Modeling and Analysis of Power Consumption in Live Video Streaming Systems

YOUSEF O. SHARRAB and NABIL J. SARHAN, Wayne State University

This article develops an aggregate power consumption model for live video streaming systems, including many-to-many systems. In many-to-one streaming systems, multiple video sources (i.e., cameras and/or sensors) stream videos to a monitoring station. We model the power consumed by the video sources in the capturing, encoding, and transmission phases and then provide an overall model in terms of the main capturing and encoding parameters, including resolution, frame rate, number of reference frames, motion estimation range, and quantization. We also analyze the power consumed by the monitoring station due to receiving, decoding, and upscaling the received video streams. In addition to modeling the power consumption, we model the achieved bitrate of video encoding. We validate the developed models through extensive experiments using two types of systems and different video contents. Furthermore, we analyze many-to-one systems in terms of bitrate, video quality, and the power consumed by the sources, as well as that by the monitoring station, considering the impacts of multiple parameters simultaneously.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols; C.4 [**Computer Systems Organization**]: Performance of Systems; I.6.5 [**Simulation and Modeling**]: Model Development

General Terms: Experimentation, Measurement, Performance, Verification

Additional Key Words and Phrases: Live video streaming, power consumption modeling, video bitrate modeling, video surveillance systems

ACM Reference format:
Yousef O. Sharrab and Nabil J. Sarhan. 2017. Modeling and Analysis of Power Consumption in Live Video Streaming Systems. *ACM Trans. Multimedia Comput. Commun. Appl.* 13, 4, Article 54 (September 2017), 25 pages.
https://doi.org/10.1145/3115505

1 INTRODUCTION

Power consumption is a major concern in live video streaming systems in general and in many-to-one live video streaming systems in particular. A many-to-one streaming system includes multiple video sources (i.e., cameras and/or sensors) streaming videos to a monitoring station. These systems are typical in video surveillance and wireless video sensor networks. The monitoring station receives video streams from all sources and runs computer vision algorithms (in the case of

A preliminary version of this work (Sharrab and Sarhan 2013) was presented at the ACM Multimedia Systems Conference (MMSys'13). This work was supported by the National Science Foundation under grant CNS-0834537.
Authors' address: Department of Electrical and Computer Engineering and the Multimedia Computing and Research Lab, Wayne State University, Detroit, MI, 48202; email: {yousef.sharrab, nabil}@wayne.edu.

https://doi.org/10.1145/3115505

automated surveillance) for determining the appropriate actions, such as controlling sources and generating alerts. The energy is consumed by the source in each of the following three phases: capturing, encoding, and transmission. Although power consumption is of utmost importance when the source is battery powered, reducing power consumption is essential even when the power is available because video sources consume orders of magnitude more resources than scalar sensors (Feng et al. 2005).

This article develops an aggregate power consumption model for the sources in live video streaming systems in general and analyzes the power consumed by the monitoring station in many-to-one systems. We model the power consumed by the video source in each of the three phases and then provide an overall model. This work has been motivated by our ongoing work on the power-aware design of large-scale video surveillance systems (Alsmirat and Sarhan 2016; Sharrab and Sarhan 2012). That work requires accurate, simple, and appropriate power consumption models. The developed models can be used to assess the impacts of various video capturing and encoding parameters, and thus can help in the dynamic control of various source settings, including resolution, frame rate, number of reference frames, motion estimation (ME) range, and quantization, to achieve the best overall trade-off among power consumption, bandwidth, and video quality. (In automated video surveillance, the computer vision accuracy can be considered instead of the quality (Alsmirat and Sarhan 2016).) Although we experiment with different platforms and video contents for validation purposes, the models do not directly capture the impacts of such factors as well as environmental factors and communication strategies; all of these factors simply translate to changing constants in the developed models. For video encoding, we develop a power consumption model for both H.264 and MPEG-4, capturing the aforementioned parameters. Since tuning various parameters is often based on power consumption, video quality, and bandwidth trade-offs, we also develop a model for the output bitrate of video encoding. The bitrate affects the medium bandwidth, the video quality, and the transmission power consumption. Moreover, we analyze the power consumed by the monitoring station due to the reception, decoding, and upscaling (to the original video resolutions as captured by the sources) of the received video streams. Furthermore, we analyze many-to-one systems in terms of bitrate, video quality, and the power consumed by the sources as well as the monitoring station, considering the impacts of multiple parameters simultaneously. Although we consider the popular H.264 and MPEG-4 standards, this study can help in deriving models for high efficiency video encoding (HEVC) and VP9, which follow similar structures of operation.

We validate the developed models through extensive experiments using two types of systems and different video contents. The first includes a regular camera and employs software-based encoding with FFmpeg/x264. This system allows better flexibility in conducting the experiments. The second includes an actual video surveillance camera with a system-on-chip (SoC) for encoding. The validation and analysis results are based on 1,620 experiements, each of which is repeated at least three times, thereby totaling more than 4,800 actual experiments.

The main unique contributions of this article can be summarized as follows. First, in contrast to prior studies, we model the power consumed in all three phases. Second, we provide the first aggregate power consumption model in terms of various capturing and encoding parameters, including quantization, number of reference frames, search range, and ME algorithms. Up to our knowledge, the impacts on these parameters were not analyzed in prior work. Third, we develop a model for the bitrate achieved by video encoding, considering the aforementioned parameters. Fourth, we validate and analyze the developed models through extensive experiments, using different types of cameras, systems, and input videos. Fifth, we provide a detailed analysis of many-to-one systems in terms of bitrate, video quality, and the power consumed by the sources, as well as that by the monitoring station, considering the impacts of multiple parameters simultaneously. Sixth, we

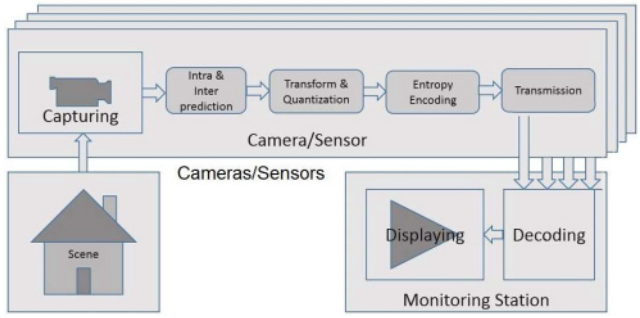


Fig. 1. Block diagram of a many-to-one video streaming system.

show that the overall computation complexity for all phases can approximately be modeled as a linear function of the pixel rate (when fixing the other parameters). The pixel rate is the product of the spatial and temporal resolutions of the raw video.

A preliminary version (Sharrab and Sarhan 2013) of this article was presented at the ACM Multimedia Systems Conference.

The rest of the article is organized as follows. Section 2 discusses background information and related work. Section 3 develops various models. Section 4 discusses the experimental setups and modeling methodology. Section 5 presents the validation results and provides an overall analysis. Finally, conclusions are drawn in Section 6.

2 BACKGROUND INFORMATION AND RELATED WORK

Power consumption is a major concern in live video streaming systems in general and in many-to-one streaming systems in particular. As shown in Figure 1, many-to-one streaming systems include multiple video sources (i.e., cameras and/or sensors) that stream videos to a monitoring station. The sources adapt their capturing and encoding parameters based on the current system state, including available resources. The monitoring station receives video streams from all sources and decodes them. In the case of automated video surveillance, it also upscales the videos to their original resolutions and then runs computer vision algorithms for determining the appropriate actions, such as controlling sources and generating alerts. Upscaling improves the accuracy of computer vision algorithms. The process at each source involves three main phases: video capturing, video encoding, and video transmission.

In the next sections, we discuss the power consumed by the source in each phase and the related work. Other related work includes studying the power consumption of video streaming from smartphones and mobile devices (Rajaraman et al. 2014), and analyzing different techniques for delivering content to the video players of smartphones (Hoque et al. 2015). Both studies did not involve any modeling.

2.1 Video Capturing Power Consumption

Cameras include image sensors, which are silicon devices that capture images. The most popular sensor type is the complementary metal oxide semiconductor (CMOS). It captures light onto an array of light-sensitive diodes, with each diode representing one pixel and converting the light photons into a charge. Each pixel has its own voltage amplifier and can be read directly on an $x - y$ coordinate system. Elouardi et al. (2007) characterized the power consumption of a smart sensor, called *PARISI* (Programmable Analog Retina-like Image Sensor I). The total power consumption

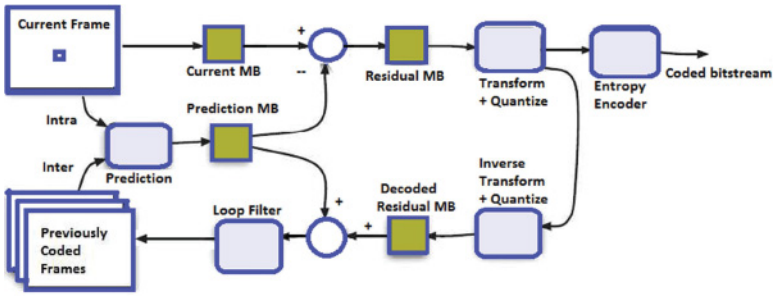


Fig. 2. Block diagram of the H.264 encoder.

for an $N \times N$ sensor with N analog processing units was shown to be given by

$$W_s = c_a N^2 + c_b N, \tag{1}$$

where c_a and c_b are constants. LiKamWa et al. (2013) developed a power consumption model for CMOS image sensors. The model is similar to that in Elouardi et al. (2007) but is more complex and includes system parameters, not only capturing and encoding parameters. This work considers the popular CMOS sensors and develops a power consumption model based on extensive experiments. Our developed capturing model is based on that of Elouardi et al. (2007).

2.2 Video Encoding Power Consumption

The main video encoders include MPEG-4 Part 2 Standard (or simply MPEG-4) and MPEG-4 Part 10 Standard (or simply H.264). As shown in Figure 1, the video encoding process can generally be divided into the following three high-level stages: the intra- and interprediction (estimation) stage; the transformation, quantization, and their inverse stage; and the entropy coding stage. In the estimation stage, both intraprediction and interprediction are used to reduce the spatial and temporal redundancies in the video, respectively. The first frame of a sequence or a random access point is typically intracoded (i.e., without using information from other frames). Each block of pixels in an intraframe is predicted using previously encoded neighboring blocks. For all remaining frames of a sequence or between random access points, intercoding is usually used, employing block motion compensation to predict blocks from other previously encoded frames. The residuals of the intraprediction and interprediction are then transformed to the frequency domain using discrete cosine transform (DCT) in MPEG-4 or Integer DCT in H.264. Subsequently, the transform coefficients are quantized, thereby reducing the overall precision of the coefficients and possibly eliminating high-frequency coefficients. The quantized transform coefficients are entropy coded and transmitted together with any possible motion vectors (MVs).

