A Rate-Energy-Distortion Analysis for Compressed-Sensing-Enabled Wireless Video Streaming on Multimedia Sensors

Scott Pudlewski, Tommaso Melodia
Department of Electrical Engineering
State University of New York (SUNY) at Buffalo
e-mail: {smp25, tmelodia}@buffalo.edu

***Abstract*—Real-time encoding and error-resilient wireless transmission of multimedia content require high processing and transmission power. This paper investigates the rate-distortion performance of video transmission over lossy wireless links for low-complexity multimedia sensing devices with a limited budget of available energy per video frame.**

An analytical/empirical model is developed to determine the received video quality when the overall energy allowed for both encoding and transmitting each frame of a video is fixed and the received data is affected by channel errors. The model is used to compare the received video quality, computation time, and energy consumption per frame of different wireless streaming systems. Furthermore, it is used to determine the optimal allocation of encoded video rate and channel encoding rate for a given available energy budget.

The proposed model is then applied to compare the energy-constrained wireless streaming performance of three encoders suitable for a wireless multimedia sensor network environment; H.264, motion JPEG (MJPEG) and our recently developed compressed sensing video encoder (CSV). Extensive results show that CSV, thanks to its low complexity, and to a video representation that is inherently resilient to channel errors, is able to deliver video at good quality (an SSIM value of 0.8) through lossy wireless networks with lower energy consumption per frame than competing encoders.

I. Introduction

Recent advances in sensing, computation, storage, and wireless networking are driving an increasing interest in multimedia [1] and people-centric [2] sensing applications. Wireless Multimedia Sensor Networks (WMSN) are self-organizing systems of embedded devices deployed to retrieve, distributively process in real-time, store, correlate, and fuse multimedia streams originated from heterogeneous sources [3]. WMSNs are enablers for applications including video surveillance, storage and subsequent retrieval of potentially relevant activities. In people-centric (also referred to as *participatory*) sensing, sensing devices, in the form of the ubiquitous mobile smart phones (e.g., Samsung Galaxy S, Iphone) are carried by individuals, thus enabling sensing, learning, visualizing, and sharing information about our daily activities.

The objective of this paper is to conduct an experiment-driven analysis of the energy-rate-distortion performance of different streaming systems designed for embedded wirelessly networked devices. Different from previous work on low-complexity encoding [4] [5], we jointly consider the effects of processing on resource-constrained devices and of wireless transmission on the performance of wireless encoders. We first develop an analytical model that can be manipulated to determine, for a given total energy budget per frame and channel condition, the optimal joint allocation of energy for wireless transmission and energy to be used for video encoding. Intuitively, for a fixed energy budget, as more energy is allocated to the encoder (resulting in less compression and a video of better quality), less energy is available to transmit that video over a wireless link, which would potentially result in an increased bit error rate and lower quality at the receiver. Conversely, as more energy is allocated to transmission, less energy is available to encode the video, resulting in a lower quality video. The developed model is used to find, for a given encoder, the optimal allocation between these two components, along with the optimal channel coding rate, which results in the optimal received video quality.

The model used in this paper is illustrated in Fig. 1. This model evaluates the video quality at the receiver of a wireless network for a given video encoder (and a specific encoding rate), family of channel codes (and specific channel encoding rate), and for a given energy budget per frame. The energy budget per frame is split between the energy needed for encoding the video and the energy required for transmitting the video over the lossy channel. These choices affect the quality of the received video at the multimedia sink. We present an optimization procedure to choose the encoded video rate and the channel coding rate that result in the optimal received video quality for a given energy budget. We then present a methodology, based on the model, to compare different encoders in terms of the energy budget required to obtain a target video quality at the receiver.

For our comparison, we focus on three video encoders with different characteristics mainly in terms of their complexity (and of the resulting rate-distortion performance). The first is Motion JPEG (MJPEG), which is a simple low-complexity encoder designed for low-power or portable devices. For this paper we use the implementation in [6]. MJPEG is a video encoder in which each frame is individually encoded according

This paper is based upon work supported in part by the Office of Naval Research under grant N00014-11-1-0848.

