Media Scaling for Power Optimization on Wireless Video Sensors

by

Rui Lu

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

by

____

August, 2007

____

Prof. Mark Claypool, Thesis Advisor

____

Prof. Robert Kinicki, Advisor

____

Prof. Emmanuel Agu, Reader

____

Prof. Michael Gennert, Head of Department

Abstract

Video-based sensor networks can be used to improve environment surveillance, health care and emergency response. Many sensor network scenarios require multiple high quality video streams that share limited wireless bandwidth. At the same time, the lifetime of wireless video sensors are constrained by the capacity of their batteries. Media scaling may extend battery life by reducing the video data rate while still maintaining visual quality, but comes at the expense of additional compression time. This thesis studies the effects of media scaling on video sensor energy consumption by: measuring the energy consumption on the different components of the video sensor; building a energy consumption model with several adjustable parameters to analyze the performance of a video sensor; exploring the trade-offs between the video quality and the energy consumption for a video sensor; and, finally, building a working video sensor to validate the accuracy of the model. The results show that the model is an accurate representation of the power usage of an actual video sensor. In addition, media scaling is often an effective way to reduce energy consumption in a video sensor.

Content

1) Introduction

With the fast developments of the last decade, sensor network technology [GW02] has been applied in scientific, industry and military applications. Intel motes [NKA05] are widely used in applications with low bandwidth such as measuring temperature, humidity and barometric pressure. These applications usually build a large scale wireless network to monitor the environment. A new trend in the wireless sensor network (WSN) is in context-aware systems, which change the system behavior according to the surrounding situation [BD]. These systems facilitate the interaction between the user and the event in progress, providing more complex data and identifying surroundings according to a stated preference. As the smart phone and iPod become a part of personal life, complex multimedia applications are integrated in the platforms with wireless network support.

Some surveillance and security applications [CH04] [JWW+04] [JWC04] use video sensors, sensor platforms equipped with cameras. Because of the complex video information, these applications need greater bandwidth and local computational ability. For example, Stargate video sensors can be deployed in hospitals to assist nurses caring for patients. Sensors enhanced by video data can provide a wealth of information. The nodes in the wireless network usually pass the visual raw data to the base station such as a PC, which analyzes the gathered video data to decide upon the events in the surroundings [YG03]. At the lowest end of the ability spectrum of video sensors is the tiny Cyclops [RBI+05] that can capture low resolution video. The Cyclops consists of an imager, a micro-controller (MCU) that controls the Cyclops sensor, a complex programmable logic device (CPLD), external SRAM and external Flash. CMUcams are cell-phone class cameras with on-board processing for motion detection. CMUcams are mainly designed for robots, thus they do not have a specific operating system [CMUcam]. Stargate video sensors have an X-Scale processor and internal memory with a Linux operating system [FCK+03]. Since Stargate includes interfaces for external memory, cameras and network cards, it supports more complicated tasks, such as video streaming, periodic sleep, and remote control. The Stargate can be configured to look for signs of life on the planets in the Personal Exploration Rover (PER), developed by CMU [PER].

Most multimedia data such as audio, image and video are more complex than ordinary data. Video data are usually large in storage size and are time sensitive. Even after compression the multimedia streams often can exceed available bandwidth on wireless networks. A video sensor does not have as high a computational processor capability as the typical PC to quickly compress the videos. Furthermore wireless transmission requires a large amount of energy [FCK+03]. Thus sending raw video data over a WSN is both a time-consuming and a power-consuming task for a tiny video sensor.

Surveillance sensors, installed outside, are powered by batteries for easy installation. The lifetime of the battery is limited. Even though improvements in processor technology make the sensors more cost-efficient and energy-efficient, the wireless transmission is still a large power cost for the sensor. In this condition, the data can be locally processed to reduce the need for transmission. Feng et al. [FCK+03] design a prioritizing buffer management algorithm to deal with intermittent network connectivity or disconnected operation to save power. They show the antenna of the wireless network draws significant power even when no data is in transmission. Some research studies attempt to improve the MAC protocol of the wireless sensor networks. Guo et al. [CLJ01] shows one way to create a separate ultra-low power "wake up" radio that constantly listens to the channel and wakes up the main receiver when a packet is about to arrive. When data flows over a wireless sensor network to the base station, it is important to find the shortest route in the wireless network. The paper [CV06] presents a routing algorithm that maximizes the lifetime of a sensor network in which all data packets are destined to a single collection node. Lifetime is maximized by optimizing the number of packets traversing each node.

As video provides considerable visual information, video sensors perform well in surveillance environment. Video compression reduces the quantity of data used to represent the video. The compressed video can effectively save storage memory and fit the available bandwidth of the network. Most video compression algorithms lose some information. This thesis explores media scaling on the video sensor. Media scaling is widely used to adjust the real-time video stream bit rate to network bandwidth constraints. Media scaling can be broadly categorized as follows:

Temporal scaling: Temporal scaling discards lower priority video frames before transmission. The resulting frames are still sequential, but the view is not as smooth as the original.

Quality scaling: Quality scaling changes quantization levels or drops chrominance and compression coefficients. The resulting frames are of a lower visual quality and may have fewer colors and details.

Media scaling can reduce network traffic without significantly impairing the video quality. Some research explores the trade-off between the bandwidth and video quality during scaling. [WCK06] combines temporal scaling and quality scaling to analyze the effect on videos over a range of network conditions. [WCK05] optimizes Forward Error Correction (FEC) to repair packet loss for streaming MPEG videos under a capacity constraint with quality scaling. These research results show that media scaling can reduce the data rate of the video streams, which indirectly decreases the transmission energy consumption.

Feng et al [FCK+03] configure a Stargate to be a low-power, high-quality video capturing platform that can serve as the basis of video-based sensor networks as well as other application areas such as virtual reality or robotics. One important feature of the Stargate is the battery monitoring capacity, available when the batteries connect to the main board. However the camera has to use the USB interface on the daughter board. [BLM+04] use the HP E3631A power supply to control the voltage and measure the current during different system states of the Stargate. This thesis follows this setup for power measurement.

The video sensor is powered by battery. Because of the power constraints of the video sensor, researchers have modified the MAC protocol [CLJ01], designed specific routing algorithms [CV06] and periodic sleep [WJD04] to improve the energy efficiency of the sensor. Gurun et al [GK06] built an energy consumption model of the Crossbow Stargate. Their model consists of a computation and a communication component, but does not consider the camera and is not suitable for evaluating the energy consumption of the Stargate video sensor. To the best of our knowledge, no one has thoroughly studied utilizing the compression technology for improving the energy efficiency on the Stragate.

The goal of this thesis is to understand the energy consumption for different components of a video sensor and build an energy consumption model to analyze the power effect of media scaling. The energy model analyzes media scaling on the improvement of the communication power. The goal is to understand the trade-off between energy consumption and video quality.

First, the test bed for our experiments is set up. The Stargate video sensor is powered by a battery. A multimeter is used to measure the power use of components of a video sensor, sending videos over a wireless infrastructure network.

The energy consumption of the Stargate video sensor is captured for different sensor states, such as video capture, video compression, network transmission and reading and writing to flash memory. The energy consumption is different when parts of the video sensor, such as camera and wireless card, are sleeping. The measurement data help to clarify the power distribution in the sensor components. To understand the details of the sensor camera, flash memory and the wireless card, several specific tools were developed for sensor power measurement. For the video compression, we upload three video samples with distinct motion features to the video sensor and measure power utilization, when the video samples are encoded with temporal scaling and quality scaling. Temporal scaling and quality scaling reduce the bit rate of the video stream and shorten the encoding time. Measuring the power of the compression process with media scaling facilitates comparing the trade-offs between video quality and sensor energy consumption.

Streaming multimedia typically use compression technology such as MPEG to sacrifice video quality for reduced video size. However, the video typically includes some redundant information. Information loss at a low level does not affect human visual quality. It is important to understand the video quality for media scaling. Peak Signal-to-Noise Ratio (PSNR) is the most common video quality metric, but does not accurately represent the perceptual quality of video [PSNR]. Video Quality Metric (VQM), developed by the Institute Telecommunication Sciences (ITS), provides an objective measurement of perceptual video quality [VQM]. It measures the perceptual effect of video impairments including unnatural motion, color distortion and blurring. Researchers prefer PSNR, because it is easy to understand and implement, but to thoroughly analyze the video quality in media scaling, both VQM and PSNR are measured in this thesis.

Using the measurements, this thesis develops an energy consumption model, which includes frame rate, video resolution, quantization level, frame distance, packet size and send rate, to analyze the power usage of the video sensor with media scaling. An actual system is built to validate the energy consumption result from the model. The results show the energy consumption model is a good fit to the actual system. A video system designer can use the model to evaluate energy consumption when designing a new video sensor system.

The system performance with the energy consumption model is analyzed, providing overall guidance for improving power use on a video sensor platform. The analysis determines that temporal scaling can reduce energy consumption by 65% and quality scaling can reduce energy consumption by 50%. Even though temporal scaling saves more power, it loses more video information than quality scaling. The MPEG encoder in the Stargate is the most time-consuming component and can support three frames per second maximum. The data processing speeds in the different components are usually inconsistent, but the buffer technology can help coordinate the pace of the components of the system. The effect of the buffer technology for the energy consumption of the system is also explored. The data buffer before sending in quality scaling can help reduce energy consumption by 75%.

Finally, the energy consumption model is used to discuss media scaling in two Stargate projects: the Personal Exploration Rover (PER) and the Environment Monitor (EM). These systems can save power with media scaling in specific scenarios and extended the lifetime of the system. Overall media scaling is an effective way to improve the energy consumption of the video sensor.

2) Background

This thesis involves both video compression and the video sensor itself. Thus this chapter introduces several video compression technologies and describes MPEG (Moving Picture Experts Group), one of the most popular video compression standards. Compression technology inherently loses some video information, so this chapter introduces PSNR and VQM, which are used to quantify the video quality. There are mainly two sensor classes: Mote and PDA class sensors. The Yale XYZ and Cyclops belong to PDA class sensor. This chapter introduces their special feature and powerful functions. Since this thesis measures the energy consumption a video sensor, some basic theories of circuits are introduced and several power measurement methods with different features are introduced.

2.1 Video Compression

Video compression reduces the quantity of data used to represent the video. The compressed video can effectively save storage memory and adapt to the available bandwidth. This thesis explores the trade off between the energy consumption and the video quality.

2.1.1 Basic Concept

During compression, each event in the video is assigned a code. Commonly occurring events are assigned a few code bits and rare events are assigned more code bits. This thesis implements the MPEG video compression in the video sensor. Discrete cosine transform (DCT) is the basis of MPEG. The definition of the DCT is now covered.