As H.264 is the primary focus of this article, let us now discuss it in more detail. Figure 2 shows its processing stages. The main features of H.264 can be summarized as follows. First, it allows using up to 16 reference frames to achieve high compression ratios, compared to only one in MPEG-4. Second, it uses variable block-size motion compensation, thereby enabling a more accurate segmentation of moving regions and higher compression ratios. The block size ranges from 4×4 pixels to 16×16 pixels, whereas MPEG-4 has a minimum block size of 8×8. Third, it employs a simplified version of the DCT. In particular, it uses a 4×4 or an 8×8 Integer DCT, whereas MPEG-4 uses an 8×8 DCT. Fourth, it employs a quantization design, which includes a logarithmic step-size control for easier bitrate management by encoders and simplified inverse-quantization scaling. Fifth, it provides two options for entropy coding: context adaptive binary arithmetic coding (CABAC) and context adaptive variable length coding (CAVLC). Both perform lossless compression

by intelligently coding the syntax elements in the video stream based on their probabilities. CABAC compresses data more efficiently than CAVLC but at the expense of increased processing at the decoder.

When coding a macroblock (MB), an H.264 encoder can choose from many different intramodes for I-frames or intermodes for B- and P-frames. Within each intermode, the encoder has a wide choice of possible MVs, leading to a huge number of options for coding an MB (Richardson 2010). The rate-distortion optimization (RDO) mode selection is an algorithm for choosing the best coding mode for each MB, based on the bitrate and distortion cost. It is used for both intraprediction and interprediction. To select the best encoding mode for an MB, the algorithm examines all possible combinations of intra- or intermodes. The bitrate cost r and distortion cost t are combined into a single cost J by $J = t + g\,r$. The RDO mode selection algorithm attempts to find the mode that minimizes the joint cost J. The trade-off between bitrate and distortion is controlled by the Lagrange multiplier g. Further details can be found in Richardson (2010).

Much of the work on H.264 dealt with managing the computation complexity (Kannangara et al. 2008; Tan et al. 2010). Sarif et al. (2015) developed a power consumption model in terms of the cycles per instruction and energy per cycle. Pu et al. (2006) considered optimization scenarios that determine how to assess the encoding parameters, but the study was limited to ME search algorithms. Specifically, it compared the bitrate, distortion, and cycles per second for each one of these scenarios. A power-rate-distortion (PRD) model of a hardware-based encoder was introduced in Kim et al. (2011) using the power-scalable architecture of H.264, considering only integer and fractional ME search range and I-frame period. Su et al. (2009) proposed a joint power-distortion optimization scheme for real-time H.264 video encoding under the power constraint. The encoding modules were divided into basic operation units, such as the sum of absolute differences (SAD) operations. Subsequently, the encoding complexity of basic operation units was determined by summing up the required processor cycles. That study considered only the ME search algorithm and did not study the spatial and temporal effects. He et al. (2005) developed a PRD framework specifically for a generic video encoder (that applies to H.263). Lu et al. (2004) developed a model for H.263 by measuring the power consumption of an H.263 encoder running with full search and fast search ME algorithms as a function of the bitrate, frame rate, and number of MBs in the frame.

We consider dynamic voltage scaling (DVS) technology, which reduces energy consumption by changing the processor speed and voltage at runtime depending on the needs of the currently running applications.

2.3 Video Transmission Power Consumption

Transmission power consumption is affected mainly by the technology or platform, distance, path-loss or environment, and bitrate. The main factors that impact the power consumption in a Wi-Fi platform are the network interface card (NIC) design (including layout, chip design, transmission output power, voltage regulations, and modulation scheme), interactions between NIC and CPU, and software protocol design (e.g., power management and drivers). Most of the recent work on transmission power consumption focused on video sensor networks. Bhardwaj and Chandrakasan (2002) developed upper bounds on the lifetime of sensor networks. He and Wu (2006) examined the resource utilization behavior of a wireless video sensor and analyzed its performance under resource constraints. Cheng and Huang (2013) studied the impact of the transmission power range on energy consumption for wireless sensor networks. Sarif et al. (2015) analyzed the power consumption in video sensor networks, using the model in He and Wu (2006). Cotuk et al. (2014) analyzed the impacts of different transmission power control strategies on wireless sensor networks in general, considering the granularity of power levels. As we are interested

Table 1. Descriptions of Used Symbols

Symbol	Description	Symbol	Description
W	Power consumption (watt)	X	Computation complexity (basic operation)
r	Bitrate (Kbit/s)	L	Pixel rate (pixel/s)
F	Frame rate (frame/s)	c	Constant
$N \times M$	Frame dimensions in pixels	$P \times Q$	Dimensions of a macroblock (MB) in pixels
K	Analog-to-digital (A/D) units (#)	Y	Bits/pixel (#)
R	Reference frames for ME (#)	d	Distance between sender and receiver (meter)
n	Path-loss index in transmission	S, S'	Displacement in pixels or subpixel for ME
q	Quantization parameter	s	Scaling factor
i	I-frames in group of pictures (#)	p	P-frames in group of pictures (#)
b	B-frames in group of pictures (#)	J	Joint cost (db)
t	Distortion (db)	g	Lagrange multiplier
N	Operations (#)	v	Voltage (volt)
f	Frequency (Hz)	V	MVs in a macroblock (MB) (#)
A	Boolean variable that is either 0 or 1		

in developing models in terms of only the video capturing and encoding parameters, we simplify previous transmission models, particularly He and Wu (2006), and adapt them accordingly.

3 MODEL DEVELOPMENT

In this section, we develop the power consumption models for each phase at the source and then develop the aggregate model. We also develop a model of the bitrate. The model for encoding is based primarily on mathematical derivation of complexity. Only two aspects of the model are driven by experiments, as noted later in Section 3.2.3 and Figure 4. Table 1 summarizes the symbols used in this section.

3.1 Modeling of the Power Consumed by Video Capturing

To model the power consumed by CMOS sensors, let us first start by generalizing Equation (1) to a general mesh of photodiodes and an associated number of A/D processing units. The per-frame power consumption W_s for a video sensor of $N \times M$ pixels and K A/D processing units can be given by

$$W_s = c_i\, N\, M + c_b\, K, \tag{2}$$

where c_i and c_b are constants. Equation (2) shows a direct relationship between the power consumption in video sensors and the spatial resolution. This equation can be extended to capturing a video by considering the temporal resolution. Thus, the total capturing power consumption W_C can be expressed as follows: $W_c = F\, W_s = F\,(c_i\, N\, M + c_b\, K)$, where F is the frame rate. The main players in the capturing power consumption are the spatial and temporal resolution. The impacts of a specific sensor type, technology, and/or implementation translate to (changing the values of) constants in the model. Our experiments confirm that the equation applies but with an additional constant:

$$W_c = F\,(c_i\, N\, M + c_b\, K) + c_j, \tag{3}$$

where c_j is a constant specifying the power consumed by the sensor when no capturing takes place. The standby power is also consistent with the findings in LiKamWa et al. (2013), but this new constant provides a much simpler way to model it.

To simplify the model, we can utilize the direct relationships between N or M and K. The value of K is typically equal to N (but conceptually it might be any fraction of it or M). Furthermore, for a megapixel camera, the $N \times M$ term dominates the K term. Therefore, the power consumption can be expressed as follows: $W_c \approx c_i\, F\, N\, M + c_j$. The power consumption can also be expressed in terms of the pixel rate L. The pixel rate is the frame rate multiplied by number of pixels in the frame and thus can be given by $L = F\, N\, M$. Consequently, the power consumption can be given by

$$W_c \approx c_i\, L + c_j, \tag{4}$$

where c_i and c_j are constants. The bitrate for the raw video is the frame size (in pixels) times the frame rate (in frames/s) times the number of bits per pixel, and thus it can be expressed in terms of the pixel rate as follows: $r = L\, Y$, where Y is the number of bits per pixel in the raw video. (In our experiments $Y = 12$, since we use the I420 color space). Therefore, the power consumption is also linear with the bitrate.

3.2 Modeling of the Power Consumed by H.264 Encoding

H.264 has high computational complexity, mainly due to its ME, complex prediction, and RDO (Shafique et al. 2010). Due to this high complexity, intraprediction (for I-frames), interprediction (for B- and P-frames), RDO, and mode selections have been active areas of research (Kim and Kuo 2007). Since the block size is adaptive, RDO operates on multiple variable block sizes, different intraprediction modes in I-frames, and different ME vectors in interframes. For each MB, RDO finds the combination (of block sizes and intraprediction modes in I-frames and block sizes and ME vectors in interframes) with the least RDO cost J (discussed in Section 2.2), among all possible combinations. For a specific MB and a specific combination, the process proceeds in the following steps: (i) compute the prediction MB, (ii) compute the residual MB, (iii) encode the residual MB (including transformation, quantization, and entropy coding), (iv) decode the MB (including inverse quantization and inverse transformation), (v) reconstruct the MB, (vi) compute distortion, and (vii) compute the cost J. This process is repeated for each combination, then the combination with the minimum cost will be selected for the MB. The whole process is repeated for each MB in the frame.

The power W_e consumed by encoding a raw video that is captured by a camera is a function of the video encoder computation complexity X_e. As discussed in Section 2.2, the computation complexity of encoding one frame is primarily the sum of the complexities of (i) interprediction and intraprediction; (ii) transformation, quantization, their inverses, reconstruction, and distortion and cost computations; and (iii) entropy encoding. Consequently, as in Zhu et al. (2001) and He et al. (2005), the encoding computation complexity X_e for F frames (taking the weighted average of different frames in a second) is given by

$$X_e = X_{inter} + X_{intra} + X_{traquant} + X_{entropy}, \tag{5}$$

where X_{inter} is the computation complexity of interprediction multiplied by the ratio of interframes to the total frames in F frames; X_{intra} is the computation complexity of intraprediction multiplied by the ratio of intraframes to the total frames in F frames; $X_{traquant}$ is the transformation, quantization, and their inverses computation complexity for F frames; and $X_{entropy}$ is the entropy encoding computation complexity for F frames.