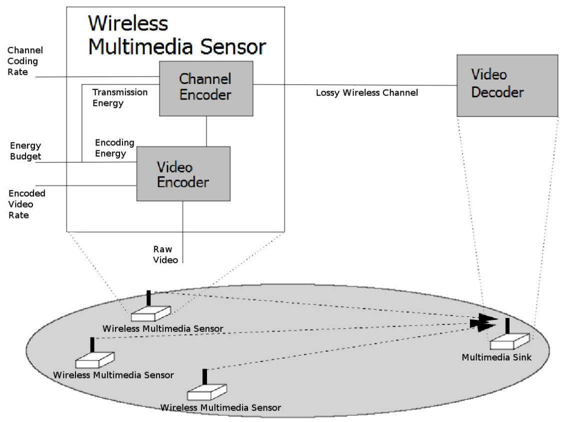

Fig. 1: Energy-Aware Video Encoding and Transmission

to the JPEG standard. Because it does not exploit dependency between frames, motion estimation between frames is not necessary, resulting in much lower encoding complexity.

Next, H.264/AVC [7], [8], [9] is considered, which represents the state of the art in video encoders. Though H.264 clearly offers the best video compression, it is a progressive video encoder, for which there are two major disadvantages in an energy constrained environment, i.e., *encoder complexity* and *low resiliency to channel errors*.

- **Encoder Complexity.** Modern predictive encoding algorithms were created with the goal of reducing the amount of data required to represent the video while maintaining the best possible video quality. Because of the need to find motion vectors, the complexity of these algorithms usually results in very high energy consumption and very high time to encode [1].
- **Limited Resiliency to Channel Errors**. Predictive encoding, as the name suggests, relies on the prediction of a frame based on previous frames. This allows for greatly reduced image size. However, since errors can propagate from one frame to the next, this also can result in very poor performance in lossy channels. This is usually combatted with forward error correction (FEC) which can increase the overall transmission energy needed to successfully deliver the video to the end receiver.

Finally, we consider a video encoder based on compressed sensing (CS) [10], [11], [12], presented in [13]. Compressed sensing (aka "compressive sampling") is a new paradigm that allows the faithful recovery of signals from $M << N$ measurements where N is the number of samples required for the Nyquist sampling. Hence, CS can offer an alternative to traditional video encoders by enabling imaging systems that sense and compress data simultaneously *at very low computational complexity for the encoder*. CS images and video are also resilient to bit errors [14]. Based on the low-complexity and high error resilience of CS signals, CSV was designed to achieve an acceptable level of compression with the lowest possible energy consumption at the video source.

The remainder of this paper is structured as follows. In Section II we present the three video encoders studied in this paper. In Section III we present the empirical models used to characterize the video encoders, and in Section IV we present the video quality model. Finally, the performance results of the three encoders are presented in Section V, while Section VI we draw the main conclusions and discuss future work.

II. Background

In this paper, we compare three video encoders that represent three different approaches to encoding.

A. Motion-JPEG (MJPEG) Video Encoder

MJPEG video encoding is an intraframe encoding scheme based directly on the JPEG image compression standard [15]. Though there is no official standardization of MJPEG, the basic concepts of most implementations are the same. Each frame is first divided into 8×8 blocks which are transformed to the frequency domain using the discrete cosine transform (DCT) [16], creating an 8×8 block of DCT coefficients.

From this point, the DCT coefficients of each macroblock are quantized and entropy encoded, resulting in a much smaller file than the original. The DCT transform naturally gives more weight to the low frequency components and less to the high frequency components. This corresponds well to the human visual system, which allows the encoder to remove many of the high frequency components with little effect on the perceived quality of the resulting image.

In MJPEG, this process is done for each frame independently. The resulting video has compression and quality comparable to JPEG image compression and can be done without significant complexity requirements at the encoder. These factors have made this protocol very useful in low complexity devices such as digital cameras.

B. H.264 Video Encoder

H.264 video compression represents the state of the art in current video compression techniques. Though a full explanation of H.264 is beyond the scope of this paper, we will present some basic concepts.

The basic functionality of the H.264 [8] encoder is similar to that of JPEG with a major addition, which is prediction. Along with the frequency transform - quantization - entropy encoding functionalities, the encoder will take an image block and compare it to other macroblocks either within the same frame (intra-prediction) or in a previous frame (inter-prediction). By finding the difference between two macroblocks and encoding that difference (which is generally very small), the encoder can greatly reduce the amount of data necessary to represent a video at a very good quality. For a full explanation of H.264 video encoding, the reader is referred to [7], [8], [9].