DCT: DCT is often used in signal and image processing, especially for lossy data compression. It helps separate the image into parts of differing importance (according to the visual quality of the video). It transfers an image from the spatial domain to the frequency domain [DCT]. The input image is A and the coefficient of compressed output image is B.

$$B(k_1,k_2)=\sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1}4\cdot A(i,j)\cdot\cos[\frac{\pi\cdot k_1}{2\cdot N_1}\cdot(2i+1)]\cdot\cos[\frac{\pi\cdot k_2}{2\cdot N_2}\cdot(2j+1)] \qquad (2.1.1)$$

The input image is N_2 pixels wide by N_1 pixels high; A(i,j) is the intensity of the pixel in row i and column j; B(k_1, k_2) is the DCT coefficient in row k_1 and column k_2 of the DCT matrix. All DCT multiplications are with real numbers. This lowers the number of required multiplications, as compared to the discrete Fourier transform. The DCT input is an 8 by 8 array of integers. This array contains each pixel's gray scale level; 8 bit pixels have levels from 0 to 255. The output array of DCT coefficients contains integers; these can range from -1024 to 1023. For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT. The lower right values represent higher frequencies, and are often small - small enough to be neglected with little visible distortion [DCT].

MPEG video compression discards some high spatial frequency information - the information which is least noticeable to the eye. The first step in this process is to convert a static picture into the frequency domain. DCT performs this transformation.

2.1.2 MPEG

***M**oving **P**icture **E**xperts **G**roup* [MPEG] is a working group of the ISO. The term also refers to the family of digital video compression standards and file formats developed by the group. The MPEG algorithm compresses the video to a smaller file that can be easily transmitted and then decompressed. It does not always compress each entire frame, but only capture changes from one frame to another. The MPEG frame includes three types of frames:
Intrapictures (I-frames): These frames are encoded only with respect to themselves.
Predictive pictures (P-frames): These are frames encoded using motion-compensated prediction from a past I-frame or P-frame.
Bidirectional pictures (B-frames): These are frames encoded using motion-compensated predictions from a past and/or future I-frame or P-frame.

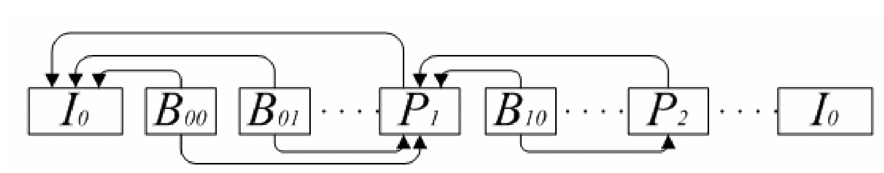


Figure 2.1 The relationship between three types of MPEG frames

The compression technology of MPEG is DCT compression. Even though it loses a little video information, the loss is not easily identified by eye. The main MPEG compression standards include:

- MPEG-1: Initial video and audio compression standard [MPEG].
- MPEG-2: Transport, video and audio standards for broadcast-quality television.
- MPEG-4: Expands MPEG-1 to support video/audio "objects", 3D content, low bit rate encoding and support for Digital Rights Management [MPEG].

In the thesis, the MPEG video compression is implemented by FFmpeg, [FFMPEG] which is a complete system to record, convert and stream audio and video. FFmpeg is developed under Linux, but it can be compiled under most operating systems, including Windows and Arm Linux. It also supports capturing and encoding in real time from camera or TV card. It includes libavcodec, the leading audio/video codec library. libavcodec is a library containing all the FFmpeg audio/video encoders and decoders. Most codecs were developed from scratch to ensure best performances and high code reusability. MPEG1 and MPEG4 are both supported in FFmpeg.

2.1.3 Video Quality

Video quality is a characteristic of any video, based on the subjective opinions about how a video looks. Usually it is measured by comparing the original video and the processed video. MPEG video compression is lossy. PSNR and VQM are two significant video quality measurement methods.

PSNR is the most widely used objective video quality metric [PSNR]. PSNR is most easily defined via the mean squared error (MSE) which for two m×n monochrome images I and K where one of the images is considered a noisy approximation of the other is defined as(I(i,j) and K(i,j) is the pixel value):

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}||I(i,j) - K(i,j)||^2 \qquad (2.1.3.1)$$

The PSNR is defined as [PSNR]:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \qquad (2.1.3.2)$$

MAX is the maximum pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. More generally, when samples are represented using linear PCM with B bits per sample, maximum possible value of MAX is 2*B*-1.

For color images with three RGB values per pixel, the definition of PSNR is the same except the MSE is the sum over all squared value differences divided by image size and by three.

The PSNR value usually varies between 20 and 50. A large value means the compressed video is as good as the original video. While a small value means the compressed video is poor. However, PSNR values do not perfectly correlate with a perceived visual quality due to the non-linear behavior of the human visual system. Human visual system requires a detailed knowledge of color, its representations, and the way the eye perceives it. VQM, developed by Institute for Telecommunication Sciences (ITS), provides a more accurate and complex measurement for perceived video quality [VQM].

VQM takes the original video and the processed video as input. [W06] describes VQM as follows:

- Calibration

  This step calibrates the sampled video in preparation for feature extraction. It estimates and corrects the spatial and temporal shift as well as the contrast and brightness offset of the processed video sequence with respect to the original video sequence.
- Quality Features Extraction

  This step extracts a set of quality features that characterizes perceptual changes in the spatial, temporal, and chrominance properties from spatial-temporal sub-regions of video streams using a mathematical function.
- Quality Parameters Calculation

  This step computes a set of quality parameters that describe perceptual changes in video quality by comparing features extracted from the processed video with those extracted from the original video.
- VQM Calculation

  VQM is computed using a linear combination of parameters calculated from previous steps.

VQM [W06] can be computed using various models based on certain optimization criteria. These models include (1) Television (2) Videoconferencing (3) General (4) Developer (5) PSNR. The general model is used in my experiment.

VQM produces a distortion value between 0 and 1. A value of 0 means the quality of the processed video is as good as the original video and a value of 1 means the processed video has really poor quality compared to the original video.

2.2 IEEE 802.11

In this thesis the video sensor uses the IEEE 802.11b to send the videos. Thus we cover the definition of IEEE 802.11. IEEE 802.11, commonly known by the brand Wi-Fi, denotes a set of Wireless LAN standards developed by working group 11 of the IEEE LAN/MAN Standards Committee (IEEE 802). MAC protocols are broadly of two types: random access and time division multiple access (TDMA). Some MAC protocols are centralized, with the base station or group leader used to control the access of the node. Others use distributed methods [WIRELESS].

802.11b and 802.11g standards use the 2.4 GHz band, operating (in the United States) under Part 15 of the FCC Rules and Regulations. Because of this choice of frequency band, 802.11b and 802.11g equipment could suffer interference from microwave ovens, cordless telephones or Bluetooth devices. The 802.11a standard uses a different 5 GHz band, which is less populated by comparison. 802.11a devices are never affected by products operating on the 2.4 GHz band.

Protocol	Frequency	Max Data Rate	Range (Indoor)
802.11a	5.15-5.25/5.25-5.35/5.725-5.875 GHz	54 Mbit/s	~30 meters
802.11b	2.4-2.5 GHz	11 Mbit/s	~35 meters
802.11g	2.4-2.5 GHz	54 Mbit/s	~35 meters
802.11n	2.4 GHz and/or 5 GHz	248 Mbit/s	~70 meters

The 802.11b amendment to the original standard was ratified in 1999. 802.11b has a maximum raw data rate of 11 Mbit/s and uses the same CSMA/CA media access method defined in the original standard. Due to the CSMA/CA protocol overhead, in practice the maximum 802.11b throughput that an application can achieve is about 5.9 Mbit/s using TCP and 7.1 Mbit/s using UDP[Throughput].

During 802.11 packet assembly, payload data from the IP layer, or the data that is being communicated, is encapsulated with MAC (media access controller) data and another four-byte segment of data that functions as a check sum and is also referred to as CRC or FCS. All of this data is assembled into an MPDU (MAC Packet Data Unit). When the packet is transmitted, the PHY layer appends a synchronization header. A complete 802.11 packet is illustrated in Figure 2.2.

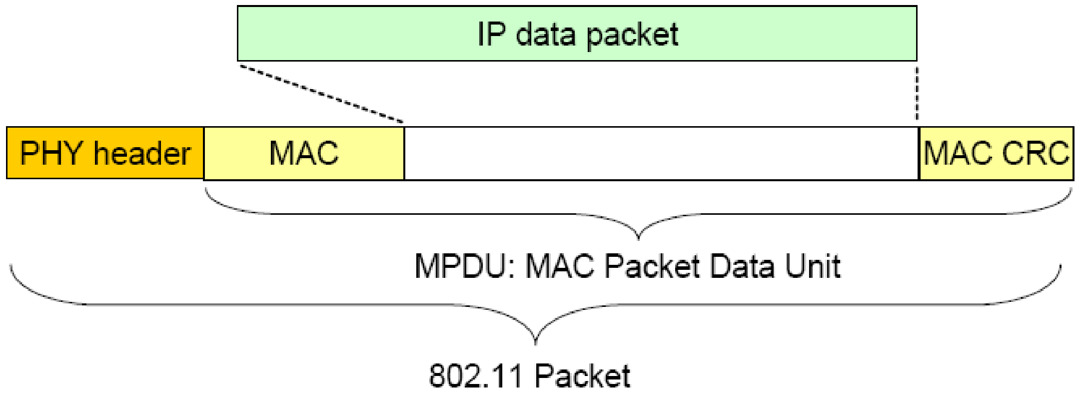


Figure 2.2 802.11 packet

2.3 Sensor Platform

The relentless pace of technological growth has led to the emergence of a variety of sensors and networked sensor platforms. Networked sensors span the spectrum of cost, form-factor, resolution, and functionality [KGS+05]. It is important to understand the features of the different platforms. Motes are tiny and suitable for large scaling network. Cyclops and Stargate are typical video sensors.

2.3.1 Motes

Subminiature motes are available today as prototypes and can be configured with a variety of sensors in large scale distributed sensor networks. Motes use the IEEE 802.15.4 compliant MAC protocol. The data rate of IEEE 802.15.4 can reach up 250kbit/s. Different research organizations have developed several types of smart motes with their own purpose, typically consisting of a microcontroller, a communication unit and onboard sensors or integrable sensor board modules [W04]. The flash memory of the motes is limited, thus the operating system needs to be compact and to perform customized tasks. TinyOS is one such operating system, developed at The University of California at Berkeley, and is installed in most of the motes [TinyOS].

While mote [W04] sensors are very small, their power, communication range and memory are limited, whereas reliability and accuracy vary. Motes are equipped with several sensors and wireless communication modules; they are an excellent choice for distributed sensor networks. In addition, motes' computational capability can enable more sophisticated sensor networks. Comparing to a single expensive high fidelity sensor, a network of mote sensors using appropriate algorithms is superior in providing local information as well as global knowledge, and is much more robust against failure [W04].