3.2.1 Interprediction ME Computation Complexity. For interprediction RDO, a combination of ME vectors and multiple block sizes are searched for the best cost. An MB can be divided into 16×16, 16×8, 8×16, or 8×8 blocks. Since each 8×8 block can be divided further into 8×4, 4×8, or 4×4 subblocks, interprediction has seven size combinations. To select the best combination for

one MB in interprediction, the encoder considers 16 + 8 + 8 + 4 + 2 + 2 + 1 = 41 size combinations, leading to 41 RDO operations, in addition to finding the lowest residual in the search range for each of these RDO operations.

We can express X_{inter} as the sum of integer ME complexity ($X_{integer}$) and fractional ME complexity ($X_{fractional}$):

$$X_{inter} = X_{integer} + X_{fractional}. \tag{6}$$

Let us first analyze the integer ME complexity. As discussed earlier, block matching estimation and compensation are used to exploit the temporal locality among successive frames in a video by predicting blocks from previously encoded frames. This process involves partitioning the current video frame into blocks of pixels and then finding the best matching block inside a reference frame for each of these blocks, using a predefined distortion criterion. The best matching block is used for predicting the block in the current frame. Instead of coding the entire block, the encoder includes only the difference between the two blocks (i.e., the residual) and the associated MV specifying the displacement between the two blocks. For additional details, please refer to Tourapis et al. (2001). One of the commonly used distortion measure is SAD. $SAD(V_x, V_y)$ is defined as the SAD for block A located at (x, y) inside the current frame compared to block B located at a displacement of (V_x, V_y) relative to block A in the reference frame. It can be found by summing the absolute differences between each pixel in block A and the corresponding pixel in block B. In the full search algorithm, if a maximum displacement of S pixels in a frame is allowed, $(2S + 1)^2$ locations have to be searched to find the best match for the current block. For a video with a frame size of $N \times M$ (in pixels) and a frame rate of F and for an encoder that uses an MB size of $P \times Q$ and R reference frames, the integer ME computation complexity $X_{integer}$ can be given by

$$X_{integer} = F \frac{N\,M}{P\,Q} R\,(2S + 1)^2\,(2P\,Q - 1 + V\,X_{MV}), \tag{7}$$

where $(2P\,Q - 1)$ represents the number of SAD operations for the MB, V is the number of MVs in the MB, and X_{MV} is the number of operations required to calculate the MV. The number of MVs is equal to the number of blocks in the MB for a P-frame and twice the number of blocks in the MB for a B-frame. MVs are coded differentially.

Equation (7) generalizes Equation (3) in Tourapis et al. (2001) to handle multiple reference frames and consider the effect of computing MVs. The complexity for computing an MV (X_{MV}) includes 3 multiplications, 3 additions, 24 shifts, 1 median of 3 MVs, and 2 subtractions. Since the search range (S) is large compared to 1, $(2S + 1)^2$ can be simplified to $4S^2$. V and X_{MV} can be regarded as constants (on average), and $P \times Q$ is 16×16. Hence, $X_{integer}$ can be given as

$$\begin{aligned} X_{integer} &\approx (4\,F\,N\,M\,R\,S^2)\left(2 + \frac{V\,X_{MV}}{P\,Q}\right), \\ &\approx c_{integer}\,L\,R\,S^2, \end{aligned} \tag{8}$$

where the pixel rate $L = F\,N\,M$.

Let us now develop an ME complexity model for encoders supporting subpixel search. Subpixel search considers movements of a noninteger number of pixels from the reference block. The ME process here proceeds in two stages: integer pixel search over a large area and a subpixel search around the best selected integer pixel (Lin et al. 2011). The complexity depends on the number of operations for interpolating in-between pixels in the block (i.e., pixels at noninteger locations). Figure 3 demonstrates the concept of half-pixel and quarter-pixel ME. First, the encoder finds the best integer match. Subsequently, the half-pixel positions immediately next to this best match are searched. Finally, the quarter-pixel positions next to the best half-pixel position are searched (Richardson 2010).

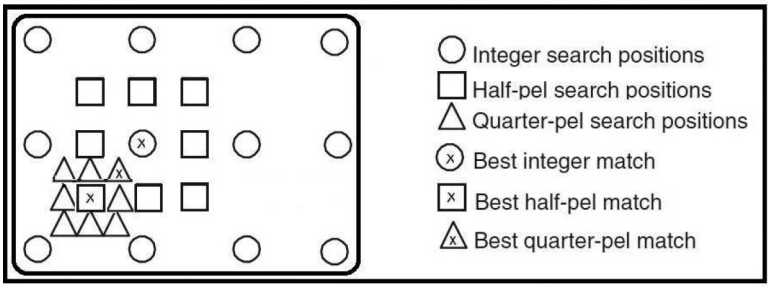


Fig. 3. Half-pixel and quarter-pixel MEs.

Table 2. Per-Pixel Computation Complexity of Interpolation for Fractional Pixel ME

Interpolation	Description	Complexity	# Operations
1/2 Pixel Luma	6-tap interpolation: a combination of 6 samples, 3 from each side of a row or a column	$X_{1/2Luma}$	5 add + 4 mul + 1 div
1/4 Pixel Chroma	Weighted mean of neighboring pixels and a constant	$X_{1/4Chroma}$	2 mul + 2 add + 1 div
1/4 Pixel Luma	Linear interpolation between adjacent samples: combination of 2 samples, 1 from each side of a row or a column	$X_{1/4Luma}$	1 add + 1 div
1/8 Pixel Chroma	Linear combination of 4 neighboring integer pixel positions	$X_{1/8Chroma}$	3 add + 4 mul + 1 div

Table 2 shows the number of interpolation operations. This complexity depends on the accuracy of the subpixel search (half a pixel, quarter a pixel, etc.). The implementation of full search in subpixel ME follows a hierarchical manner. For quarter-pixel, eight half-pixel pixels around the best integer pixel are examined first and then eight quarter-pixel pixels around the best half-pixel pixel are checked (Xu and He 2008). Note that half-pixel resolution MVs in the Luma component require quarter-pixel resolution vectors in the Chroma components (assuming 4:2:0 sampling). Similarly, quarter-pixel resolution MVs in the Luma component require eighth-pixel resolution vectors in the Chroma components. Assume that S' represents the range of the subpixel search in pixels, and X_{p1} and X_{p2} represent the numbers of pixel interpolation operations for half-pixel accuracy and quarter-pixel accuracy, respectively. Based on Table 2, X_{p1} is the number of operations in both the first and second rows (i.e., $X_{p1} = X_{1/2Luma} + X_{1/4Chroma}$) and X_{p2} is the number of operations in the third and fourth rows (i.e., $X_{p2} = X_{1/4Luma} + X_{1/8Chroma}$). The computation complexity $X_{fractional}$ of fractional pixel ME can be given by

$$\begin{aligned} X_{fractional} &= F \frac{N\,M}{P\,Q} \left(2S' + 1\right)^2 \left(2P\,Q - 1 + P\,Q\left(X_{p1} + A_{1/4}\,X_{p2}\right)\right) \\ &\approx L\left(2S' + 1\right)^2 \left(2 + X_{p1} + A_{1/4}\,X_{p2}\right), \end{aligned} \tag{9}$$

where $A_{1/4}$ is a Boolean variable that is either 0 or 1 for half- and quarter-pixel accuracy, respectively. X_{p1} and X_{p2} are constants as explained earlier. S' is fixed to 1 in subpixel accuracy ME, because the search is only one subposition in the surrounding eight directions, whether it is half- or quarter-pixel accuracy search. Therefore, $X_{fractional}$ can be expressed as

$$X_{fractional} = c_{fractional}\,L. \tag{10}$$

Table 3. Number of Operations to Compute a 4×4 Luma Prediction Block

Mode	Description	# Operations
Mode 0	Vertical: The upper row's samples are extrapolated vertically.	4×4 copy
Mode 1	Horizontal: The left column's samples are extrapolated horizontally.	4×4 copy
Mode 2	DC: The block is predicted by the mean of upper row's and left column's samples (an average of 8 values for the block).	(8-1) add + 1 div + 4×4 copy
Mode 3	Diagonal Down-Left: The samples are interpolated at a 45-degree angle between lower left and upper right. It rounds the value of three neighboring pixels, each divided by an integer.	$4 \times 4 \times$ (3 mul + 2 add + 1 round)
Mode 4	Diagonal Down-Right: The samples are extrapolated at a 45-degree angle down and to the right.	Same as Mode 3
Mode 5	Vertical-Left: Extrapolation at an angle of approximately 26.6 degrees to the left of vertical.	Same as Mode 3
Mode 6	Horizontal-Down: Extrapolation at an angle of approximately 26.6 degrees below horizontal.	Same as Mode 3
Mode 7	Vertical-Right: Extrapolation or interpolation at an angle of approximately 26.6 degrees to the right of vertical.	Same as Mode 3
Mode 8	Horizontal-Up: Interpolation at an angle of approximately 26.6 degrees horizontal.	Same as Mode 3

Based on Equations (6), (8), and (10), the total interprediction (both integer and fractional) complexity for a full search X_{inter} can be given by

$$X_{inter} \approx c_{integer} L R S^2 + c_{fractional} L, \tag{11}$$

where $c_{integer}$ and $c_{fractional}$ are constants.

3.2.2 Intraprediction Computation Complexity. H.264 exploits the spatial correlation between adjacent blocks in intraprediction. For the Luma prediction, the prediction block is formed for each 4×4 block or for a 16×16 block. One mode is selected from the supported modes, which are nine modes for a 4×4 Luma block, four modes for a 16×16 Luma block, and four modes for each Chroma block. The encoder selects the best mode using RDO. As an illustrating example, in intraprediction RDO, the number of mode combinations for one MB (16×16 pixels) is $N_8(16N_4 + N_{16})$, where N_8, N_4, and N_{16} represent the number of modes of an 8×8 Chroma block, a 4×4 Luma block, and a 16×16 Luma block, respectively. To select the best mode for one MB in intraprediction, the encoder performs $4(16 \times 9 + 4) = 592$ RDO calculations (Kim et al. 2006).

