \quad \text{(amps)} \tag{1}$$

is taken as the current at the idle status. Then the current $I_0(t)$ with the H.263 program running is recorded. The difference $I(t) = I_0(t) - \bar{I}_{idle}$ is taken as the actual current level for H.263 coder. For each set of parameters, the energy consumed over the active period $\mathcal{S}_2$

$$E = \int_{\mathcal{S}_2} \left[I_0(t) - \bar{I}_{idle} \right] V_c \, \mathrm{dt} \quad \text{(joules)} \tag{2}$$

is calculated. For example, the waveform recorded from the oscilloscope when H.263 coder is running at $R_s = 100$ kbps, γ=99% for a four-second video sequence "akiyo.qcif" is illustrated in Fig. 1(b). Due to the limited computation speed of the laptop, the four-second raw video data actually took about 20 seconds to compress. Envisioning that the energy consumed by a faster computer being able to implement it in four seconds will be proportional to the measured energy consumption, we calculate the power consumption by dividing the measured energy by four seconds. In general, the actual power consumption by different computers when implementing the same algorithm may differ from our particular set-up, and we can count for this difference by a scaling factor c_{scale}.

To confirm that a video encoder spends a sizable amount of power, we also measure the power consumption by a WLAN adapter while transmitting a large file. We transmit a file of 15,206,400 bytes by a WLAN adapter (Symbol LA2101). The file is transmitted at a bit rate of 215.05 kbytes/s in 70.71 seconds. The average current is 0.0165 (amps) and the power consumption is $0.0165 \times 14.8 = 0.244$ (watts). Comparing it with the average power consumption (without scaling) of $0.5 \times 14.8 = 7.4$ (watts) for the H.263 encoder, we observe that the power consumption of a software H.263 video encoder is much larger than that of a radio transmitter. Notice that a hardware implemented program may spend significantly lower power on video encoding, but it is likely that the power consumption for video coding is at least on the same order of magnitude as radio transmission, such that it is important for a mobile system to consider both the parameters for signal processing and transmission when minimizing the overall power consumption by the system.

3. POWER CONSUMPTION MODEL

3.1. Modelling power consumption of a H.263 encoder

In this section, we present power consumption models for a H.263 encoder and further validate the models using measured data based on a laptop computer running a H.263 encoder software.

For an INTRA macroblock, the computation consists of DCT transform, quantization and entropy coding. The energy consumed for computing DCT coefficients E_{DCT} is determined by the dimension of DCT transform blocks. For block-based H.263 coder, all transform blocks have the same size, hence E_{DCT} is a constant. The DCT coefficients are then divided by quantization step sizes, scanned using a zigzag scan and converted into symbols. Each symbol consists of a run-length of zeros and a non-zero value. These symbols are converted into binary streams using Huffman entropy coding. We denote the energy consumed by quantization as E_Q, and the energy consumed by the entropy coding as E_H. Obviously, the energy required for quantization is independent of the bit rate, since different rates are obtained by the same operation but different quantization parameters. However E_H for entropy coding may be dependent on the bit rate, because at a low bit rate, more coefficients are quantized into zeros and fewer symbols need to be Huffman coded. But the exact form of the dependency of the energy consumption for entropy coding is not clear. For simplicity, we assume that E_H is linear to the bit rate R_s, i.e. E_H = $c_H R_s$. We will validate this assumption by measurements. Thus we model the energy consumption of an INTRA macroblock by

$$E_I = E_{DCT} + E_Q + c_H R_s. \tag{3}$$

For a macroblock coded in the INTER mode, extra computation E_{ME} is required for motion estimation. Motion estimation entails the search of a best matching macroblock in the reference frame over a certain search range. The computation required and hence energy consumption depend on the search range, the search algorithm and its implementation. We consider the case where the default search range specified by the H.263 standard is used. One way to estimate motion is by exhaustive search. It determines the optimal matching macroblock by comparing it with all candidate macroblocks in the reference frame and finding the one with the minimum error. The displacement between the two macroblocks is estimated motion vector. This algorithm is denoted as "full search" in our paper. Let the macroblock size be $N \times N$ pixels, and the search range be $\pm R$ pixels in both horizontal and vertical directions. For the integral-pixel search, the total number of candidate matching macroblocks is $(2R+1)^2$, the number of operations for estimating the motion vector is then $(2R+1)^2N^2$. One advantage of this algorithm is that it can be implemented in hardware using simple and modular design. However, full search requires intense computation. For this reason, various fast algorithms have been developed. In this paper, a fast algorithm with spiral order search is considered [8]. Most image sequences have smooth motion and high spatial correlation. It is quite likely that the motion vector of a block is close to the motion vectors of its neighbors. Hence, the search window center can be predicted using the motion vectors of the predictor blocks. If the spatial correlation information is right, the best match should be around this predicted center. The spiral search uses the motion vectors of the predictor blocks to get a predicted search window center. The search starts at the center of the search window, and moving outward spirally. Within one candidate macroblock in the search window, the sum of absolute difference (SAD) is accumulated line by line. After each line, the current SAD is compared with the previously determined minimum SAD, the search for the candidate macroblock terminates when the current SAD is already greater than the minimum SAD. Hence the computation for each candidate macroblock is no larger than N^2. As the search moves away from the center, the SAD calculation tends to get terminated earlier and earlier. This can significantly speed up motion estimation without reducing prediction accuracy. That is, the spiral search method will find a motion vector that leads to the same minimum SAD as the full-search method. When more than one motion vectors can lead to the same minimum SAD, the spiral search may yield one that is closer to the motion vector of the neighboring block than the full-search method. Overall, the spiral search method results in the same prediction accuracy (in terms of SAD or PSNR) as the full-search method, but at significantly reduced computation, and an added benefit of a smoother motion field. For these reasons, this method is commonly employed. This algorithm is denoted as "spiral fast search" in this paper.