2.3.2 Yale XYZ

The XYZ platform [LS05] takes a forward step in the direction of instantiating a new sensor node platform designed to support mobility experiments in sensor networks. XYZ is built around an ML67Q500x series ARM/Thumb microcontroller from OKI Semiconductor and a CC2420 radio with a 250 kbps raw data rate from Chipcon [LS05]. The choice of the OKI microcontroller provides a wealth of peripherals and flexible modes of operation. The Chipcon radio uses an IEEE 802.15.4 compliant MAC protocol. Its new features include support for two different CPU sleep modes and a long-term ultra low power sleep mode for the entire node. This allows the XYZ and its peripheral boards to transition into deep sleep for extended time intervals. To support mobility hardware control and computation, XYZ [LS05] supports a wide dynamic power options. In low power configuration the node resembles existing small low power nodes. When needed, the node can scale up its resources to perform more powerful computations. It dose not support video functions.

2.3.3 Cyclops

Here one video sensor is introduced. In [RBI+05] Cyclops consists of an imager, a micro-controller (MCU), a complex programmable logic device (CPLD), an external SRAM and an external Flash. The MCU controls the Cyclops sensor. It can set the parameters of the imager, instructs the imager to capture a frame and run local computation on the image to produce an inference. The CPLD provides the high speed clock, synchronization and memory control that is required for image capture. The combination of the MCU and the CPLD provides the low power benefits of a typical MCU with on-demand access to high speed clocking through a CPLD. Furthermore, the CPLD can perform a limited amount of image processing such as background subtraction or frame differentiation at capture time [RBI+05]. Cyclops uses external SRAM to increase the limited amount of internal MCU memory and provide the necessary memory for image storage and manipulation. Each module in Cyclops has several power states. Typically the lowest energy consumption sleep states have the highest wake-up cost. Consequently, the application should choose the sleep level based on its amortized cost [RBI+05].

2.3.4 Stargate

The Stargate video sensor system is comprised of four parts: a Stargate sensor [Stargate] (400 MHz, Intel PXZ255 Processor; Embedded Linux BSP Package; 64 MB of SDRAM; 32MB of Intel StrataFlash; Ethernet, Serial, JTAG, USB Connectors via 51-pin Daughter Card interface; Li-lon battery Option, PCMCIA and Compact Flash), a Logitech 4000 USB-based video camera and an AmbiCom wireless LAN IEEE 802.11b CompactFlash Card. A picture of the Stargate video sensor is shown in Figure 4.2. It's powered by 4 2500mAh AA NI-MH rechargeable batteries in power measurement experiments.

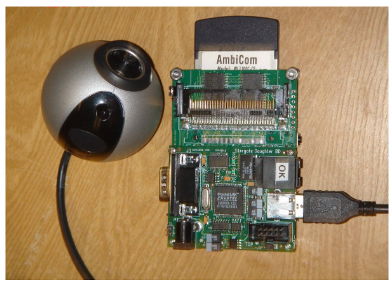


Figure 4.2 Stargate Video Sensor

The kernel of the Stargate is ARM Linux 2.4.19 and only occupies 10 Mbytes memory. It has originally supported several languages such as C/C++, Java and Perl. The C code of Stargate has to be compiled and built in a typical PC, which is patched with a cross compile environment. After the compilation, the executable files can be sent by HyperTerminal over the serial cable or by the SCP command over wireless network. The transmission speed of the serial cable is slow. For relatively large files, it is better to use the network rather than the serial cable. Because of the powerful computation ability and the plentiful external interfaces of the Stargate, this thesis chose it to build the Stargate video sensor.

2.4 Power Measurement

2.4.1 Concepts of Electric Circuit

This thesis discusses power issues on wireless sensors, so the following electric units are often mentioned:

P is the power (watt or W)

I is the current (ampere or A)

V is the potential difference (volt or V)

R is the resistance (ohm or Ω)

E is the energy (joule or J)

The amount of electric **current** (I, measured in amperes) [EC] through some surface, e.g., a section through a copper conductor, is defined as the amount of electric charge (measured in coulombs) flowing through that surface over time. If Q is the amount of charge that passed through the surface in the time T, then the average current I is:

$$I = \frac{Q}{T} \quad (2.4.1.1)$$

Voltage (V, measured in volts) is the difference of electrical potential between two points of an electrical or electronic circuit. Voltage drop is the reduction in voltage in an electrical circuit between the source and load. Voltage drop may be neglected when the impedance of the interconnecting conductors is small relative to the other components of the circuit, such as the resistance inside of the battery. Figure 2.3 illustrates the voltage drop definition. This circuit has one power source (battery) and two resistances (R1 and R2). The current flow out from the positive end of the battery and flows in from the negative end of the battery. One wire connects the circuit to the ground, so $V_c = V_{in} = 0$, The voltage of this point is usually assumed to be zero volts. In the experiment the wires are short, thus the resistance on the wires are assumed to be neglectable, which then yields the relationship: $V_a = V_{out} = V_{battery}$. When the current flows over the resistance, the voltage drop on R1 is $V_1 = V_a - V_b$ and the voltage drop on R2 is $V_2 = V_b - V_c$. Overall, the voltage from the battery costs by the R1 and R2, thus $V_{battery} = V_1 + V_2$.

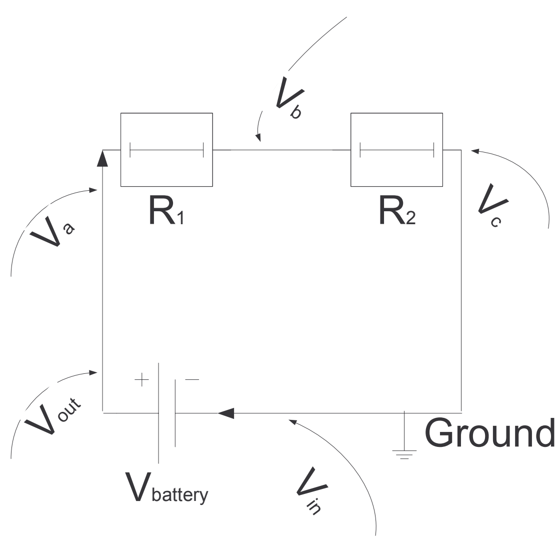


Figure 2.3 Voltage drop

Resistance (R, measured in ohms) is a measure of the degree to which an object opposes an electric current through it.

Ohm's law [EC] predicts the current in an (ideal) resistor (or other ohmic device) to be applied voltage divided by resistance:

$$I = \frac{V}{R} \qquad (2.4.1.2)$$

Electric **power** (P, measured in watts) is defined as the rate at which electrical energy is transferred. In direct current resistive circuits, instantaneous electrical power is calculated using Joule's Law, which is named after the British physicist James Joule, who first showed that electrical and mechanical energy were interchangeable.

$$P = IV \qquad (2.4.1.3)$$

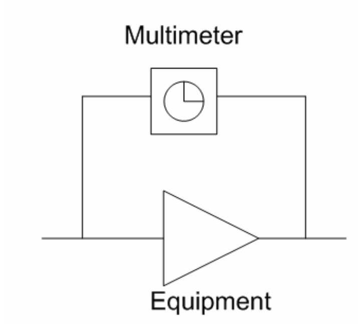
Energy (E, measured in Joules) is the product of the power and the time.

$$E = PT \qquad (2.4.1.4)$$

2.4.2 Measurement Methods

A large number of embedded computing applications are power critical. This has led to a significant research effort in power estimation and low power design. But currently few power measurement tools are available.

The multimeter is a common external power measurement tool. It is necessary to understand the circuit before measurement with a multimeter. When a multimeter measures the voltage of an electrical device, it should be in a parallel connection circuit as in Figure 2.4. When a multimeter measures the current of the device, the multimeter and the device should be in a series connection circuit as in Figure 2.5. Many multimeters have self-protection mechanism for avoiding the danger caused by connection errors. Our experiments use a multimeter, as it is suitable for black box measurement, regardless of the infrastructure of the equipment.



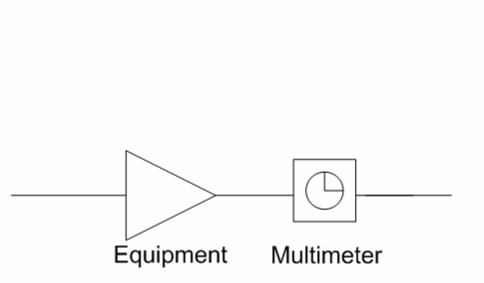


Figure 2.4 Measure the voltage Figure 2.5 Measure the current

Currently, some internal power measurement tools are available for the lower level of the design-at the circuit level. The DS2438 Smart Battery Monitor [P06] provides some functions to monitor real-time battery performance. It integrates the current accumulator which keeps a running total of the current going into and out of the battery. Because each DS2438 contains a unique silicon serial number, multiple DS2438s can exist on the same 1-Wire bus. This allows multiple battery packs to be charged or used in the system simultaneously. This kind of power measurement chip is designed for specific equipment, not working on others. In the Stargate sensor, the DS2438 can not accurately measure the power if the daughter board is installed. That is why we choose the multimeter.

Instead of the power measurement tool in the hardware, some researchers use power analysis software [GK06]. The basic idea is to measure the instruction level energy consumption then develop a model for the processor. The software can be employed to evaluate the power cost of embedded software. In addition, it can also be used to search the design space in software power optimization. For example, the power reduction reaches up to 40% on the Intel 486DX2, obtained by rewriting code using the information provided by the software with the instruction level power model. Before the development of the measurement software, instruction level power had to be measured. This kind of measurement tool is specific to their systems.

3) Related Research

This chapter introduces related research work on Stargate sensors, including performance improvements, energy consumption models and system simulations. Then, we describe several power improvement methods to the MAC protocol and video compression. Finally, we describe research on media scaling for network congestion.

3.1 Stargate Sensor

Stargate, developed by the Crossbow Technology, Inc., is a powerful single board computer with enhanced communications and sensor signal processing capabilities [Stargate]. It is designed mainly for wireless sensor network and robotics. The Stargate sensor includes a USB camera, an Xscale processor and a wireless card. The camera can capture videos from the surrounding environment, the processor can parse and analyze the data and the wireless card can transmit the data to the base station.

Feng et al. [FCK+03], configure the Stargate to be a low-power and high quality video sensor, which can be used to build up large scale wireless sensor networks and intelligent robotics. They propose a buffer management system with priority to deal with intermittent network connectivity or disconnected operation to save power. They designed a bit-mapping algorithm for the efficient querying and retrieval of video data. They find that the Stargate can achieve about 3 frames per second across the USB bus. They measure the energy consumption of the system with an HP-3458A digital multimeter connected to a PC. Our configuration is similar to the one used in this paper. But this research measures the energy consumption of different individual components and analyzes their power performance.

Sitbon et al. [SFN+07] have develop a multimodal, multimedia sensor networking toolkit, named SenseTK. It is intended to bridge the diversity of low-level hardware for sensor networking applications. It is convenient for users that need to build applications for specific purposes. SenseTK facilitates the software development on Stargate, enabling the fast development and the performance improvement of the video sensor.