We develop Tables 3, 4, and 5 to assist in computing the complexity of intramode selection X_{intra}. These tables also include a brief description of each intramode. The complexity can be found as follows:

$$X_{intra} = F N M (4 N_c) \left(\frac{N_{I4}}{4 \times 4} \times 9 + \frac{N_{I16}}{16 \times 16} \times 4 \right), \tag{12}$$

Table 4. Number of Operations to Compute a 16 × 16 Luma Prediction Block

Mode	Description	# Operations
Mode 0	Vertical: Copy row.	16 × 16 copy
Mode 1	Horizontal: Copy column.	16 × 16 copy
Mode 2	DC: Average of 32 values for the block.	(32-1) add + 1 div + 16 × 16 copy
Mode 3	Plane: A linear plane function fitted to the upper and left-hand samples H and V. Clipping ensures $0 < result < 255$.	16 × 16 × (5 add + 2 mul + 1 compare + 1 clip)

Table 5. Number of Operations to Compute an 8 × 8 Chroma Prediction Block

Mode	Description	Operations (#)
Mode 0	Vertical: Copy row.	8 × 8 copy
Mode 1	Horizontal: Copy column.	8 × 8 copy
Mode 2	DC: Average of 32 values of macroblock.	(16-1) add + 1 div + 8 × 8 copy
Mode 3	Plane: A linear plane function fitted to the upper and left-hand samples H and V.	8 × 8 × (5 add + 2 mul + 1 compare + 1 clip)

where N_{l4}, N_{l16}, and N_c are the average number of operations in each of Tables 3, 4, and 5, respectively. They represent the number of operations to compute a 4 × 4 Luma prediction block, a 16 × 16 Luma prediction block, and an 8 × 8 Chroma prediction block. For example, $N_{l16} \times 4$ is the total number of operations in Table 4 or the average number of operations in the table multiplied by 4. In addition, $N_{l4} \times 9 \times 16$ is the total number of operations in Table 3 multiplied by 16, where 16 is the number of 4 × 4 blocks in the MB. Finally, $4N_c$ is the total number of operations in Table 5. The total number of operations in each of these three tables is constant, and thus N_{l4}, N_{l16}, and N_c are constants. Consequently, the intramode selection complexity can be simply given by

$$X_{intra} = c_{intra} L. \tag{13}$$

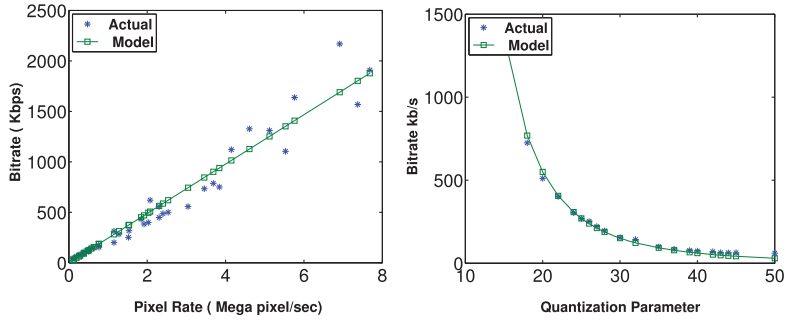
3.2.3 Quantization, Pixel Rate, and Bitrate Relationships. For homogeneous video contents, we determine by extensive experiments that the bitrate is linearly proportional to the pixel rate L and inversely proportional to the quantization parameter to a certain power, as shown in Figure 4. The used experimental setup is discussed in Section 4. (*R-Squared* represents the coefficient of determination.) Hence, the bitrate r can be expressed as

$$r = c_{rate} \frac{L}{q^c}, \tag{14}$$

where q is the quantization parameter, and c and c_{rate} are constants. As expected, the quantization parameter has a great impact on the bitrate.

3.2.4 Computational Complexities of Transformation, Quantization, Their Inverses, Reconstruction, Distortion, and Cost. Based on (He et al. 2005), the computation complexity $X_{traquant}$ to encode the residual MB (including transformation, quantization, and entropy coding), decode the MB (including inverse quantization and inverse transformation), reconstruct the MB, compute distortion, and compute the cost J can be expressed as

$$X_{traquant} = F\, x_{nzmb}\, m_{nzmb}, \tag{15}$$



(a) Bitrate vs. Pixel Rate (R-Squared = 0.947) (b) Bitrate vs. QP (R-Squared = 0.995)

Fig. 4. Relationships between bitrate and pixel rate and between bitrate and quantization parameter [Experimental Setup I].

Table 6. Nonzero MB's Complexity X_{nzmb} of Transform, Quantization, Inverse Quantization, Inverse Transform, Reconstruction, Distortion, and Cost

Step	Description	Operations (#)
Transform	Number of ops to compute transform ($Y = AXA^T$): 1 transpose and 2 4×4 matrix multiplications by blocks in macroblock	(16 copy + $2 \times$ (4 mul + 3 add) $\times 4 \times 4$ entries) $\times$ 16
Quantization	Number of ops to compute quantization	(4 mul + 3 add) $\times 4 \times 4 \times$ 16 entries
Inverse quantization	Number of ops to compute inverse quantization	(4 mul + 3 add) $\times 4 \times 4 \times 16$ entries
Inverse transform	Number of ops to compute inverse transform $Z = A^T YA$: one transpose and two 4×4 matrix multiplications by blocks in macroblock	(16 copy + $2 \times$ (4 mul + 3 add) $\times 4 \times 4$ entries) $\times$ 16
Reconstruction	Number of ops to compute the reconstructed macroblock	(4×4 add) $\times$ 16
Distortion	Number of ops to compute distortion and the sum of squared distortion (SSD) between the original and the reconstructed macroblock	(4×4 add + 4×4 mul + $(4 \times 4 - 1)$ add) $\times$ 16
Single cost	Number of ops to compute the single cost for the mode combination: $J = t + gr$	1 add + 1 mul
Minimum cost for Intraprediction	Number of ops to find the minimum cost among all mode combinations for the macroblock	(1 initialize + $592 \times$ (1 compare + 1 equal)) $\times$ 16
Minimum cost for interprediction	Number of ops to find the minimum cost among all mode combinations for the macroblock	(1 initialize + $41 \times$ (1 compare + 1 equal)) $\times$ 16

where F is the frame rate; m_{nzmb} represents the number of nonzero MBs in the video frame; and x_{nzmb} is the computation complexities of the transform, quantization, and their inverses for one nonzero MB. Note that a nonzero MB is an MB that has nonzero transform coefficients after quantization. Only nonzero MBs go through the transformation and quantization processes. Also note that x_{nzmb} is a constant because it is a systematic algorithm with a specified number of operations (Table 6) and $F \times m_{nzmb}$ is directly proportional to the bitrate. Therefore,

$$X_{traquant} = c_{traquant}\, r = c_{traquant}\, c_{rate}\, L/q^c = c_{qnt}\, L/q^c, \tag{16}$$

where $c_{traquant}$ and c_{rate} are constants and r is the bitrate. The video content coupled with the encoding algorithm and parameters (e.g., quantization) impact the number of nonzero MBs.

We develop Table 6 to determine the complexities of various steps. In the table, the steps involving transform, quantization, inverse quantization, inverse transform, reconstruction, distortion, and single cost are repeated either 592 times in case of intraprediction or 41 times in case of interprediction.

3.2.5 Entropy Computation Complexity. The entropy complexity $X_{entropy}$ of a frame is linearly proportional to bitrate (He et al. 2005). Based on Equation (14), we can express it as

$$X_{entropy} = c_{entropy}\, r = c_{entropy}\, c_{rate}\, L/q^c = c_{bit}\, L/q^c. \tag{17}$$

3.2.6 Overall Power Consumption Model. Based on Equations (5), (11), (13), (16), and (17), the complexity X_e of encoding F frames can be expressed as

$$\begin{aligned} X_e &= c_{integer}\, L\, R\, S^2 + (c_{fractional} + c_{intra})\, L + (c_{qnt} + c_{bit})\, L/q^c \\ &= c_{integer}\, L\, R\, S^2 + c_L\, L + c_{Lq}\, L/q^c, \end{aligned} \tag{18}$$

where $c_{integer}$, $c_{fractional}$, c_{intra}, c_{qnt}, c_{bit}, c_L, and c_{Lq} are constants; L is the pixel rate; R is the number of references; and S is the search range.

Let us now discuss how the overall power consumption can be modeled in terms of encoding complexity. As in He et al. (2005), the power consumption W_e for the encoder can be expressed as $W_e = c_{eff}\, v_{DVS}^2\, f_{CLK}$, where v and f_{CLK} are supply voltage and clock frequency, and c_{eff} is the effective switched capacitance of a processor with an energy-scaling feature, such as DVS (discussed in Section 2.2). However, v is approximately linearly proportional to f_{CLK}. As in Burd and Brodersen (1996), the voltage (v_{DVS}) and clock frequency (f_{CLK}) relationship is given by $v_{DVS} = c_1\, f_{CLK} + c_2$, where c_1 and c_2 are constants. Moreover, f_{CLK} is proportional to the computation complexity: $f_{CLK} = c_3\, X_e + c_4$, where c_3 and c_4 are constants. Subsequently, the power consumption can be expressed as

$$\begin{aligned} W_e &= c_{eff}\, (c_a\, X_e + c_b)^2\, (c_d\, X_e + c_e) \\ &= \left(\left(c_1\, R\, S^2 + \frac{c_2}{(q + c_q)^c} + c_3\right) L + c_4\right)^2 \left(\left(c_5\, R\, S^2 + \frac{c_6}{(q + c_q)^c} + c_7\right) L + c_8\right), \end{aligned} \tag{19}$$

where c_a, c_b, c_d, c_e, c_q, c_1, c_2, c_3, c_4, c_5, c_6, c_7, and c_8 are constants.