C. Compressed Sensed Video (CSV) Encoder

CSV [13] uses compressed sensing to take advantage of both the spacial correlation within a frame (intra-frame) and the temporal correlation between frames (inter-frame). For intra-frame encoding, a frame is represented by a vector $\mathbf{x} \in R^N$.

We assume that there exists an invertible $N \times N$ transform matrix $\mathbf{\Psi}$ such that

$$\mathbf{x} = \mathbf{\Psi}\mathbf{s} \tag{1}$$

where $\mathbf{s}$ is a K-sparse vector, i.e., $||s||_0 = K$ with $K < N$, and where $|| \cdot ||_p$ represents p-norm. This means that the image has a sparse representation in some transformed domain, e.g., wavelet. The signal is measured by taking $M < N$ measurements from linear combinations of the element vectors through a linear measurement operator $\mathbf{\Phi}$. Hence,

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{s} = \tilde{\mathbf{\Psi}}\mathbf{s}. \tag{2}$$

Although, in general, $\mathbf{x}$ can not be recovered directly, [11] shows that if the measurement matrix $\mathbf{\Phi}$ is sufficiently incoherent with respect to the sparsifying matrix $\mathbf{\Psi}$, and K is smaller than a given threshold (i.e., the sparse representation $\mathbf{s}$ of the original signal $\mathbf{x}$ is "sparse enough"), then the original $\mathbf{s}$ can be recovered by finding the sparsest solution that "matches" the measurements in $\mathbf{y}$.

To exploit this inter-frame redundancy within the framework of compressed sensing, we take the algebraic difference between the CS samples. Then, this difference is *again compressively sampled* and transmitted. If the image being encoded and the reference image are very similar (i.e. have a very high correlation coefficient), then this difference image will be sparser and have less variance than either of the original images, and can therefore be transmitted at the same quality using fewer samples and fewer bits per pixel than the original image.

For a full explanation of CSV video encoding, the reader is referred to [13].

III. Rate Distortion of Video Encoders in Unreliable Channels

In this section, we develop a model of the video quality (in SSIM [17]) after transmission over a noisy channel. We are interested in modeling the received video quality as a function of the encoded video rate r_v, the channel coding rate r_{ch}, the total energy budget E_B per frame and the channel quality.

To analyze the rate distortion performance of video encoders, we must first develop a model that accurately predicts the effect of compression and bit errors on the video quality. In a lossless channel, video distortion can be modeled [18] [19] as

$$\alpha(r_v) = D_0 - \frac{\Theta}{r_v - R_0}, \tag{3}$$

where D_0, Θ and R_0 are video dependent constants determined through linear least squares estimation techniques.

Though this model works very well when there are no errors, any bit errors can decrease the quality of the received video. Unlike typical data networks, however, the video does *not* have to be received perfectly for it to by acceptably received by the user. This can be seen by observing a plot of the received video quality as a function of the bit error rate of the received video, as is shown in Fig. 2. For this plot, the videos were encoded to an acceptable quality, transmitted through a binary symmetric channel with varying bit error rates (BER) and then decoded. For low BER, there is almost no effect in the received SSIM. After the BER increases past a certain level, however, the video quality drops off significantly.

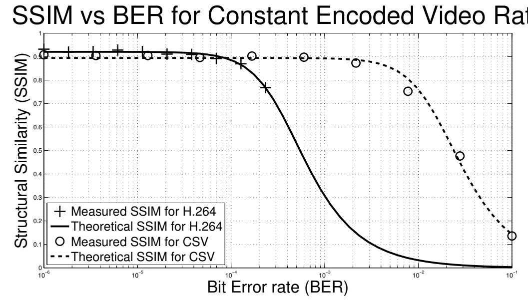

Fig. 2: SSIM vs BER for H.264 and CSV Encoders

Based on this observation, we have modeled the error performance as a low pass filter using

$$U(r_{ch}, r_v) = \frac{\alpha(r_v)}{\sqrt{1 + \tau^2 (BER(r_{ch}, r_v))^2}} \tag{4}$$

where r_{ch} is the channel coding rate (in $\frac{bits\ in}{bits\ out}$), r_v is the encoded video rate in $\mathrm{kbit/s}$, $U(r_{ch}, r_v)$ is the quality of the received video in SSIM as a function of r_{ch} and r_v. The encoder dependent constant τ is used to indicate where the quality begins to decrease.