Gurun and Krintz [GK06] build an energy consumption model of the Crossbow Stargate. Their model consists of a computation and a communication component. They use an Agilent 54621A oscilloscope and a light weight device driver to collect data from CPU hardware performance monitors. Their computation model employs six parameters: cycles per instruction (CPI), instruction cache misses, instructions not delivered, data stalls, instruction TLB (Translation Lookaside Buffer) misses and data TLB misses. They show a CPI based one-input model is effective and performs best for estimating full-system computational energy. My model also includes a video camera component. The camera costs a different amount of energy consumption depending upon the resolution and the capture rate. It is difficult to measure the camera power with this CPI power model. Our model focuses on macroscopically evaluating the energy consumption of the system.

Wen et al build a Stargate full-system simulator, named SimGate [WGC+], that simulates the components of Stargate including the processor, memory hierarchy, communications (serial and radio), and peripherals. SimGate supports Linux and can run any program on it without modification. On average, it is 20 times slower than a real device when using functional emulation, but can be a useful tool to understand and improve the system infrastructure of the Stargate.

3.2 Power Improvement

Battery life has improved, but not as rapidly as has computer performance. Battery life largely constrains the development of various embedded systems applications. Voltage scaling and specialized circuit techniques have been the main strategies for low-power designs, and these will continue to be important areas in the future.

Wireless sensors differ in power performance characteristics. The most appropriate sensor depends on the type of the tasks. Kulkarni et al. [KGS+05] design a multi-tier camera sensor network and map each task to the least power-hungry tier with sufficient resources. This system has an order of magnitude reduction in energy usage while providing comparable surveillance accuracy to a single sensor. In the Stargate video sensor system, since tasks only need the relevant resources active, idle components can be shut down to extend the battery life.

Feng et al. [FCK+03] measure the energy consumption of various system states of the Stargate video sensor without media scaling. They design a prioritizing buffer management algorithm to deal with intermittent network connectivity or disconnected operation to save power. We intend to use media scaling to alleviate the workload of the video stream and gain a general evaluation on the relation between media scaling and the energy consumption.

Coleri Ergen et al. [CV06] show that a great deal of power is consumed by wireless networks. Part of power optimization research has been done on the network's medium access control (MAC) protocol [SP06], which determines how the radios are operated. It has a decisive influence on battery lifetime. The experiment shows the radio draws a lot of energy, but an appropriate MAC protocol can help improve the energy consumption of the sensor.

Several methods of standard random access are designed to reduce a node's radio activities such as idle channel listening, overhearing packets not intended for itself, and packet transmissions. [CLJ01] shows one way to create a separate ultra-low power "wake up" radio that constantly listens to the channel and wakes up the main receiver when a packet is about to arrive. Some researchers use periodic listen and sleep modes to decrease idle listening [WJD04]. We also measure the energy consumption during the sleep state, which is an effective way to improve the power performance of the video sensor.

3.3 Media Scaling

Temporal scaling and quality scaling are widely-used technologies for adjusting to available bandwidth constraint. They can reduce the bit rate of the video stream while maintaining acceptable video quality.

When a compressed video is sent over a network with a capacity constraint, two important factors affect the video quality: video compression and packet loss [WCK06]. As little as 3% MPEG packet losses can cause 30% of the frames to be undecodable [WCK06]. Wu et al. [WCK05] optimize Forward Error Correction (FEC) to repair packet loss for streaming MPEG videos under a capacity constraint with quality scaling. Adjusting FEC with quality scaling significantly improves the performance of congested networks. This suggests quality scaling can mitigate the load of the video streams to promote power efficiency.

Wu et al. [WCR06] combine temporal scaling and quality scaling to analyze the effect over a range of network conditions. Their analytic experiments show that quality scaling typically performs better than temporal scaling. When the network capacity is low and the packet loss rate is high, the combination of temporal scaling and quality scaling performs better than quality scaling alone. They show that the effect of quality scaling and temporal scaling is related to the motion features of the video. That motivates us to test the power effects of media scaling on several video samples with different motion characteristics.

Krasic et al. [KWF03] evaluate the efficacy of priority drop in best effort computing and networking environments. They show that the video source can stream over a wide range of network bandwidths and maintain real-time performance. Instead of the network transmission being the bottleneck, battery lifetime becomes the most important factor in wireless sensor networks. They show that media scaling can significantly improve the quality of the multimedia stream. The video quality directly affects the energy consumption of sensors. This thesis intends to use temporal scaling and quality scaling to reduce the energy consumption for video sensor networks.

4) Methodology

This chapter introduces the configuration of the test bed and the features of the experiment tools. Additionally the system configuration for measuring the energy consumption of the individual components is described. After the power measurement of media scaling, the video quality is measured in the video master. Finally we build an energy consumption model to explore the trade-off between the energy consumption and the video quality.

4.1 Setup Test Bed

The wireless video sensor network testbed for this investigation (Figure 4.1) has a desktop computer acting as the video master to remotely control the Stargae video sensor through a serial cable. The video stream is captured by the video camera, temporarily buffered in the memory of the sensor, transmitted over an 802.11b ad-hoc network, and then stored in the video master. 4 NI-MH rechargeable batteries power the video sensor. The data rate of the wireless card in the Stargate is set to 11Mbps. A FLUKE 177 true RMS multimeter is used to measure the overall average current of the sensor platform under different system states and analyze the distribution of power of individual sensor components. The distance between the video master and the video sensor is approximately 1 meter to keep the signal strength excellent throughout all the power measurements. The video master is a typical PC with Intel(R) Pcntium(R) 4 CPU 2.8GHz, SMC EZ Connect Wireless USB Adapter and Windows XP operating system.

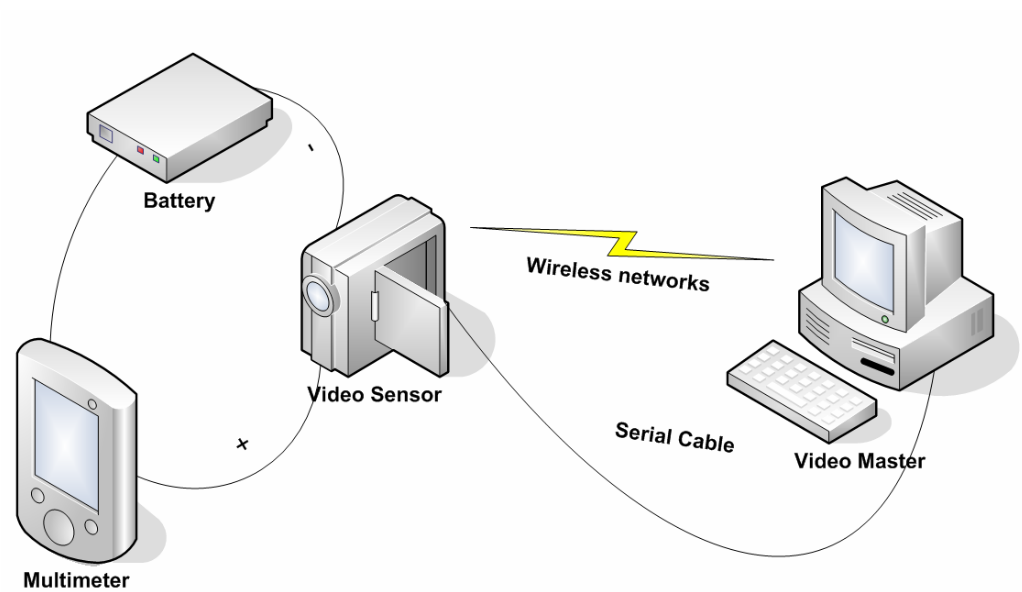


Figure 4.1. Video Sensor Platform

The circuit graph is shown in figure 4.2. At first, we measure the export voltage of the battery (V_{source}), the voltage drop of the sensor (V_{sensor}) and the current (I) of the system. The voltage drop of the battery is $V_{battery} = V_{source} - V_{sensor}$. The resistance of the battery can be calculated by $R_{battery} = \frac{V_{battery}}{I}$.

In each experiment, the export voltage of the battery is measured and recorded before the experiment as V_{start} and after the experiment as V_{end}. The average voltage ($V_{source} = \frac{V_{start} + V_{end}}{2}$) is used for later power calculations. The FLUKE multimeter samples the current once per second, reporting the average $I_{average}$ over a 50 second interval. The basic electric theory in Chapter 2.4 is used to analyze the circuit in Figure 4.2, the average power used by the sensor is the export power of the battery minus the power of the battery resistance.

$$P_{avg} = V_{source} I_{avg} - I_{avg}^2 R_{battery} . \tag{4.1.1}$$

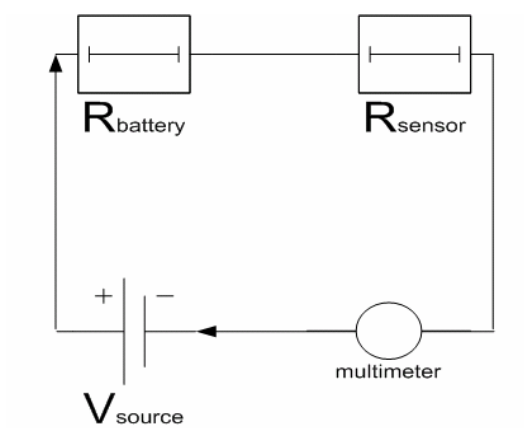


Figure 4.2 the circuit graph

4.2 Build Components

The goal of the thesis is to build an energy consumption model to analyze the video sensor and use the model to explore the power improvement of media scaling. Before the establishment of a model, it is necessary to determine the energy consumption of individual Stargate components.

The multimeter probe is a little large, compared to the circuit on the chip. So it can not measure the individual power in such a small video sensor. We can only measure the full-system energy consumption under different conditions. To isolate the energy consumption of individual components, separate software had to be devised.

1) Video capture: The Logitech 4000 USB-based video camera captures video with varying motion characteristics at a fixed frame rate between 1 and 30 fps. *videotime*, modified from Stargate's sample application, can capture different resolution videos at a variety of capture rates. In a typical video sensor, the video stream would go directly from the camera to wireless card. So in the video capture experiment, the videos from the camera are discarded and not saved to the flash memory. By measuring the current and voltage during the video capture process, the power utilized for video capture ($P_{capture}$) can be determined.

2) Compression and scaling: FFmpeg supports setting the frame rate, the quantization level and the resolution. To isolate the power cost of video capture, the raw video is uploaded in the flash memory before the experiment. For temporal scaling, the raw videos are selectively discarded by a script. In the next step, the processed video are encoded by FFmpeg at a given quantization level. After measuring the current and voltage during the video compression, the encoding power can be measured ($P_{flash-encode}$), that includes the cost of reading raw videos and writing compressed videos into the flash memory. Due to the flash memory limitation of the Stargate, the length of the video is only 2 seconds. The compression time $T_{flash-encode}$ for 2 seconds of video is reported using the script.