3.1.1. *Full search*

In this subsection, the algorithm with full search is investigated. Since the opreation number for each macroblock is the same, E_{ME} is a constant. Hence the energy consumed by coding an INTER macroblock is modelled by

$$E_P = E_{DCT} + E_Q + c_H R_s + E_{ME}. \tag{4}$$

Because a macroblock is forced to be coded in the INTRA mode after each $T-1$ INTER frames, the average power consumption is:

$$P_s(R_s,\gamma) = f_s N_{MB} \frac{E_I + (T-1)E_P}{T}, \tag{5}$$

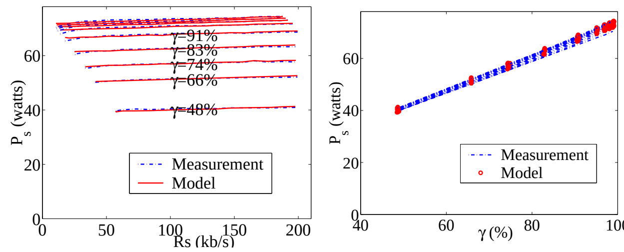

Fig. 2. Source power consumption with full search of video sequence "akiyo.qcif".

Table 1. Model parameters for the power consumption of a H.263 video coding using full search.

sequence	a_s	b_s	c_s
akiyo	6.54	65.91	0.0136
carphone	6.84	64.87	0.016
claire	6.37	63.98	0.014
container	5.02	68.06	0.019
foreman	6.32	66.66	0.02

where f_s is the encoding frame rate and N_{MB} is the number of macroblocks in one frame. The source encoder power consumption model in (5) can also be written as

$$P_s(R_s,\gamma) = a_s + b_s\gamma + c_s R_s, \tag{6}$$

where $a_s = f_s N_{MB}(E_{DCT} + E_Q)$, indicating the power consumption required for computing DCT, quantization, $b_s = f_s N_{MB} E_{ME}$, indicating the power required for performing motion estimation, and $c_s = f_s N_{MB} c_H$, indicating the relationship between the power consumption and the bit rate. Equation (6) clearly shows that the power consumption of the source coder increases linearly with γ. In the following section, we validate the proceeding power consumption model by measurement data.

Thirteen four-second QCIF test sequences are encoded at 10 fps. These sequences are selected to illustrate the impact of different motion and spatial detail on the power consumption models. Dashed lines in Fig. 2 shows the results of these measurements for the sequence "akiyo.qcif". We observe from these figures that the power is primarily determined by the INTER rate γ. The variation of the power consumption with the bit rate R_s is relatively small and almost linearly. We derive constants a_s, b_s and c_s from the measurement data using the least squares method. Results for "akiyo.qcif", "carphone.qcif", "claire.qcif", "container.qcif" and "foreman.qcif" are shown in Table 1. The power consumption curves corresponding to the model using the fitted parameters are also shown in Fig. 2. Our model fits quite well with the average power consumption at different R_s values, when γ varies over a large range. We see that the values of a_s, b_s and c_s are roughly equal for all sequences. As expected, the parameter b_s, which indicates the power consumption by motion estimation, is

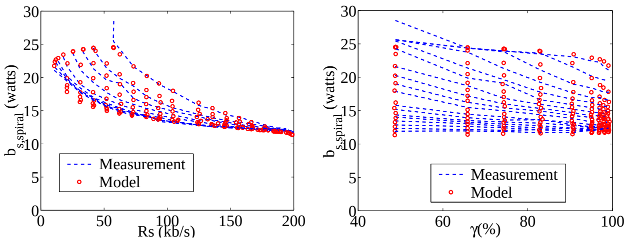

Fig. 3. Power consumed by motion estimation of a H.263 encoder using spiral fast search for video sequence "akiyo.qcif".

much larger than a_s, which counts the power consumption by DCT transform and quantization.