A. SNR Model

Consider the energy budget per frame E_B as the energy available to the system during each frame period $t_f = \frac{1}{fps}$ where fps represents the number of frames per second of the video. We can then express the average energy available for frame transmission as

$$E_E(r_v) = E_{E,max} \cdot t_e(r_v), \tag{5}$$

where $E_{E,max}$ is the maximum energy available to the encoder during the frame period, and $t_e(r_v)$ is the processor load, i.e., the time fraction of a frame that the encoder needs to encode video at rate r_v. The transmitted energy per video frame E_T is defined as

$$E_T = (E_B - E_E(r_v)), \tag{6}$$

i.e., the total energy available reduced by the energy needed to encode the video.

For the encoders considered in this paper, the empirical models

$$t_e(r_v) = a\, r_v + b, \tag{7}$$

and

$$t_e(r_v) = c\, - \frac{T}{r_v + d}, \tag{8}$$

accurately model the processor load as a function of the encoded video rate, as shown in Fig. 3 where a, b, c, d and T are platform dependent positive constants determined through linear regression analysis of the encoder implementation.

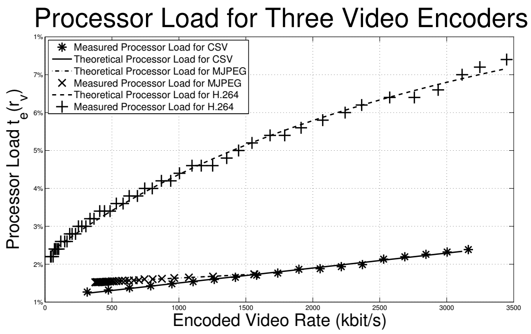

Fig. 3: Processor Load vs Encoded Video rate

To obtain these results, all three encoders are run at all available encoded video rates on the same platform. For this model, the platform is an Intel Core 2 Duo processor running Ubuntu 10.10. The time $t_f = \frac{1}{fps}$, defined as the inverse of the framerate of the video, is used as the maximum allowed encoding time, i.e. the mean encoding time per frame for a real time video *must* be less than t_f. The actual encoding time per frame, t_v is measured or estimated and compared to t_f. We can then find the value $t_e = \frac{t_v}{t_f}$ which represents the fraction of time used to encode each frame.

For example, if a 30 fps video of 3000 frames takes 15 seconds to encode (i.e., 200 fps encoding time) at some rate r_{v_0}, we find that each frame takes $t_v = \frac{1}{200}$ of a second to encode. In the video, each frame lasts $t_f = \frac{1}{30}$ of a second. This means that on average, for each frame, $t_e = \frac{30}{200} = 15\%$ of the frame time is needed to encode each frame. Taking the maximum encoder energy use per frame as 0.5 J (the value for the system used to test), then it will take on average $0.5 \times 15\% = 75mJ$ to encode that video at r_{v_0}.

We can then give the SNR model as

$$SNR(r_{ch}, r_v) = \frac{L \cdot r_{ch} \cdot d_{free} \cdot (E_B - E_E(r_v))}{N_0 \frac{r_v}{r_{ch} \cdot fps}}, \qquad (9)$$

where L is the path loss, N_0 is the noise power and d_{free} is the free distance of the channel code r_{ch}. As r_v increases, the energy needed to encode the video increases while the transmission energy per bit decreases, causing the SNR to decrease.

IV. Energy-Rate-Distortion Optimization

In Section III, we developed a set of equations to model the received video quality after transmission through a noisy channel with a finite energy budget per frame. In this section, we use this model to find the video encoder that results in the highest received video quality as a function of the energy budget E_B. We must first seek to find the optimal allocation of r_{ch} and r_v for a fixed energy budget. By repeating this for multiple values of E_E, we can then develop a model of the *optimal* received video quality as a function of E_B. The goal of this optimization is to determine i) which encoder requires the lowest energy to achieve a target video quality, and ii) what values of r_{ch} and r_v should be chosen to obtain that quality.