   This thesis uses Pre-Encoding Temporal scaling (PETS) to selectively discard the uncompressed pictures before encoding. The set of distances used in temporal scaling investigation (i.e. the distance between sequential images after PETS) are: 0, 2, 4, 8, and 16. For example, for frame distance 2, the sequence of the frames, "GGGGGGGGG" encoded and transmitted would be "G- -G - -G - -". Table 1 provides the complete set of scaling pattern for the frame distance used.

Frame Distance	Scaling Pattern
0	GGGGGGGGGGGGGGGGG
2	G- - G- - G- - G- -G- - G- -
4	G- - - -G- - - - G- - - - G- -
8	G- - - - - - - -G- - - - - - - -
16	G- - - - - - - - - - - - - - - -G

Table 1. Temporal scaling

Quality scaling uses quantization in the coding to save bits. The frame distance is 0 in Quality scaling. The quantization level can be selected from 1 to 31. The set of quantization levels used in the study is 2, 4, 8, 16, 31.

FFmpeg compresses the raw YUV video to MPEG4 video with the Group of Pictures (GOP) pattern, 'IBBPBBPBBPBB'. Three types of videos are tested in the study: low motion (person is reporting news), medium motion (the person demonstrating sign language), and high motion (panning of a moving car). They are got from Wu's paper [WCK06].

3) Transmission and reception: The video master remotely controls the video sensor over the serial cable and the video sensor sends video streams over the wireless network. The wireless network is set to 802.11b, 11Mbytes/s and RTS/CTS disabled. To isolate the reading video cost from the flash memory, UDPsender in the video sensor sends random data to the video master and UDPreceiver in the video master receives them. The packet size (bytes) and send rate (packets per second) are set as command line arguments. After measuring the current and voltage during sending the data, one can determine the power used for sending: P_{send} .

4) File operation: The Stargate has 32 MB flash memory and 64 MB SDRAM. To eliminate the flash memory cost during the compression process, we use two simple programs to read and write the different size files in the sensor. The power of reading and writing (P_{read} and P_{write}) and the operation time (T_{read} and T_{write}) are recorded.

4.3 Measure Video Quality

Since most video compression technology is lossy, it is necessary to consider the video quality factor during compression. PSNR is the most popular video quality metric. FFmpeg provides an embedded real time PSNR measurement during the compression process. After setting parameters for FFmpeg to measure PSNR during the compression, FFmpeg will report the overall PSNR after the completion of compression.

VQM, developed by ITS, provides an objective measurement for perceived video quality [VQM]. VQM takes the raw YUV video and the processed YUV video to compute the VQM value. So it is necessary to convert the MPEG video back to a YUV file for VQM computation.

In temporal scaling, we have to reconstruct the original YUV video from the transformed MPEG file. FFmpeg does not support the direct transfer of the MPEG file to the multiple YUV frames, but supports decoding to separate JPEG frames. After inserting their neighbors to the discarded frames' positions, the JPEG files can be converted by the JPEG tool, jpegtopnm and ppmtoy4m, to one YUV file. The whole process is shown in the figure 4.4. However, in quality scaling, no frames need to be inserted, thus the converstion from one MPEG file to one YUV file can be directly done by FFmpeg.

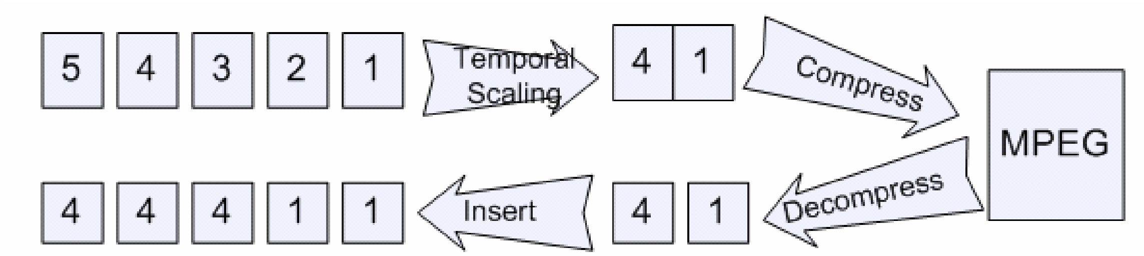


Figure 4.4. Compression and decompression for temporal scaling

4.4 Measure Power

After setting up the test bed, we measure the energy consumption of each component. None of the components enter the sleeping state during the experiment.

1. Capture power: The power used during capture ($P_{capture}$) at different frame capture rates are recorded. The video resolutions considered are 176x144, 352x288 and 640x480.

2. Encoding power: The power used during encoding ($P_{flash-encode}$) and the encoding time ($T_{flash-encode}$) for the three prepared videos are measued. Because of the memory limitation, only 2 seconds of video are used. Temporal scaling and quality scaling affect the energy consumption, and should be seen in steps 2, 3 and 4.

3. Transmission power: The power of sending (P_{send}) for different packet sizes and the different sending rates is recorded.

4. Reading and writing power: The reading power (P_{read}) and the read time (T_{read}) for different size files from the flash memory are recorded, as well as the writing power (P_{write}) and write time (T_{write}) for different size files to flash memory are recorded.

4.5 Build Model

An energy consumption model is built to analyze the energy consumption of the sensor in media scaling. Adjusting the arguments to a model can help explore the power consumption. Our model has several parameters that affect the energy consumption of the video sensor including: resolution (square pixels), capture rate (frames/second), frame distance, quantization level, sending rate (packets/second), and packet size (bytes). The estimated energy consumption is estimated from the model. The model mainly includes four parts: Base Energy, Capture Energy, Encode Energy and Send Energy. The first column has the components of the energy consumption model and the second column includes the factor, which affects each component.

Component	Parameters			
Send Energy	Send Rate	Packet Size		
Encode Energy	Resolution	Video Motion	Temporal Scaling	Quality Scaling
Capture Energy	Resolution	Frame Rate		
Base Energy	Max Time			

Figure 4.5 Energy consumption model

1. *Base Energy* (E_{base}) is the energy consumed in the idle state. The videos from the camera are piped to the encoder and finally the compressed videos are piped to the wireless card. The process time of the whole process is determined by the most time consuming component of the system. When the encoder is the most time consuming part , the situation should be like the figure 4.6. The time, which is spent on the way from camera to the encoder or from the encoder to wireless card, is small and neglectable. It is difficult to measure the overlap time between each component, but the total time is close to the longest time, the encoding time. Whereas in Figure 4.7, the bottleneck of the system is the camera. The capture time is close to the total time. So the total base energy consumption is approximately as follows:

$$E_{base} = P_{base} max(T_{capture}, T_{send}, T_{encode}) \qquad (4.5.1)$$

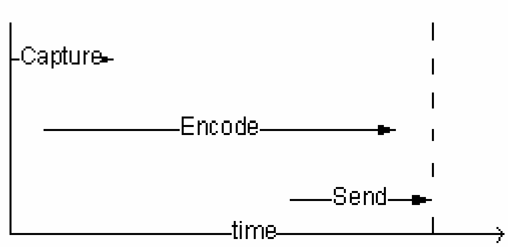


Figure 4.6 The bottleneck is encoder

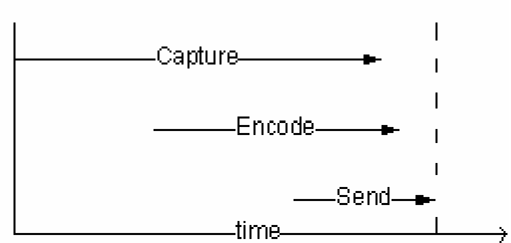


Figure 4.7 The bottleneck is camera

2. *Capture Energy* ($E_{capture}$) is determined by resolution and capture rate. After the analysis of the data from the experiment, and capturing different motion videos the relation equation can be obtained.

$$E_{capture} = f(resolution, capture\_rate) \quad (4.5.2)$$

The total energy consumption of capture, after eliminating the base energy consumption, is as follows:

$$E_{capture} = (P_{capture} - P_{base})T_{capture} \quad (4.5.3)$$

3. *Encode Energy* (E_{encode}) is determined by the video motion, the resolution, the frame distance (the effect is the same as controlling the capture rate) and the quantization level. In this study, FFmpeg compresses the three YUV videos (described in Chapter 4.2), with a resolution 352x288 and 30 frames/second to MPEG videos. This encoding process includes the cost of reading the YUV file and writing MPEG file to flash memory. In a real system, the video, coming from the camera, is compressed and sent directly over the wireless network, without the flash memory cost. So it is necessary to eliminate the flash memory cost from the model.

The total energy consumption of encoding after eliminating the base energy consumption is as follows:

$$E_{flash-encode} = (P_{flash-encode} - P_{base})T_{flash-encode} \quad (4.5.4)$$

The reading and writing energy consumption after eliminating the base energy consumption is as follows:

$$E_{read} = (P_{read} - P_{base})T_{read} \quad (4.5.5)$$

$$E_{write} = (P_{write} - P_{base})T_{write} \quad (4.5.6)$$

The total energy of encoding after eliminating the read/write energy is as follows:

$$E_{encode} = E_{flash-encode} - E_{read} - E_{write} \quad (4.5.7)$$

We can not measure the encoding power (P_{encode}) without read/write cost. However the video compression process usually has a heavy CPU load. In order to determine T_{encode} in the model, we measure the power ($P_{full-load}$) when 100% CPU is occupied (run command "perl -e '1**1 while 1'"), assuming $P_{encode} = P_{full-load}$.

4. *Send Energy* (E_{send}) is determined by packet size and send rate. The UDPsender transmits the MPEG files over the wireless network with different packet sizes and send rates. The relationship can be obtained:

$$E_{send} = f(packet - size, send - rate)$$

The total energy consumption of sending after eliminating the base energy consumption is as follows:

$$E_{send} = (P_{send} - P_{base})T_{send} \qquad (4.5.8)$$

Finally, the total energy consumption of the whole system is the combination of the four parts:

$$E_{total} = E_{base} + E_{capture} + E_{encode} + E_{send} \qquad (4.5.9)$$

We calculate the power in the energy consumption model and validate it with an actual system

$$P_{total} = \frac{E_{total}}{max(T_{capture}, T_{send}, T_{encode})} \qquad (4.5.10)$$

4.6 Validation

To verify the accuracy of the energy consumption model, we develop an actual system for the Stargate video sensor. FFmpeg captures low motion video (someone talking) from the environment and compresses it into MPEG videos. Then the compressed videos are piped to the UDPsender and sent over the wireless network. The UDPreceiver in the video master receives and saves the videos sent by the UDPsender. Frame rate, quantization level, packet size and send rate can be varied for these experiments, using 2228 bytes packet and 83 packets per second in the experiment (see chapter 5.1.3). Three experiments were conducted to test the performance of the actual Stargate system with FFmpeg.

I. The capture rate was set at 30 frames per second to test the actual frame rate over 10 seconds, 20 seconds, 30 seconds, 40 seconds and 50 seconds.