From Equation (19), we notice that the consumed encoding power depends on the video parameters (spatial and temporal resolutions), video content, performed algorithms (for intra- and interprediction, transform, quantization, etc.), and encoding parameters (the fraction of various frame types in the GOP, number of reference frames, quantization, and ME range, etc.). The video content coupled with the encoding algorithm and parameters (e.g., quantization parameter) impact the number of nonzero MBs. The model considers the full search approach and does not directly capture optimization techniques that abort the search early based on some statistics and other algorithms, such as fast intra-/interprediction. The computation complexity of the transform and quantization and their inverses is directly proportional to the number of nonzero MBs in the frame, which is directly proportional to the bitrate and inversely with quantization parameter (Sun and Pao 1998; Wang et al. 2004b). The complexity of entropy encoding is directly proportional to the bitrate (He et al. 2005). Furthermore, the loop filter complexity is a function of the number of MBs and frame rate. This leads us to conclude that the H.264 complexity is directly proportional to a weighted sum of the pixel rate and the bitrate.

3.2.7 General Bitrate Model. The bitrate is a function of the pixel rate, quantization parameter, number of references, and ME search range. Based on Equation (14) and extensive experiments analyzing the impacts of the number of references and the ME range (including those shown later in Figures 8(b) and 9(b)), we can develop a general model for bitrate as a function of the pixel rate, quantization parameter, number of references, and ME search range:

$$r = c_n \frac{(c_t - c_s R)\,(c_g - c_f S)\,L}{(q + c_q)^c}, \tag{20}$$

where c_n, c_t, c_s, c_g, c_f, and c_q are constants; L is the pixel rate; R is number of references; S is the ME search range; and q is the quantization parameter. The linear relationship with R and S will be evident in the validation results.

3.3 Modeling of the Power Consumed by Video Transmission

In the last phase of live video streaming, the video is transmitted to the receiver(s). According to Bhardwaj and Chandrakasan (2002) and He and Wu (2006), the power W_t consumed in wireless transmission when it is chosen such that the bit error rate (BER) at the receiver side is very low can be expressed as

$$W_t = (c_x + c_z\, d^n)\, r, \tag{21}$$

where c_x and c_z are wireless model constants, d is the transmission distance, n is the path-loss index, and r represents the transmission bitrate. Equation (21) can be generalized for wired transmission by assuming that the path-loss index n is zero. Therefore, the transmission power for wired transmission can be given by

$$W_{wired} = c\, r, \tag{22}$$

where c is a wire model constant and r is the transmission bitrate. Equations (21) and (22) indicate that the power consumption of transmitting the video is linearly proportional to the transmission bitrate.

In our experiments of wireless video transmission, we confirmed that the model in Equation (21) applies but with an additional constant, specifying the power consumption of the wireless circuit when no transmission takes place. For the same technology, platform, distance, path-loss, or environment, the model can be simplified as follows:

$$W_t = (c_x\, r + c_y), \tag{23}$$

where c_x and c_y are constants.

3.4 Modeling the Aggregate Power Consumption

Equations (4), (19), and (23) can be used to construct the aggregate power consumption model for the video source. The aggregate power consumed W_{agg} as a function of the resolution and frame rate can be found as follows:

$$W_{agg} = (c_{ap}\, L + c_{bp})^2\, (c_{af}\, L + c_{bf}) + c_i\, L + c_j + c_x\, r + c_y.$$

Using Equation (14), the model can be simplified to

$$W_{agg} = (c_{ap}\, L + c_{bp})^2\, (c_{af}\, L + c_{bf}) + c_{cp}\, L + c_{dp}, \tag{24}$$

where c_{ap}, c_{bp}, c_{af}, c_{bf}, c_{cp}, and c_{dp} are constants. Table 7 illustrates the physical significance of various constants in the aggregate power consumption model. Only the main factors are considered.

Table 7. Physical Significance of Various Constants

Constants	Reflect the Power Consumption of:
c_{ap}, c_{af}	DVS circuit capacitance, encoding parameters, encoding power consumption per pixel, and video content
c_{bp}, c_{bf}	Slope of linear relationship between frequency and voltage in DVS circuits and DVS capacitance
c_{cp}	Wireless distance, environment, transmission scheme, and capturing power consumption per pixel
c_{dp}	Power consumed by video sensor and transmitter circuits when they are not active

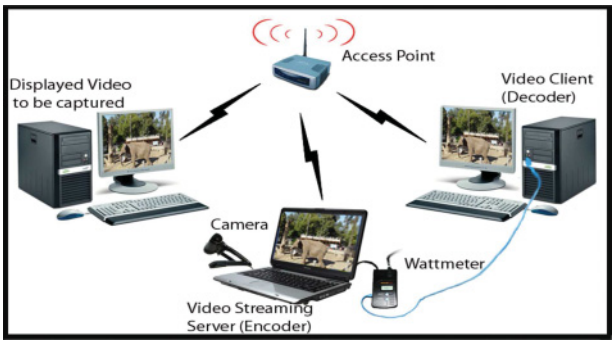


Fig. 5. Illustration of Experimental Setup I.

4 EXPERIMENTAL SETUPS AND VALIDATION METHODOLOGY

We validate the developed models through extensive experiments using two types of systems. The first uses a regular video camera and employs software-based encoding using FFmpeg/x264. This system allows better flexibility in conducting the experiments. The second includes an actual video surveillance camera with an SoC for encoding. We conduct experiments using three experimental setups. Experimental Setup I and Experimental Setup III are based on the first system, whereas Experimental Setup II is based on the latter, but the input videos for these two vary. In all setups, the power consumption is measured by an advanced power meter: Watts Up? Pro ES AC.

Figure 5 shows Experimental Setup I. To ensure repeatable measurements, a video rendered on a desktop computer is captured by a Dell Inspiron 1525 laptop computer with an Intel Core 2 Duo CPU (Model T5750) running at 2GHz with 3GB memory, 802.11n Wireless LAN, Ethernet LAN, and an external video camera (Logitech Webcam Pro 9001). The external camera is directed to that desktop computer, which plays a specific movie (from the beginning to the end). The rendered video includes scenes of five children running and playing in a zoo, with much details and fast movements (Sarhan 2017). The camera feeds the captured video in raw format to the laptop computer, which encodes the video with FFmpeg in the case of MPEG-4 and x264 in the case of H.264. The video is streamed using VideoLan VLC streaming server (Version 1.0.5) running on the computer. For validation, we also include some results using the latest VLC version (VLC 2.2.4) on the same platform. The distance between the server and client is within 1m.

We measure the power consumed by the streaming server for H.264 and MPEG-4 encoding. For encoding, we vary the spatial (i.e., frame size) and temporal (i.e., frame rate) resolutions generally from 160×120 up to $1{,}280 \times 720$ and from 1 to 30 fps, respectively. For H.264, we also study the

Table 8. Characteristics of the Standard Video Sequences Used in Experimental Setup III

Sequence	Duration (s)	Resolution	Frames (#)
Silent	10	CIF	300
Akiyo	10	CIF	300
Deadline	45.8	CIF	1,374
SignIrene	18	CIF	540

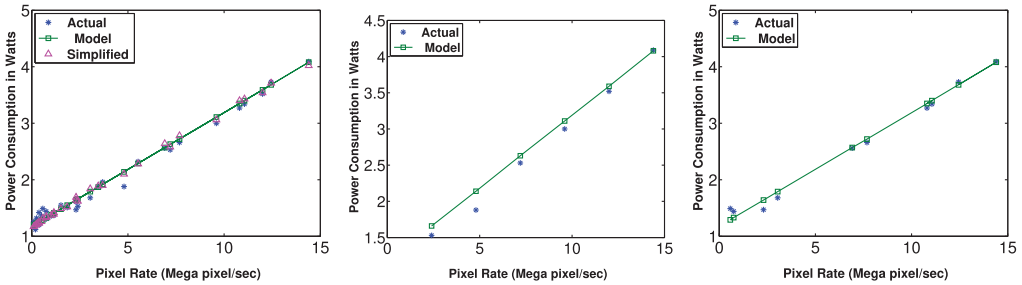
effect of varying the quantization parameter, the number of references, and ME range. In each experiment, the video is played for 22 minutes and 41 seconds. The reported power is the average power consumption during the entire video period. Each reported value is the average of 1,361 power readings, each of which is obtained during 1 second of the video. We assume the default encoding parameters in both x264 and FFmpeg except for those that are under study. Specifically, in validating the model, we assume the following values if they are not under study: $R = 3$, $S = 16$, $q = 22$, scaling factor $s = 22$, $F = 30$, $N \times M = 352 \times 288$, and maximum $L = 3041280$ pixel/s.

To minimize the effect of other processes while running the experiments, we run the computer with a bare minimum set of processes and drivers. In addition, each experiment is repeated four times, then the overall results are averaged. Furthermore, the power consumption due to other system processes running on the laptop computer is measured before each experiment and then is subtracted from the total power consumption. We initially measure the aggregate power of the three phases. To separate the power consumption due to each phase, we follow the following procedure. First, we measure the power consumption of only capturing and encoding and then subtract it from the aggregate power to get the transmission power consumption. Second, we stream the stored video (thereby no capturing is involved) from the laptop computer to the destination, measure the power consumption for this task, and then subtract the amount from the aggregate power consumption to get the capturing power consumption. Third, we subtract the capturing and the transmission power consumption from the aggregate power consumption to get the encoding power consumption.

In Experimental Setup II, we use a CMOS networked surveillance camera for further model validation. The used camera is Vivotek IP7139, which supports MPEG-4 and has built-in 10/100Mbps Ethernet and 802.11b/g WLAN. The distances between the camera and the monitoring station is within 1m. As the differences in power consumption for different temporal and spatial settings can be in a fraction of a watt, we capture a TV channel with the camera for an average of 10 hours in each experiment. The captured video is streamed to a desktop computer using a built-in streaming server that is supplied by the camera's manufacturer. The reported power consumption is the average of 36,000 power readings during the recording and streaming period. We experiment with both wired and wireless transmission.

In Experimental Setup III, we conduct experiments to further study and validate the impact of changing both the resolution and quantization/bitrate on encoding power consumption. This setup has the same system as Experimental Setup I, but the average results of four standard video sequences are reported: Silent, Akiyo, Deadline, and SignIrene, as described in Table 8. We downscale each video sequence from the original size down to 10% (specifically, we consider 100%, 90%, . . . , 10% of the original size). For each of these sizes, we also produce different quality levels by varying the quantization parameter (from 1 to 31). We measure the power consumption while encoding and then find the bitrate of the encoded video.