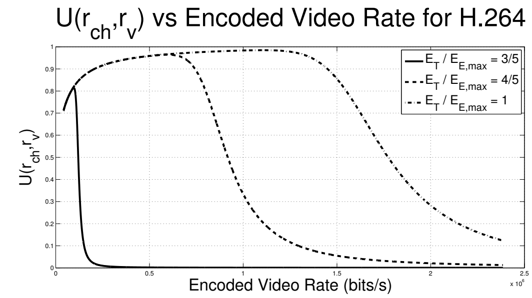

Fig. 4: Utility Function vs. Encoded Video Rate for H.264

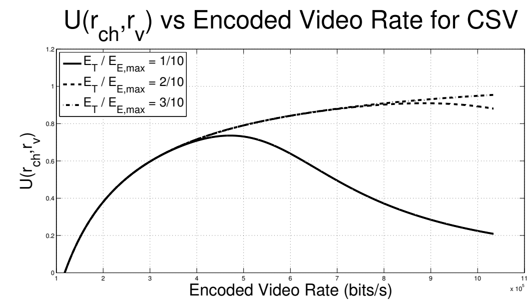

Fig. 5: Utility Function vs. Encoded Video Rate for CSV

We begin by holding E_B constant and finding the optimal allocation of r_{ch} and r_v. This problem can be modeled as the optimization problem

$$\begin{aligned} \underset{r_{ch}, r_v}{\text{maximize}} \quad & U(r_{ch}, r_v) \\ \text{subject to} \quad & E_B \geq E_E(r_v) + E_T, \end{aligned} \qquad (10)$$

where E_T is the energy available for transmission. Based on the analysis in Section III, we can formulate the problem as

$$\begin{aligned} \underset{r_{ch}, r_v}{\text{maximize}} \quad & \frac{D_0 - \frac{\Theta}{r_v - R_0}}{\sqrt{1 + \left(\tau \cdot Q\left(\sqrt{SNR(r_{ch}, r_v)}\right)\right)^2}} \\ \text{subject to} \quad & E_B \geq E_E(r_v) + E_T. \end{aligned} \qquad (11)$$

The solution to this problem results in the optimal channel and encoded video rates for a given energy budget and for a given encoder. Plots of the objective function for H.264, CSV and MJPEG are presented in Fig. 4, Fig. 5 and Fig. 6 showing the optimal rates for the given energy values. These are plotted for different values of $\frac{E_T}{E_{E,max}}$, which is the ratio of the total energy budget to the maximum energy per frame. For clarity, the function as plotted is actually $max r_{ch} U(r_{ch}, r_v,)$ vs r_v.

The maximum value of this objective function is the optimal video quality, and the values of r_v and r_{ch} that achieve that point are the optimal encoding rates. However, because the Q error function has no closed form solution, the problem must be solved numerically (though the Q error function can be approximated for some values of SNR, the resulting optimization problem is still a non-linear, non-convex discrete program without any obvious solution). To simplify the analysis, note that in all cases, the quality of the received video follows the

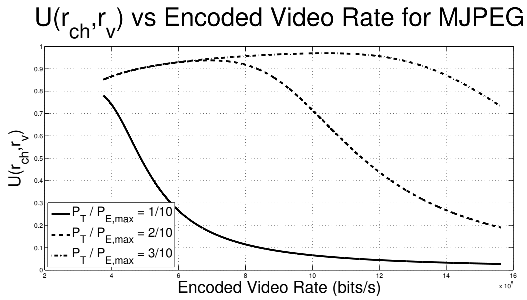

Fig. 6: Utility Function vs. Encoded Video Rate for MJPEG

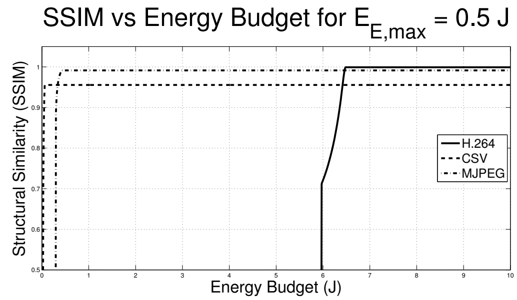

Fig. 7: SSIM vs Total Energy Budget for $E_{E,max} = 0.5J$

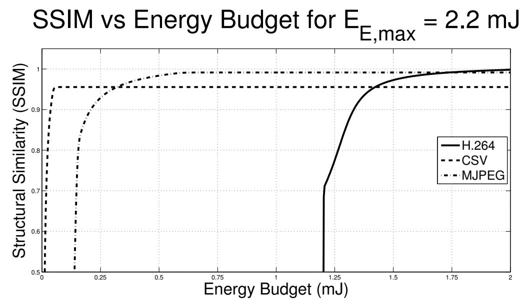

Fig. 8: SSIM vs Total Energy Budget for $E_{E,max} = 2.2mJ$

same pattern. The“filter-like” form of (4) results in a very sharp decrease in the quality after the rate increases beyond a certain point. It is safe to assume that this dramatic drop-off is due to the BER at that rate increasing beyond the cutoff point, driving the received video quality down.