II. The power of the actual system was measured at the different capture rates and compared against the evaluation value from the energy consumption model. The effect of controlling the capture rate is the same for temporal scaling. The quantization level is set to 6. (see details in Section 5.2).

III. The power of the actual system with quality scaling is measured and compared against the evaluation value from the Stargate energy consumption model. The frame rate is set to 2 frames per second, as explained in Chapter 6.

4.7 Analyze Result

The measurement results for a variety of activity states, video capture, video transmission and video compression procedure in media scaling are discussed in Chapter 5. Also there are some video quality measurement results from media scaling. The Stargate energy consumption model is used to discuss the energy distribution in Stargate sensor system and how to improve the power performance of the system.

5) Results

This chapter describes the measurement results for the Stargate sensor and analyzes the effects of media scaling in the system. An energy consumption model is constructed based on the actual data to evaluate the power of the system. Finally the accuracy of the energy is validated by an actual system.

5.1 Power Distribution

The power of the video sensor is calculated using equation 4.1.1. The resistance of the battery ($R_{battery}$) is measured to be approximately $0.58\,\Omega$. The total power of the circuit (VI_{avg}) includes two components: the power of the battery resistance ($I_{avg}^2 R_{battery}$) and the power of the video sensor.

The description and analysis of the power measurement for the different activities is shown next. In the capture process, the Stargate video sensor is setup to capture three types of videos: low motion (a person is sitting in front of the computer and talking using a headset), medium motion (two people are talking to each other with some motion), and high motion (a person is casually playing football in the office). We measure the energy consumption of the video sensor when capturing these three types of videos. The measurement scenarios are defined as follows:

a) *Base*: The idle state or base state represents energy consumption when only the basic operating system tasks are running. This serves as a reference for other tasks. The camera and the wireless card can be turned off by commands.
b) *Full load*: The full load status represents Stargate's Intel PXA255 XScale RISC processor working intensively. An infinite loop is used to keep the processor busy in the experiment and the multimeter captures the energy consumption of the system in full load. There are no extra activities inside the loop (*perl -e '1**1 while 1'*).
c) *Read and write*: The Stargate has flash memory to store data. A program writes raw data to a file with file size as a parameter and another program reads the file without any additional operations on the content.
d) *Video capture*: The Logitech 4000 USB-based video camera captures video at a fixed frame rate for 50 seconds. The video file is not saved in flash memory.
e) *Sleep*: The command *sys_ suspend* puts the whole system to sleep for a fixed time. If you just want the web camera to sleep, one method is to just remove the module of camera

from kernel (*rmmod usbohci-sa1111*). In the Stargate, the CompactFlash slot actually is considered to be a PCMICA slot by the kernel. You can use the PCMICA control command, cardctl suspend, can be used to put the wireless network to sleep.

f) *Encode*: A file of YUV images stored in flash memory is compressed to an MPEG file by FFmpeg. The reading and writing power cost is included. The GOP pattern is "IBBPBBPBBPBB" for all videos.

Table 1 shows the measured power (watts) during different states of activity. They are the average values of 5 runs, when each run lasts for 50 seconds. In the first column, the processor, wireless network card and camera are active. The camera sleeps in the second column, the wireless network card sleeps in the third column and both wireless card and camera are sleeping in the fourth column.

The CPU occupation is the CPU utilization percentage, tested by the command "top". In the capture row, "N/A" indicates the video sensor can not capture images when the camera is sleeping.

Components Combination / Status	Processor Radio Camera	Processor Radio	Processor Camera	Processor	CPU occupation
Base (P_{base})	2.34	1.57	1.44	0.52	0%
Full load (P_{full})	3.37	2.58	2.44	1.51	100%
Read (P_{read})	3.28	2.52	2.37	1.42	99%
Write (P_{write})	3.11	2.31	2.19	1.23	66%
Capture ($P_{capture}$)	2.71	N/A	1.78	N/A	33%
Sleep (P_{sleep})	0.12	0.12	0.12	0.12	0%
Encode ($P_{flash-encode}$)	3.26	2.44	2.21	1.32	100%

Table 2. Average power (watts) in different states

The base power for the system is 2.34 watts. The full load state draws about 1 watts over the idle state. The read power is about 1 watt and more than the write power, 0.77 watts. Video capture draws about 1/3 watts at 30 frames per second. However the system only draws about 1/8 watts if in the sleeping state. Thus, periodic sleep is an effective way to improve energy consumption. Encoding draws an additional 1 watt of power.

5.1.1 Camera

Three kinds of videos, described in section 5.1, were captured. The data shows that capturing the different motion videos draws the same power. Then we measure the energy for capturing different size frames. The camera and the wireless card are active, so the base power is 2.34 watts. The Logitech QuickCam supports 640x480 video resolution. The CIF (352x288) and QCIF (176x144) are standard resolutions. The x-axis of Figure 5.1 is the capture rate (frames/second) and the y-axis is the power (watts), not including the base power. The three trend lines represent the power for the different resolutions. The high resolution videos draw more power than the low resolution videos. The power for capture grows linearly with the capture rate. The x-axis of Figure 5.2 is the total pixels (Kilo pixels) and the y-axis is the capture power (watts), not including the base power. The trend line represents the power drawn versus for different total numbers of pixels.

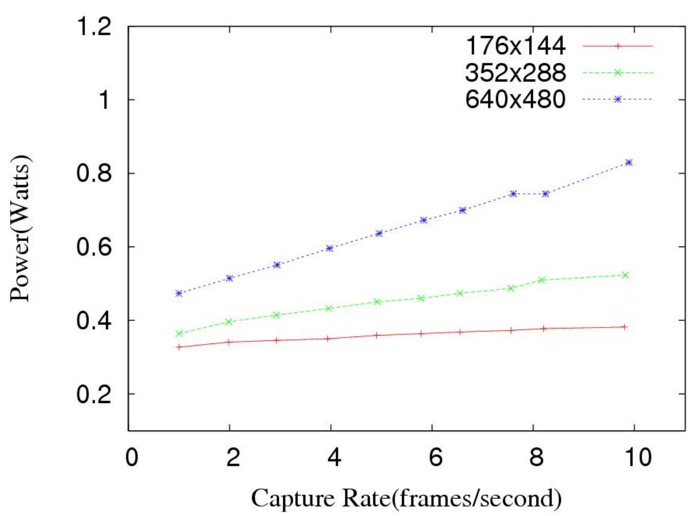


Figure 5.1 The power for the different capture rate

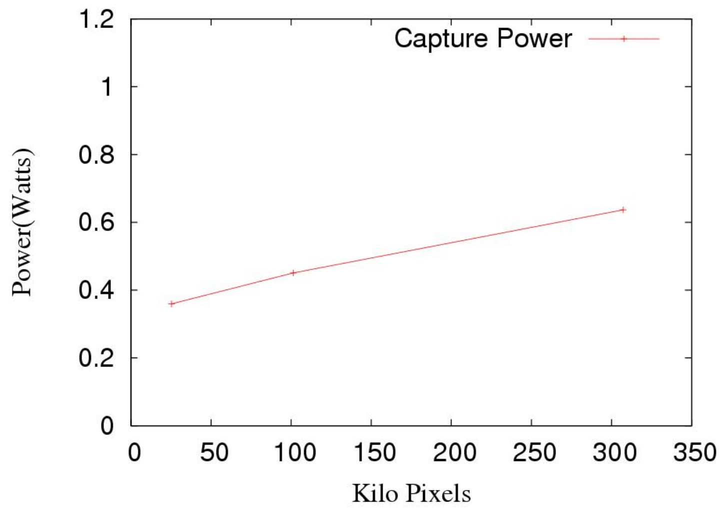


Figure 5.2 The power for the different pixels

After the analysis of the data, in Table 3, we get the relationship between the capture power ($P_{capture} - P_{base}$) and the capture rate (frames per second). Matlab fits the data to liner relation. The correlation coefficient is near 1, meaning that the linear relation equation fits the data well. The capture rate is less than or equal to 30 frames per second.

Video Resolution	Capture Power (watts)	Norm of Residuals	Correlation Coefficient	Comments
176x144	$P_{capture} - P_{base}$=0.0061x+0.33	0.008	0.97	x<=30(frames/sec)
352x288	$P_{capture} - P_{base}$=0.018x+0.36	0.019	0.98	
640x480	$P_{capture} - P_{base}$=0.039x+0.44	0.021	0.99	

Table 3. The capture power in the energy model

5.1.2 Flash Memory

The Stargate has 32 MB Flash memory and 64 MB SDRAM. The read and write operation on the flash memory make heavy use of the CPU and draw approximately 1 Watt power. Thus, avoiding I/O operation to flash memory can be an effective way to reduce energy consumption. In the experiments, the power for the read and write operation is a constant, but the read and write times vary. The x-axes of Figure 5.3 and 5.4 are the file sizes (MB) and the y axes are the times (milliseconds). The two trend lines represent the read and write times on the different size files. According to the following figures, the read and write time grow linearly with the file size.

Because of memory limitation, we choose 2 second length video in the compression process. The raw YUV file is approximately 10 MB and the MPEG file is less than 1MB. So the read file size is less than 15 MB and the write file size in less than 1MB in the experiments.

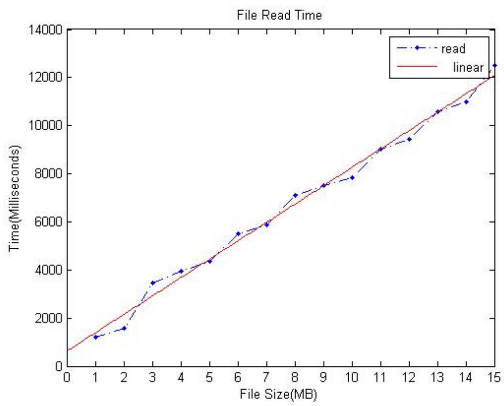


Figure 5.3 Read file from flash memory

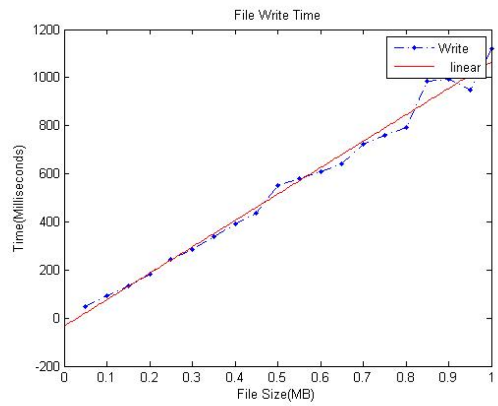


Figure 5.4 Write file to flash memory

After the analysis of the data, the relationship between the time and the file size is presented in Table 4. The correlation coefficient is nearly 1, meaning that the liner relation fits the data well.