With Experimental Setup III, we also analyze the power consumption at the monitoring station of many-to-one video streaming systems due to receiving, decoding, and upscaling. Additionally,



(a) Spatial and Temporal Effects (R-Squared = 0.987)

(b) Temporal Effect at 800×600 (R-Squared = 0.976)

(c) Spatial Effect at 30 fps (R-Squared = 0.988)

Fig. 6. Validation of video capturing power consumption [Experimental Setup I, $c_i = 2.014.10^{-7}$ watt/pixel, $c_j = 1.175$ watt].

we analyze the quality of the received videos. As discussed earlier, upscaling the video greatly improves the video quality. We use the decoded frames to measure the quality compared to the original video. As a metric for perceptual video quality, we use the Structural SIMilarity Index (SSIM) (Wang et al. 2004a) between two images. It improves the popular peak signal-to-noise ratio (PSNR) metric by considering the similarity of the edges between the two images being compared, and it is more consistent with human visual perception. Since the human eye is more responsive to brightness than to color, we use only the Luma (Y) components in the YUV color space. SSIM provides a quality reconstruction metric that considers the similarity of the edges between the produced image and the ideal one, whereas other metrics are based on computing the mean squared reconstruction error.

5 MODEL VALIDATION RESULTS AND ANALYSIS

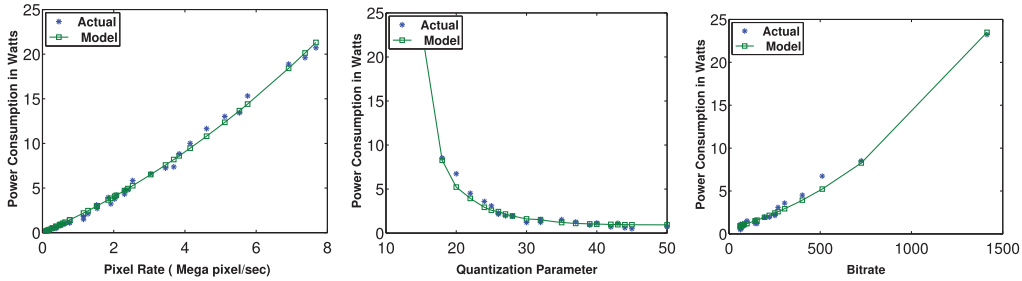
To analyze the goodness of the fit for the developed models, we use and report the *coefficient of determination* (R-Squared). The main raw data can be found in Sarhan (2017). The results for Experimental Setup I are shown first. For this setup, VLC 1.05 is used unless otherwise indicated. In addition, wireless transmission is assumed unless otherwise indicated.

5.1 Validation of the Capturing Model

Figure 6(a) validates the developed capturing model (Equation (3)) and the simplified capturing model (Equation (4)) when both the spatial and temporal resolution are varied. The results show that the model in both forms accurately represents the real behavior. Figure 6(b) and (c) validate the model when only the temporal resolution or spatial resolution is varied, respectively.

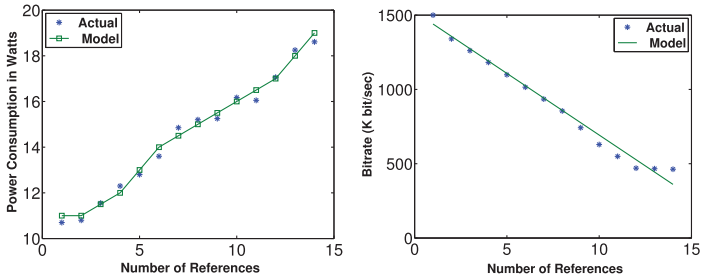
5.2 Validation of the Power Consumption and Bitrate Models of H.264 Encoding

Figure 7 validates the developed power consumption model for encoding (Equation (19)) for variable frame sizes, frame rates, and quantization parameters. The bitrate in Figure 7(c) is varied by changing the quantization parameter. Note that the quantization parameter has a great impact on power consumption and bitrate. Figure 8(a) and (b) validate the power consumption model (Equation (19)) and the bitrate model (Equation (20) as the number of reference frames is varied), respectively. Similarly, Figure 9(a) and (b) validate the power consumption and bitrate models as the ME range is varied, respectively. The inverse linear relationship of the bitrate with the number of reference frames and ME range is clearly evident.



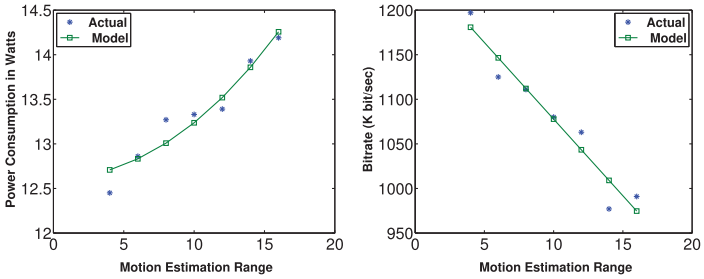
(a) Spatial and Temporal Effects (R-Squared = 0.995) (b) Consumed Power vs. QP (R-Squared = 0.991) (c) Consumed Power vs. Bitrate (R-Squared = 0.991)

Fig. 7. Validation of the impact of the spatial, temporal, and quantization parameter on encoding power consumption in H.264 (Experimental Setup I).



(a) Consumed Power (R-Squared = 0.988) (b) Bitrate (R-Squared = 0.983)

Fig. 8. Validation of the impact of number of reference frames in H.264 (Experimental Setup I).



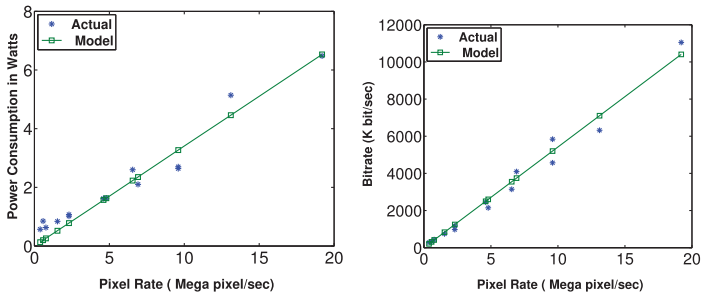
(a) Consumed Power (R-Squared = 0.919) (b) Bitrate (R-Squared = 0.932)

Fig. 9. Validation of the impact of ME range in H.264 (Experimental Setup I, TESA algorithm).

Table 9 shows the constant values for the general power consumption model of H.264 (Equation (19)) using Experimental Setup I. For the general bitrate model (Equation (20)), the constant values on Experimental Setup I are as follows: $c_n = 0.0124$, $c = 3.16$, $c_g = 1249.5$, $c_f = 17.18$, $c_t = 1523.36$, $c_s = 83.03$, and $c_q = 0$.

Table 9. Constant Values for the H.264 Power Consumption Model (Experimental Setup I)

Constant	Value	Constant	Value	Constant	Value
c_1	$7.437.10^{-9}$	c_2	$4.8.10^{-5}$	c_3	$7.96.10^{-11}$
c_4	$2.3392.10^{-4}$	c_5	$1.58.10^{-6}$	c_6	$1.46.10^{-3}$
c_7	$1.71.10^{-8}$	c_8	0	c	3.16
c_q	0				



(a) Consumed Power (R-Squared = 0.940)

(b) Bitrate (R-Squared = 0.980)

Fig. 10. Validation of the spatial and temporal effects on MPEG-4 encoding power consumption (Experimental Setup I).

Table 10. Constant Values for the MPEG-4 Power Consumption and Bitrate Models (Experimental Setup I)

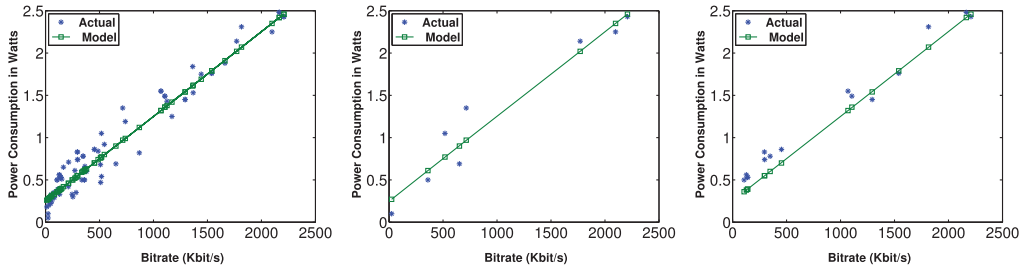
Constant	Value	Constant	Value	Constant	Value
c_1	$16.77.10^{-9}$	c_2	$9.86.10^{-7}$	c_3	0
c_4	10.0	c_5	$2.16.10^{-8}$	c_6	$9.86.10^{-7}$
c_7	0	c_8	0	c	0.5
c_q	3.0	c_n	$2.7.10^{-3}$	c_g	1249.5
c_f	17.18	c_t	1523.36	c_s	1522.36

5.3 Validation of the Power Consumption and Bitrate Models of the MPEG-4 Encoder

Although the encoding power consumption and bitrate models (Equations (19) and (20)) are developed for H.264 due to its popularity and efficiency, they can apply to MPEG-4, which shares most of the features with H.264 and follows a similar operation structure. The main differences are discussed in Section 2. H.264 performs many more operations by allowing more reference frames, examining more intramodes, working with smaller block sizes, and so forth. None of these, however, impacts the model, as they are only quantitative in nature. The developed tables and equations for the numbers of operations for intramode selection do not hold for MPEG-4, but the complexity model remains the same. Figure 10 demonstrates that both the developed models apply for MPEG-4, but with different constants. Table 10 shows the constant values for Experimental Setup I.