3.2. Spiral fast search

Then measurement is taken for spiral fast search. Unlike a H.263 encoder using regular full search, the power consumption differs among video sequences. For smooth video sequence, like "akiyo.qcif", the power consumption is much lower than that by a fast-moving sequence, "foreman.qcif". Since the only difference between two algorithms are the motion estimation strategy which has the same accuracy, we argue that E_{DCT}, E_Q, and E_H are the same for both algorithms. Hence we are able to separate the power consumption by spiral fast search by subtracting the power consumption of DCT transform, quantization and entropy coding from the overall power consumption using

$$b_{s,spiral} = (P_{s,spiral} - a_s - c_s * R_s)/\gamma, \tag{7}$$

where $P_{s,spiral}$ is the measured power consumption of an H.263 video coder with spiral fast search. Then we are ready to model the spiral order motion estimation power consumption. We observe that the power consumption of motion estimation depends on both the INTER rate and the bit rate. An empirical function using both the INTER rate and the bit rate is used to model it:

$$b_{s,spiral} = \alpha_0 - \alpha_1\gamma - (\alpha_2 - \alpha_3\gamma)R_s + \frac{\alpha_4 - \alpha_5\gamma}{R_s} \tag{8}$$

Hence the model for the overall power consumption is

$$P_{spiral} = a_s + b_{s,spiral}\gamma + c_s R_s. \tag{9}$$

The fitting result is shown in Fig. 3 as well as the measurement data. We observe that significant less power is consumed compared with full search (i.e., $b_{s,spiral} < b_s$). Furthermore we observe from Table 2 that with spiral fast search parameters for motion estimation power consumption vary significantly among sequences.

4. CONCLUSION AND FUTURE WORK

In this paper, we measure the power consumption of a H.263 encoder running with different motion estimation algorithms.

Table 2. Constant for spiral order motion estimation

sequence	α_0	α_1	α_2	α_3	α_4	α_5
akiyo	19.5	5.4	0.069	0.055	1060	987
carphone	24.1	3.2	0.056	0.037	401	343
claire	23.9	7.7	0.074	0.056	284	236
container	38.9	13.1	0.093	0.066	267	219
foreman	32.4	8.1	0.079	0.059	267	222

Different models are proposed based on our measurement data. We observe that one set of parameters fits all test sequences for full search. However, model parameters vary significantly among sequences for spiral fast search. More generally, if other algorithms are used, a_s and b_s in Eq. (6) could take other forms.

In [9], the authors reported that a uniform power cost could be associated for all the instructions without loss in accuracy. Hence it is reasonable to assume that the same H.263 algorithm implemented using an application specific integrated circuit (ASIC) or a DSP processor will still follow the model in Eq. (6), but the scaling factor c_{scale} will be much smaller.

5. REFERENCES

[1] X. Lu, Y. Wang, E. Erkip, and D. Goodman, "Power efficient multimedia communications over wireless channels," *IEEE J. Select. Areas Commun.*, pp. xxx–xxx, Dec 2003.

[2] Q. Zhang, Z. Ji, W. Zhu, and Y. Q. Zhang, "Power-minimized bit allocation for video communication over wireless channels," *IEEE Trans. on Circuits and Syst. for Video Technol.*, vol. 6, pp. 398–410, June 2002.

[3] "ITU-T Recommendation H.263: Video Coding for Low Bit Rate Communication," 1998.

[4] K. Stuhlmüller, N. Färber, M. Link and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Select. Areas in Commun.*, vol. 18, no. 6, pp. 1012–1032, June 2000.

[5] Z. He and S.K.Mitra, "A unified rate-distortion analysis framework for transform coding," *IEEE Trans. on Circuits and Syst. for Video Technol.*, vol. 11, no. 12, pp. 1221–1236, 2001.

[6] Y. Wang, J. Ostermann, and Y. Q. Zhang, *Video Processing and Communications*, Prentice Hall, 2002.

[7] University of British Columbia, "H.263+ codec," available at http://dspftp.ece.ubc.ca.

[8] M. Alkanhal, D. Turaga, and T. Chen, "Correlation Based Search Algorithms for Motion Estimation," in *1999 Picture Coding Symposium*, 1999.

[9] M. Singh and V. K. Prasanna, *System-Level Energy Tradeoffs for Collaborative Computation in Wireless Networks*, Kluwer Academic Publishers, 2002.