	Time (Milliseconds)	Norm of Residuals	Correlation Coefficient	Comments
Read	T_{read} =804.5x+191.96	1568	0.99	0<x(MB)<=15
Write	T_{write} =1098x-33.73	156.2	0.99	0<x(MB)<=1

Table 4. Read and write time to flash memory

5.1.3 Wireless Network

The maximum packet size in 802.11b is 2312 bytes. If one chooses the maximum size packet, the AmbiCom wireless CompactFlash Card can send 83 packets per second at most using OS signals with only 1% CPU load. The x axis of Figure 5.5 is the packet size (Bytes) and the y axis is the power (watts). The two trend lines represent the sending and receiving power for the different packet sizes. The send rate is set to be 100 packets per second, not 83 packets per second. Since the CPU clock is not accurate, allowing the send rate can only to be roughly controlled. When the packet size is larger than 1500 bytes, the actual packet rate can not reach 100 packets per second. The basic power of sending is approximately 0.3 watts. Meanwhile, the 2228 byte packets only cost 0.04 watts more power than the 28 byte packets. The bit rate of the wireless card can be set to 1MB, 2MB and 11MB. When the Stargate video sensor attempts to send as much data as possible, the transmission power at these different bit rates are same, but it obtains the largest bandwidth at 11MB. The bit rate is set to be 11MB in all subsequent experiments.

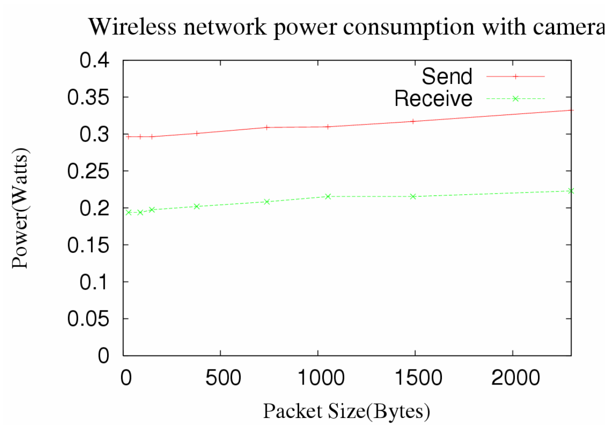


Figure 5.5 Power vs. Packet size

The power of wireless network with different send rate is measured. The packet size is 2228 bytes in this experiment. The x axis of Figure 5.6 is send rate (packets/second) and the y axis is the power efficiency (watts). The two trend lines represent the sending and receiving power for the different send rates. The largest send rate sends 10 times more data, but only costs 25% more power than the smallest send rate. Thus a higher data rate is preferable for power efficiency.

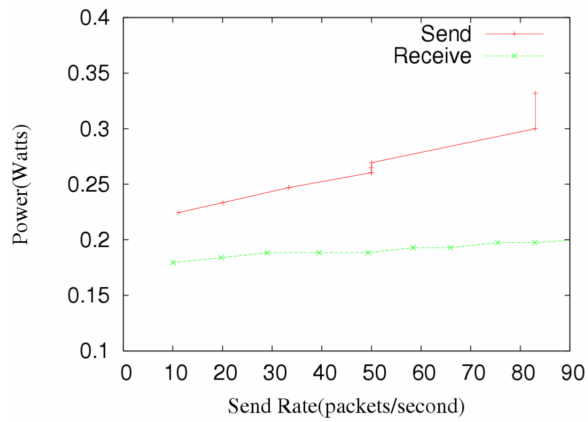


Figure 5.6 Power vs. Send rate

After the analysis of the data, in Table 5, the relationship between the power (watts) of the wireless network and the send rate x (packets/second) is generated. The correlation coefficient is nearly 1, meaning the linear relation fits the data well. The send rate is less than 83 packets per second.

	Power of transmission (watts)	Norm of Residuals	Correlation Coefficient	Comments
Send	$P_{send} - P_{base}$=0.0009x+0.216	0.012699	0.99	x(packets/sec) <=83
Receive	$P_{receive} - P_{base}$=0.00023x+0.179	0.0043238	0.95	

Table 5. The sending and receiving power with camera (watts)

5.2 Video Quality Measurement

The selection of an appropriate quantization level for temporal scaling is critical. We use a PC to measure the quality (PSNR and VQM) of the video for different quantization levels, selecting from 1 (best quality) to 31(worst quality). The three kind of video are described in section 4.2. The length of the videos used is 10 seconds. The x-axis of Figure 5.7 and 5.8 are the quantization level. The y-axis of Figure 5.7 is PSNR and the y-axis of Figure 5.8 is (1-VQM). When the quantization level is between 1 and 10, PSNR is larger than 30 db. all the videos have good quality. So we select 6 for the quantization level during temporal scaling. In the figures, the high motion videos have smaller PSNR values and bigger (1-VQM) values. The default calibration of the VQM program causes the VQM curves of videos to have some intersections as the quantization decreases.

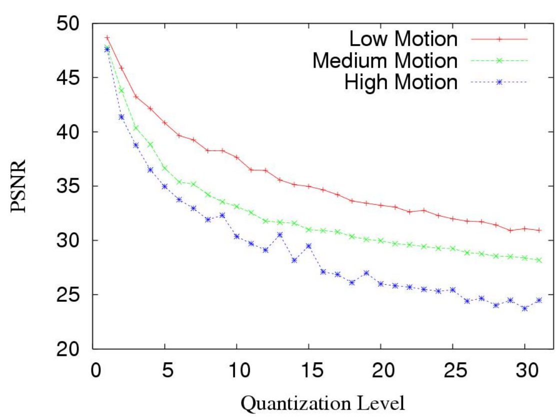


Figure 5.7 PSNR vs. Quantization Level

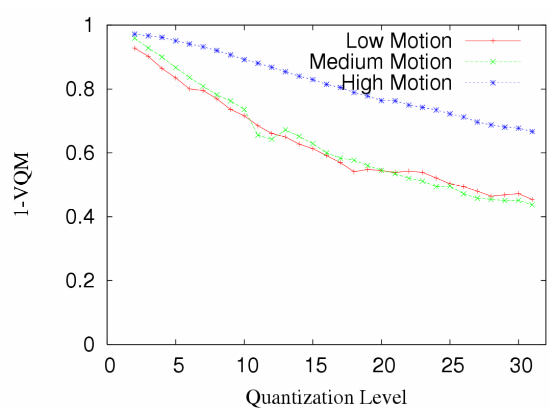


Figure 5.8 (1-VQM) vs. Quantization Level

The x axis of Figure 5.9 is the quantization level and the y axis is the bit rate (Mb/s) reported by FFmpeg. The quantization level greatly affects the data rate after compression. The bit rate, reduces from 7 Mb/s to 1 Mb/s, which can reduce wireless network energy consumption during transmission.

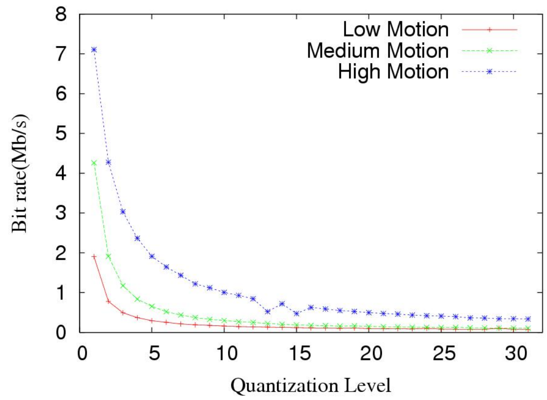


Figure 5.9 Bit Rate vs Quantization Level

5.3 Media Scaling

After measuring the three kind of videos, described in section 4.2, the power ($P_{flash-encode}$) is the same for different scaling levels and different motion features. The encode time $T_{flash-encode}$ of 60 frames of video (30 frames per second) for different temporal and quality scaling levels is measured. The energy of encoding: $E_{flash-encode} = P_{flash-encode}T_{flash-encode}$ The encoder needs to read the YUV files and write the MPEG files. The raw YUV video is 8.91 MB.

Table 6 shows that in temporal scaling (quantization level=6) level 16, 16 frames following one frame are discarded. The frame rate decreases from 30 frames per second to 1.76 frames per second. Most videos include plenty of redundancy information so that even though some frames are discarded, it does not always greatly affect the visual quality. In temporal scaling, the compression process takes less encoding time and saves a lot power. In quality scaling, the compression power is affected by the quantization level. The output MPEG file size decreases more quickly for quality scaling than for temporal scaling.

Temporal scaling	YUV size(MB)	Encoding time(s)	MPEG File size(KB)
0	8.910	20.90	129.62
2	2.970	8.38	58.70
4	1.782	4.35	37.85
8	1.040	2.94	27.94
16	0.594	2.05	20.64

Quality scaling	YUV size(MB)	Encoding time(s)	MPEG File size(KB)
2	8.910	30.72	473.931
4	8.910	25.22	208.32
8	8.910	22.56	95.14
16	8.910	18.78	47.95
31	8.910	15.71	30.17

Table 6.Encoding of a medium motion video

The x axis of Figure 5.10 is the frame distance and the y axis is the encoding time (seconds) for temporal scaling. The x axis of Figure 5.11 is the quantization level and the y axis is the encoding time (seconds) for quality scaling. The three trend lines represent the encode time for the different motion videos. They obviously show that the high motion video takes longer to encode, thus drawing more power than the low motion video. The encoding time decreases exponentially with the scaling level. The encode time $T_{flash-encode}$ drops quickly from frame distance 0 to 4 in temporal scaling and from quantization 2 to 8 in the quality scaling. That gives us some hope to improve the energy consumption of the system with temporal scaling and quality scaling.

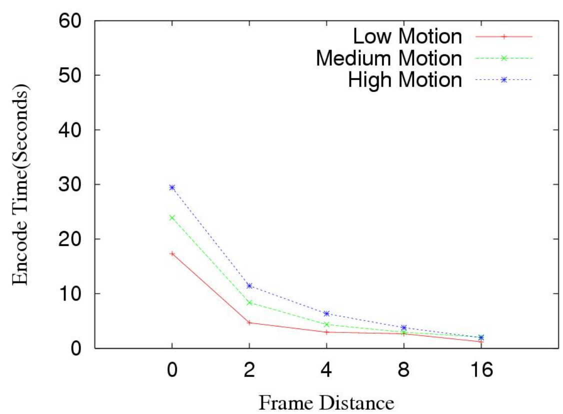


Figure 5.10 The encode time for temporal scaling

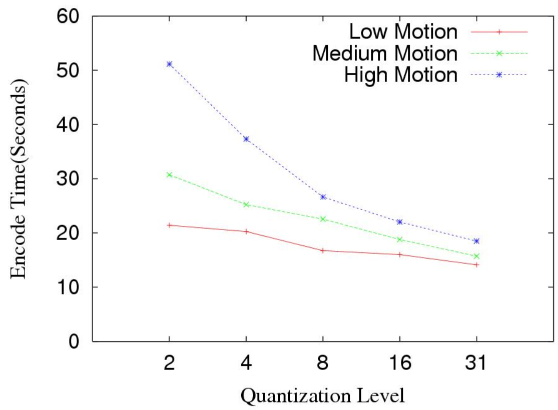
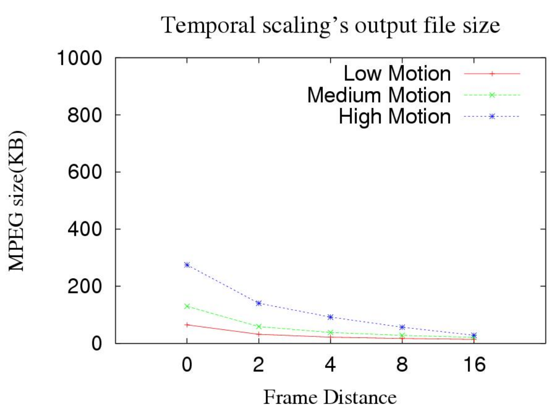


Figure 5.11 The encode time for quality scaling

The x axis of Figure 5.12 is the frame distance and the y axis is the compressed video size (KB) for temporal scaling. The x axis of Figure 5.13 is the quantization level and the y axis is the compressed video size (KB) for quality scaling. The three trend lines represent the compressed video size for the three videos. The compressed high motion video is relatively larger than the compressed low motion video. The MPEG file size decreases exponentially with the frame distance and quantization level.