5.4 Validation of the Transmission Model

Figure 11(a) validates the developed transmission model when both the spatial and temporal resolutions are varied, whereas Figure 11(b) and (c) show the results when varying only the temporal

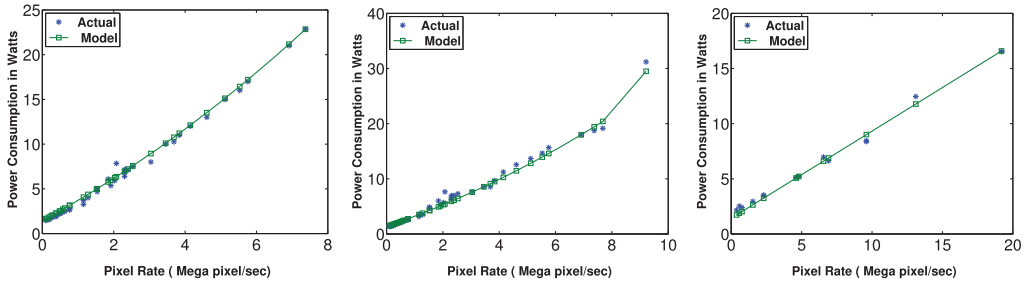


(a) Transmission: Spatial and Temporal Effects (R-Squared = 0.942)

(b) Temporal Effect at 800×600 (R-Squared = 0.953)

(c) Spatial Effect at 30 fps (R-Squared = 0.945)

Fig. 11. Validation of the transmission power consumption model (Experimental Setup I, $c_x = 0.001$ watt/bit, $c_y = 0.25$ watt).



(a) H.264 with VLC 1.0.5, (R-Squared = 0.994)

(b) H.264 with VLC 2.2.4 (R-Squared = 0.982)

(c) MPEG-4 (R-Squared = 0.990)

Fig. 12. Validation of the aggregate power consumption model (Experimental Setup I).

or spatial resolution, respectively. The observed variations from the actual experimental results are primarily due to measurement errors as the power consumed in transmission is low when compared to other phases.

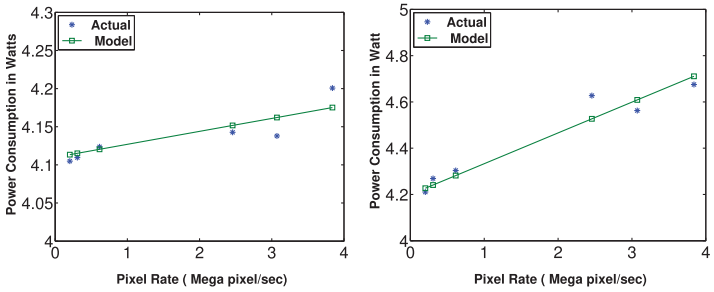
5.5 Validation of the Aggregate Power Consumption Model

Figure 12 validates the aggregate power consumption model (Equation (24)) using Experimental Setup I (with regular camera and software-based encoding) for both H.264 and MPEG-4. The results for H.264 are shown for both VLC streaming server 1.0.5 and 2.2.4. These results demonstrate the accuracy of the model and that it applies for H.264, MPEG-4, and different versions of H.264 encoders, but with different constant values.

Figure 13 shows the validation results using Experimental Setup II, which uses the CMOS surveillance camera. The results for both wired and wireless transmission are shown. As expected, wireless transmission consumes more power than wired. The constants for wired are $c_{ap} = 0$, $c_{bp} = 1 \times 10^{-8}$, $c_{af} = 2 \times 10^{-12}$, $c_{bf} = 0.2$, $c_{cp} = 1 \times 10^{-8}$, and $c_{dp} = 4.077$, and for wireless are $c_{ap} = 0$, $c_{bp} = 5.5 \times 10^{-8}$, $c_{af} = 2 \times 10^{-12}$, $c_{bf} = 0.5$, $c_{cp} = 5.5 \times 10^{-8}$, and $c_{dp} = 4.155$.

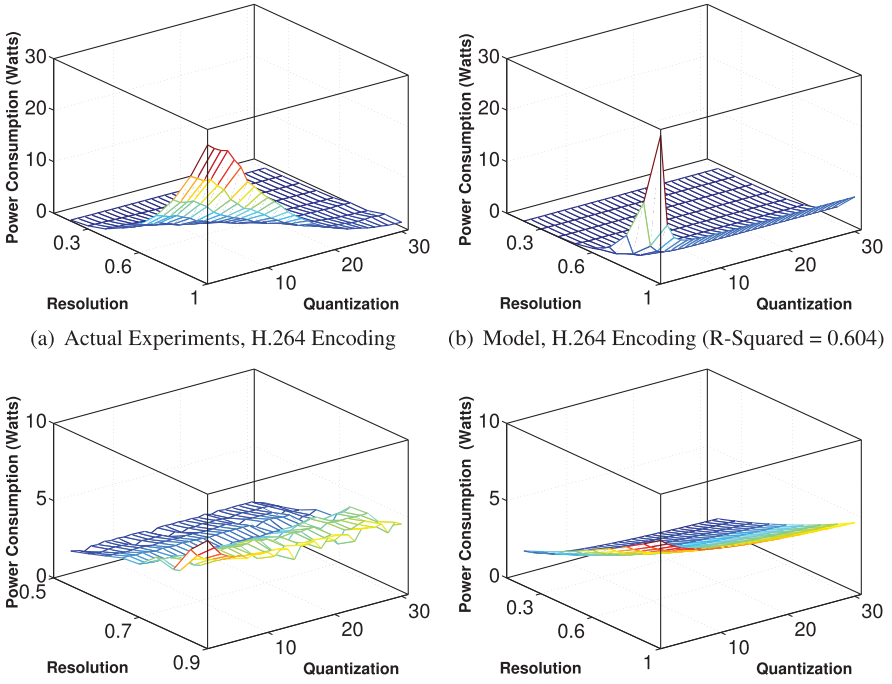
5.6 Further Validation and Analysis by Varying Both the Spatial Resolution and Quantization

Since the spatial resolution and quantization parameter are major contributors to encoding complexity, power consumption, and bitrate, let us analyze the overall behavior and validate the



(a) Wired Tx (R-Squared = 0.764) (b) Wireless Tx (R-Squared = 0.927)

Fig. 13. Further validation of the aggregate power consumption model (Experimental Setup II, MPEG-4).

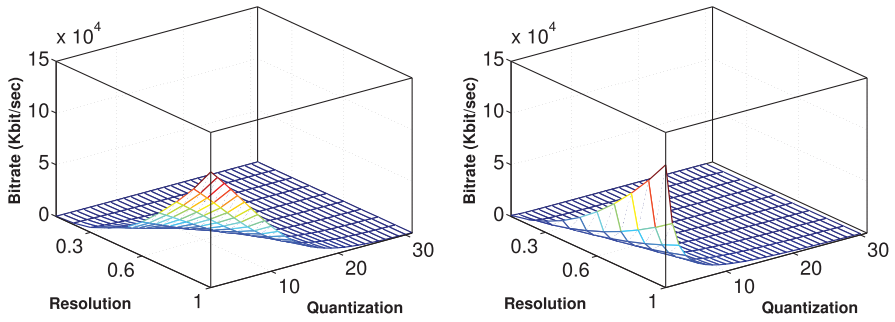


(a) Actual Experiments, H.264 Encoding (b) Model, H.264 Encoding (R-Squared = 0.604)

(c) Actual Experiments, MPEG-4 Encoding (d) Model, MPEG-4 Encoding (R-Squared = 0.776)

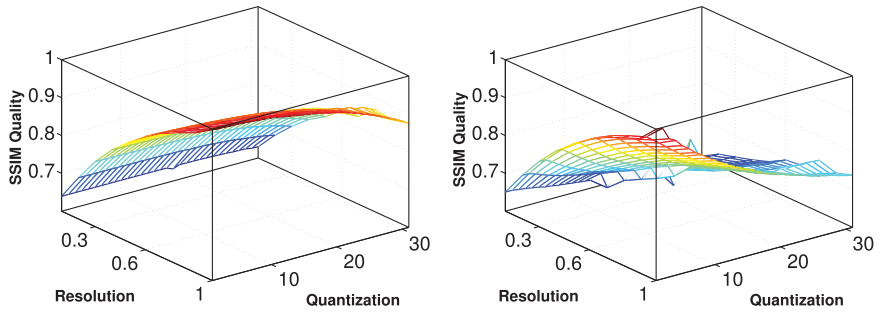
Fig. 14. Effect on power consumption by varying quantization and resolution (Experimental Setup III).

developed models when varying both parameters at the same time. Figures 14 and 15 illustrate the overall impacts of spatial resolution and quantization parameter on the encoding power consumption and achieved bitrate, respectively, and further validate the developed models. Similarly, Figure 16 shows the SSIM video quality results by comparing the decoded and the original videos. These results demonstrate that downscaling the spatial resolution before transmission and then upscaling to the original resolution by the monitoring station can significantly reduce power consumption and bitrate without having a considerable impact on video quality for a wide range of downscaling levels. By combining quantization and spatial resolution adaptations in H.264 encoding, the bitrate is reduced to 1% of the original bitrate and the consumed power is reduced



(a) Actual Experiments, H.264 Encoding (b) H.264 Encoding Model (R-Squared = 0.767)

Fig. 15. Effect on bitrate by varying quantization and resolution (Experimental Setup III).



(a) H.264 Encoding (b) MPEG-4 Encoding

Fig. 16. Effect on SSIM quality by varying quantization and resolution (Experimental Setup III).

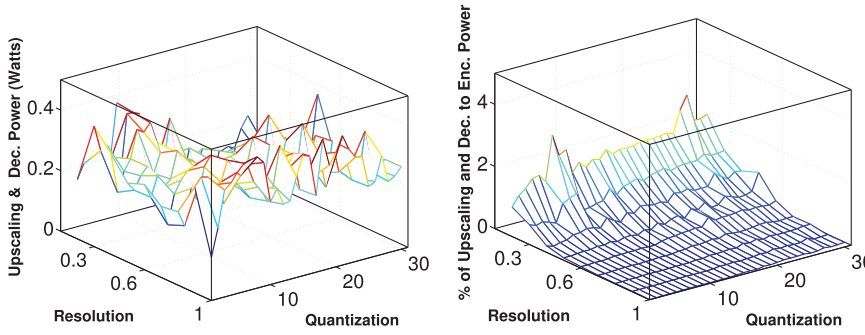
to 4% of the original, while reducing the quality to only 88% of the original. For MPEG-4, the bitrate is reduced to 1% and the power is reduced to 45%, while keeping the quality higher than 78% of the original.

5.7 Analysis of Power Consumption by the Monitoring Station in Many-to-One Video Streaming Systems

Let us now analyze the power consumed by the monitoring station for receiving, decoding, and upscaling the received video streams. Figure 17(a) shows the consumed power, whereas Figure 17(b) shows the percentage of power consumption for handling one stream by the monitoring station to the encoding power consumed by the source. Note that the consumed power is smaller than 0.5 watt per stream, and the percentage of the consumed power relative to the encoding power consumption is smaller than 2% for the spatial resolution of half the original and quantization parameter smaller than 20.