Based on this, we assume that the optimal value would be obtained by a video encoder and channel encoder rate that are very close to that cutoff point. This leads to the much simpler optimization problem

$$\underset{r_{ch}}{\text{minimize}} \quad \left| Q\left(\sqrt{SNR(r_{ch}, r_v)}\right) - \frac{1}{2\pi\tau} \right|^2 \tag{12}$$

which states that the optimal point is the one that causes the BER to be as close as possible to that cutoff. This analysis reduces the original two dimensional optimization problem (10) to an optimization problem over a single dimension. For practical channel coders [20], the length of r_{ch} is generally less than 10. In comparison, the length of r_v can be 30 (MJPEG), 50 (H.264) or r_v can be continuous (CSV). by removing the search over r_v, we are reducing the majority of the search space of the problem, which will greatly reduce the complexity.

Simple tests show that in all cases, the values obtained from (12) are close to the optimal solution. The maximum error in SSIM was 0.31% for CSV, 1.12% for MJPEG and 0.94% for H.264.

V. Performance Evaluation

The objective of the optimization problem (10) or the simplified optimization problem (12) is two-fold. First, it allows comparing the performance of different video encoders. Second, once the optimal encoder is found, it finds the optimal values for the encoded video rate and the channel encoder rate that result in the optimal performance.

A. Analyzing Different Encoders

A major advantage of the analysis presented in the previous sections is that it is independent of any specific platform or encoder. To compare the performance of different encoders, we need to explore the design space varying the values for noise power, path loss, and $E_{E,max}$ of the system. To determine the optimal encoder for a specific platform, we need to empirically determine the energy-rate performance for the platform, and the rate distortion performance for the type of video being encoded.

Below we give example plots with different processors resulting in different values for $E_{E,max}$. First, we consider the case where the video is originating at a relatively high powered system with a maximum encoding energy cost of 0.5 J (i.e. the energy to encode a frame on a desktop or laptop computer), and is shown in Fig. 7. Even though the higher power system is able to encode video faster, the limiting factor in this system is the energy required to encode at *any* quality. Once encoding is possible, the SNR required to achieve a “good” quality received video is easily achieved. The second two situations are shown in Fig. 8 and Fig. 9. These two plots are generated with maximum encoding cost of 2.2 mJ (the energy to encode a frame on a small sensor node) and 0.167 mJ respectively. These values were chosen to represent smaller platforms that have significantly lower processor energy requirements.

In all of these simulations, there is a tradeoff between energy and received video quality. The CSV encoder results in a lower maximum received video quality, but can generally achieve that max quality at a much lower energy requirement than either MJPEG or H.264. For example say we want to achieve a 0.8 SSIM (“good” quality) with a maximum encoding cost of 2.2 mJ, as shown in Fig. 8. We can see that the CSV encoder crosses the 0.8 SSIM level very close to 0 mJ. The MJPEG encoder crosses around 0.15 mJ while the H.264 encoder crosses at 1.25 mJ. This means that *we can achieve the same quality for much lower energy cost using CSV.*
Clearly, the analysis is dependent on the noise power, path loss, encoder implementation and other application specific factors. For example, if the path loss is increased (or the noise

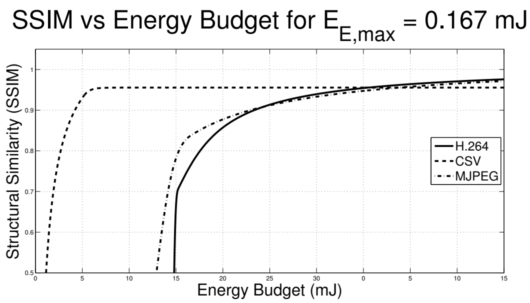

Fig. 9: SSIM vs Total Energy Budget for $E_{E,max} = 0.167mJ$

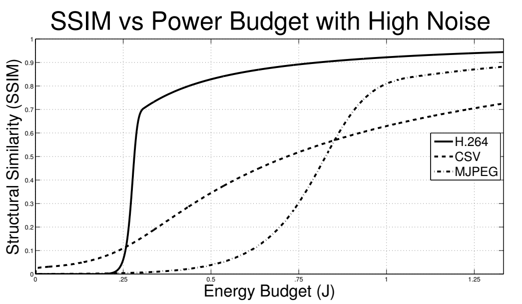

Fig. 10: SSIM vs Total Energy Budget for $E_{E,max} = 0.5J$ with $20dB$ Higher Noise Power

power is decreased) by 20 dB for the high power system shown in Fig. 7, the results are reversed, as shown in Fig. 10.