6 CONCLUSIONS AND FUTURE WORK

We have developed an aggregate power consumption model for live video streaming systems. The model can help in the dynamic control of various camera/sensor settings, including resolution, frame rate, and quantization, to achieve the best overall trade-off in terms of power consumption, bitrate, and quality. Specifically, we have modeled the video capturing, encoding, and transmission aspects and then have provided an overall model of the power consumed by the video sources. We



(a) Consumed Power (b) %Consumed Power Relative to Encoding

Fig. 17. Power consumption by the monitoring station (Experimental Setup III).

have also analyzed the power consumed by the monitoring station in many-to-one systems due to receiving, decoding, and upscaling the received video streams. In addition, we have analyzed the perceived quality at the monitoring station. Moreover, we have modeled the output bitrate of video encoding. Furthermore, we have validated the developed models through extensive experiments using two different systems and different video contents.

The main conclusions can be summarized as follows. First, the overall computation complexity for all phases can approximately be modeled as a linear function of the pixel rate when varying only the frame rate and frame size. Second, for high spatial and/or temporal resolution, the video encoding consumes more than 90% of the power, whereas capturing consumes less than 6% and transmission less than 4%. Third, H.264 consumes more than three times the power consumed by MPEG-4 in the considered software-based encoding system. Fifth, the quantization parameter affects power consumption in an exponential fashion. Sixth, other encoding parameters, such as the number of references and the ME search range, vary the power consumption by up to 10%. Seventh, the tuning of parameters must be done based on power consumption, video quality, and bitrate trade-offs. Eighth, the complexities of interprediction, intraprediction, RDO mode selection, and subpixel search can be expressed as a linear function of the pixel rate. Similarly, the aggregate power consumption is a linear function of the pixel rate. Ninth, by combining quantization and spatial resolution adaptations in H.264 encoding, the bitrate is reduced to 1% of the original bitrate and the consumed power is reduced to 4% of the original, while reducing the quality to only 88% of the original. For MPEG-4, the bitrate is reduced to 1% and the power is reduced to 45%, while reducing the quality to only 78% of the original. Tenth, the power consumed by upscaling and decoding one stream by the monitoring station is smaller than 0.5 watt per stream in the considered system. The percentage of this power relative to the encoding power consumption is smaller than 2% for a spatial resolution of half the original and a quantization parameter smaller than 20.

In future work, we will adapt the encoding model to other encoders, including HEVC and VP9. We will also develop an objective function for video adaptations that helps in achieving any desired trade-off among power consumption, bandwidth, and accuracy.

REFERENCES

Mohammad Alsmirat and Nabil J. Sarhan. 2016. Cross-layer optimization for automated video surveillance. In *Proceedings of the IEEE International Symposium on Multimedia (ISM'16)*. 243–246.

Manish Bhardwaj and Anantha P. Chandrakasan. 2002. Bounding the lifetime of sensor networks via optimal role assignments. In *Proceedings of IEEE INFOCOM*, Vol. 3. 1587–1596.

Thomas D. Burd and Robert W. Brodersen. 1996. Processor design for portable systems. *Journal of VLSI Signal Processing Systems* 13, 2, 203–222.

Rei-Heng Cheng and Chiming Huang. 2013. The impact of the transmission power range on energy consumption for wireless sensor networks. In *Proceedings of the International Conference on Ubiquitous and Future Networks (ICUFN'13)*. 77–81.

Huseyin Cotuk, Kemal Bicakci, Bulent Tavli, and Erkam Uzun. 2014. The impact of transmission power control strategies on lifetime of wireless sensor networks. *IEEE Transactions on Computers* 63, 11, 2866–2879.

Abdelhafid Elouardi, Samir Bouaziz, Antoine Dupret, Lionel Lacassagne, Jacques-Olivier Klein, and Roger Reynaud. 2007. Image processing vision systems: Standard image sensors versus retinas. *IEEE Transactions on Instrumentation and Measurement* 56, 5, 1675–1687.

Wu-Chi Feng, Ed Kaiser, Wu Chang Feng, and Mikael Le Baillif. 2005. Panoptes: Scalable low-power video sensor networking technologies. *ACM Transactions on Multimedia Computing, Communications and Applications* 1, 2, 151–167.

Zhihai He, Yongfang Liang, Lulin Chen, Ishfaq Ahmad, and Dapeng Wu. 2005. Power-rate-distortion analysis for wireless video communication under energy constraints. *IEEE Transactions on Circuits and Systems for Video Technology* 15, 5, 645–658.

Zhihai He and Dapeng Wu. 2006. Resource allocation and performance analysis of wireless video sensors. *IEEE Transactions on Circuits and Systems for Video Technology* 16, 5, 590–599.

Mohammad Ashraful Hoque, Matti Siekkinen, Jukka K. Nurminen, Mika Aalto, and Sasu Tarkoma. 2015. Mobile multimedia streaming techniques: QoE and energy saving perspective. *Pervasive and Mobile Computing* 16, 96–114.

C. S. Kannangara, II. E. Richardson, and A. J. Miller. 2008. Computational complexity management of a real-time H.264/AVC encoder. *IEEE Transactions on Circuits and Systems for Video Technology* 18, 9, 1191–1200.

Changsung Kim and C.-C. Jay Kuo. 2007. Feature-based intra-/intercoding mode selection for H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology* 17, 4, 441–453.

Jongho Kim, Donghyung Kim, and Jechang Jeong. 2006. Complexity reduction algorithm for intra mode selection in H.264/AVC video coding. In *Proceedings of the Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS'06)*. 454–465.

Jaemoon Kim, Jungsoo Kim, Giwon Kim, and Chong-Min Kyoung. 2011. Power-rate-distortion modeling for energy minimization of portable video encoding devices. In *Proceedings of the IEEE International Midwest Symposium on Circuits and Systems (MWSCAS'11)*. 1–4.

Robert LiKamWa, Bodhi Priyantha, Matthai Philipose, Lin Zhong, and Paramvir Bahl. 2013. Energy characterization and optimization of image sensing toward continuous mobile vision. In *Proceedings of the ACM Annual International Conference on Mobile Systems, Applications, and Services (MobiSys'13)*. 69–82.

Weiyao Lin, Krit Panusopone, David M. Baylon, Ming-Ting Sun, Zhenzhong Chen, and Hongxiang Li. 2011. A fast sub-pixel motion estimation algorithm for H.264/AVC video coding. *IEEE Transactions on Circuits and Systems for Video Technology* 21, 2, 237–242.

Xiaoan Lu, Thierry Fernaine, and Yao Wang. 2004. Modelling power consumption of a H.263 video encoder. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'04)*. 77–80.

Wei Pu, Yan Lu, and Feng Wu. 2006. Joint power-distortion optimization on devices with MPEG-4 AVC/H.264 codec. In *Proceedings of the IEEE International Conference on Communications (ICC'06)*. 441–446.

Swaminathan Vasanth Rajaraman, Matti Siekkinen, and Mohammad A. Hoque. 2014. Energy consumption anatomy of live video streaming from a smartphone. In *Proceedings of the IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC'14)*. 2013–2017.

Iain E. G. Richardson. 2010. *The H.264 Advanced Video Compression Standard* (2nd ed.). Wiley.

Nabil J. Sarhan. 2017. Supplementary Information for Modeling and Analysis of Power Consumption in Live Video Streaming Systems. Retrieved July 11, 2017, from http://www.ece.eng.wayne.edu/~nabil/power_modeling/power.html.

Bambang A. B. Sarif, Mahsa Pourazad, Panos Nasiopoulos, and Victor C. M. Leung. 2015. A study on the power consumption of H.264/AVC-based video sensor network. *International Journal of Distributed Sensor Networks* 11, 304787:1–304787-10.

Muhammad Shafique, Bastian Molkenthin, and Jörg Henkel. 2010. An HVS-based adaptive computational complexity reduction scheme for H.264/AVC video encoder using prognostic early mode exclusion. In *Proceedings of the Design, Automation, and Test in Europe Conference and Exhibition*. 1713–1718.

Yousef O. Sharrab and Nabil J. Sarhan. 2012. Accuracy and power consumption tradeoffs in video rate adaptation for computer vision applications. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'12)*. 410–415.

Yousef O. Sharrab and Nabil J. Sarhan. 2013. Aggregate power consumption modeling of live video streaming systems. In *Proceedings of the ACM Multimedia Systems Conference*. 60–71.

Li Su, Yan Lu, Feng Wu, Shipeng Li, and Wen Gao. 2009. Complexity-constrained H.264 video encoding. *IEEE Transactions on Circuits and Systems for Video Technology* 19, 4, 477–490.

Ming-Ting Sun and I-Ming Pao. 1998. Statistical computation of discrete cosine transform in video encoders. *Journal of Visual Communication and Image Representation* 9, 2, 163–170.

Yih Han Tan, Wei Siong Lee, Jo Yew Tham, Susanto Rahardja, and Kin Mun Lye. 2010. Complexity scalable H.264/AVC encoding. *IEEE Transactions on Circuits and Systems for Video Technology* 20, 9, 1271.

Alexis M. Tourapis, Oscar C. Au, and Ming L. Liou. 2001. Predictive motion vector field adaptive search technique—enhancing block based motion estimation. In *Proceedings of the Visual Communications and Image Processing Conference*. 883–892.

Yingkun Wang, Yuanhua Zhou, and Hua Yang. 2004b. Early detection method of all-zero integer transform coefficients. *IEEE Transactions on Consumer Electronics* 50, 3, 923–928.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004a. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4, 600–612.

Xiaozhong Xu and Yun He. 2008. Improvements on fast motion estimation strategy for H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology* 18, 3, 285–293.

Ce Zhu, Xiao Lin, Lap-Pui Chau, Keng-Pang Lim, Hock-Ann Ang, and Choo-Yin Ong. 2001. A novel hexagon-based search algorithm for fast block motion estimation. In *Proceedings of Acoustics, Speech, and Signal Processing*, Vol. 3. 1593–1596.

Received September 2016; revised June 2017; accepted June 2017