To get a more general comparison, Fig. 11 shows the achievable received video quality as the relative ratio of maximum encoder energy to total energy budget is increased. This allows us to view the optimal received video quality without the dependency on a specific platform. Because of its low encoding cost, CSV is able to achieve good video quality even when the cost of encoding the video increases. Since H.264 needs more energy to encode the video, it is unable to produce a video when the relative cost of encoding becomes too high.

VI. Conclusions and Future Work

We have introduced a rate-energy-distortion analysis of video transmission over wireless links with a limited energy budget for low-complexity sensing devices. Three video encoders; H.264, MJPEG and CSV are modeled and compared in realistic situations. It can be seen that CSV outperforms H.264 and MJPEG when the encoding energy is high compared to the video transmission energy. However, when energy is not as restricted, H.264 can achieve better performance because of its better rate distortion performance.

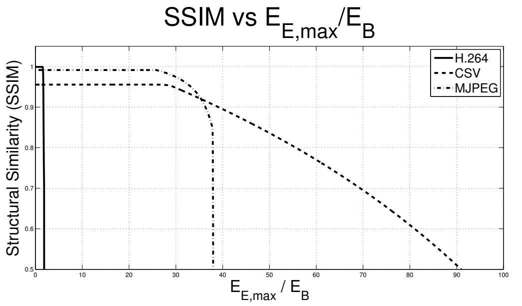

Fig. 11: SSIM vs $\frac{E_{E,max}}{E_T}$

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A Survey on Wireless Multimedia Sensor Networks," *Computer Networks (Elsevier)*, vol. 51, no. 4, pp. 921–960, Mar. 2007.

[2] A. Kansal and S. Nath and J. Liu and F. Zhao, "SENSE-WEB: An Infrastructure for Shared Sensing," *IEEE MultiMedia*, vol. 14, no. 4, pp. 8 – 13, October 2007.

[3] Y. Gu, Y. Tian, and E. Ekici, "Real-Time Multimedia Processing in Video Sensor Networks," *Signal Processing: Image Communication Journal (Elsevier)*, vol. 22, no. 3, pp. 237–251, March 2007.

[4] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 5, pp. 645 – 658, May 2005.

[5] Z. He, W. Cheng, and X. Chen, "Energy Minimization of Portable Video Communication Devices Based on Power-Rate-Distortion Optimization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 596 –608, May 2008.

[6] F. Bellard, http://www.ffmpeg.org.

[7] "Advanced Video Coding for Generic Audiovisual Services," ITU-T Recommendation H.264.

[8] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[9] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammar, and T.Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *IEEE Circuits and System Magazine*, vol. 4, no. 1, pp. 7–28, April 2004.

[10] D. Donoho, "Compressed Sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[11] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[12] E.J. Candes and J. Romberg and T. Tao, "Stable Signal Recovery from Incomplete and Inaccurate Measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, Aug. 2006.

[13] S. Pudlewski, T. Melodia, and A. Prasanna, "Compressed-Sensing-Enabled Video Streaming for Wireless Multimedia Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. preprint, 2011.

[14] S. Pudlewski, A. Prasanna, and T. Melodia, "Resilient Image Sensor Networks in Lossy Channels Using Compressed Sensing," in *Proc. Of IEEE Intl. Workshop on Information Quality and Quality of Service for Pervasive Computing (IQ2S)*, Mannheim, Germany, March 2010.

[15] "Digital Compression and Coding of Continuous-Tone Still Images - Requirements and Guidelines," ITU-T Recommendation T.81, 1992.

[16] N. Ahmed and T. Natarajan and K. R. Rao, "Discrete Cosine Transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, Jan 1974.

[17] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[18] K. Stuhlmuller, N. Farber, M.Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012–1032, Jun 2000.

[19] S. Pudlewski and T. Melodia, "A Distortion-minimizing Rate Controller for Wireless Multimedia Sensor Networks," *Computer Communications (Elsevier)*, vol. 33, no. 12, pp. 1380–1390, July 2010.

[20] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 389–400, Apr 1988.