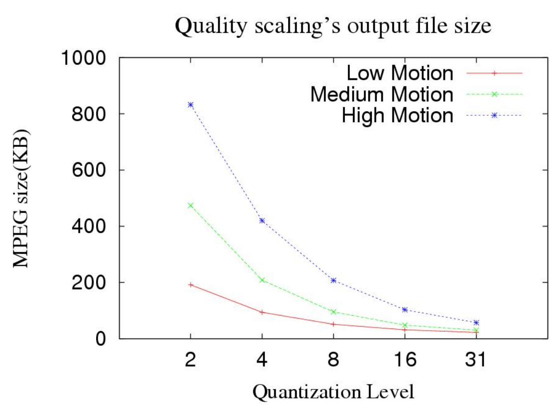


Figure 5.12 Temporal scaling vs Frame distance Figure 5.13 Quality scaling vs Quantization level

In the experiments, the encoder reads the raw video from flash memory and writes the compressed videos to flash memory. In the real sensor capture scenarios, the encoder obtains the raw video from the camera and passes the compressed video to the wireless network card for transmission with the compression done in the primary sensor memory. So it is necessary to eliminate the reading and writing files cost for the model.

We eliminate the base energy consumption from the read file operation (E_{read}), the write file operation (E_{write}) and the encoding video operation ($E_{flash-encode}$) on the flash memory:

$$E_{read} = (P_{read} - P_{base})T_{read} \quad (5.3.1)$$

$$E_{write} = (P_{write} - P_{base})T_{write} \quad (5.3.2)$$

$$E_{flash-encode} = (P_{flash-encode} - P_{base})T_{flash-encode} \quad (5.3.3)$$

We eliminate the I/O energy consumption from the encoding video energy consumption:

$$E_{encode} = E_{flash-encode} - E_{read} - E_{write} \quad (5.3.4)$$

The length of the video in the experiment is 60 frames, so the energy consumption of every frame is as follows:

$$E_{frame} = E_{encode} / 60 \quad (5.3.5)$$

The x axis of Figure 5.14 is the frame distance and the x axis of Figure 5.15 is the quantization level. The y axes of Figure 5.14 and Figure 5.15 are the energy for 2 second medium motion video. The Figure 5.14 and Figure 5.15 show the power of reading, writing and encoding 60 frames on flash memory. As the compressed video is small, the energy consumption of writing is not easily seen in the figures. However the energy consumption of reading is quite large, indicating the need for eliminating the reading and writing energy consumption in order to obtain the encoding energy.

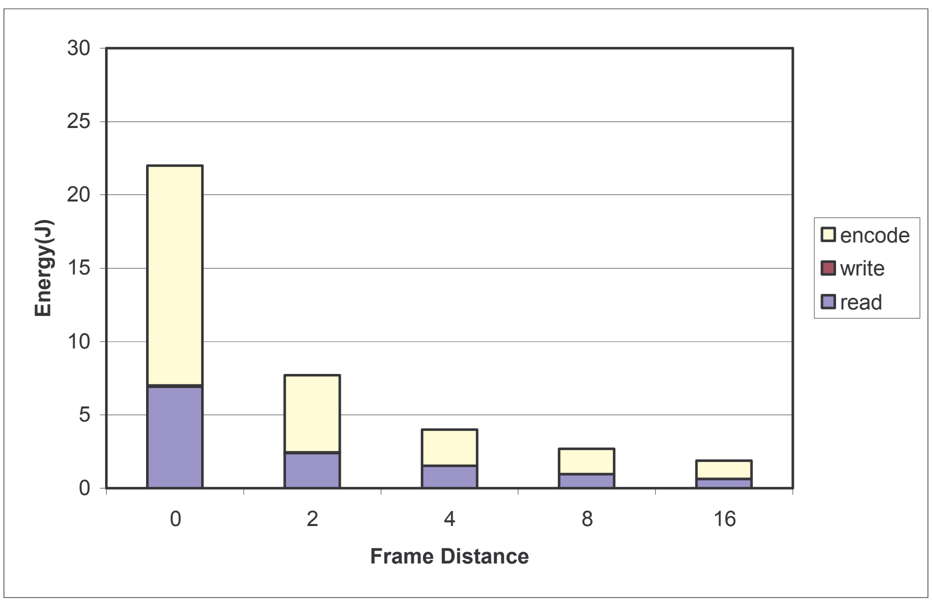


Figure 5.14 Energy consumption for temporal scaling

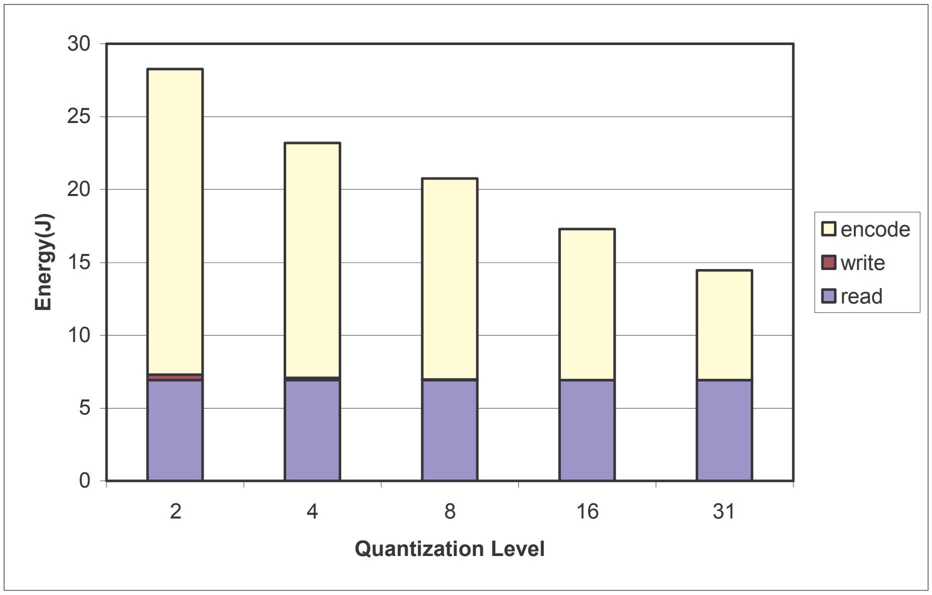


Figure 5.15 Energy consumption for quality scaling

5.4 Energy Consumption Model

The energy consumption equations for video capture, video compression and video transmission over wireless network have been generated. The energy consumption of the video sensor is affected by many factors, such as resolution, motion feature and video quality. An energy consumption model for a video sensor can help us understand the power distribution on the components and analyze the effect of media scaling on power use. Thus based on the measurement result, this section generates an energy consumption model.

Assuming the video sensor components can be processed in parallel, the base energy of the whole system is determined by the most time consuming component in the system.

$$E_{base} = P_{base} max(T_{capture}, T_{send}, T_{encode}) \quad (5.4.1)$$

In order to calculate E_{base}, we have to figure out the maximum of $T_{encode}, T_{send}, T_{capture}$. The x axis of Figure 5.16 is the frame distance and the y axis is the total processing time of medium motion video for temporal scaling (seconds). When T_{encode} is the maximum, then the encoder is the bottleneck and the raw video from camera can not be encoded in real-time. The output video data rate depends on the encoder. When the temporal scaling level is 8 or 16, the frame rate decreases from 30 frames per sec to approximately 2 or 3 frames per second, $T_{capture}$ is at its maximum and the processor does not have enough video to compress and waits for the incoming data. The junction of the encoding line and the capturing line is between a frame distance of 4 and 8. The frame rate at this junction is approximately 3 frames per second. So when the capture rate of the camera exceeds the 3 frames per second, the encoder can not catch up with the speed of the camera. The x axis of Figure 5.17 is the quantization level and the y axis is the time of medium motion video for quality scaling (seconds). In quality scaling, the encoder is always the bottleneck of the video stream.

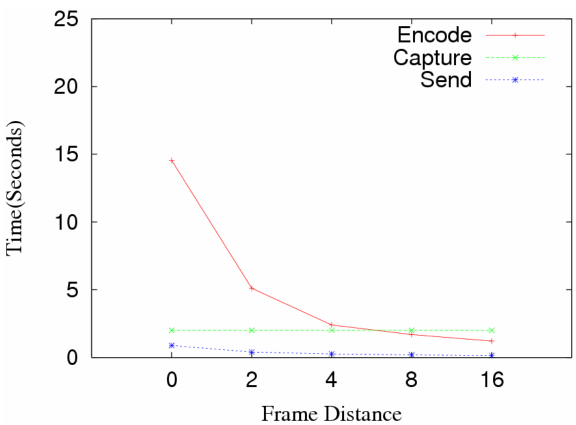


Figure 5.16 Time in temporal scaling

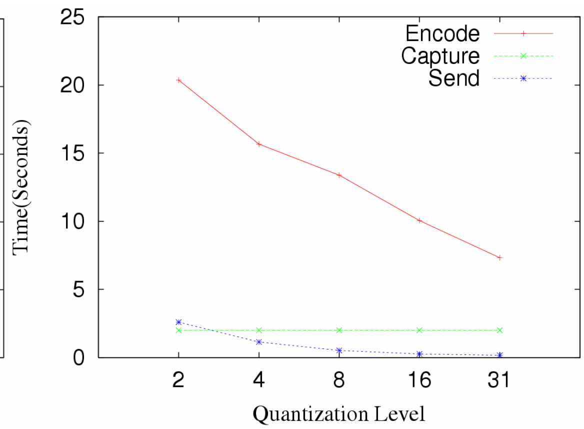


Figure 5.17 Time in quality scaling

After determining the component that takes the most amount of time for the whole video sensor process, we can evaluate the energy consumption for each part. The total energy consumption of the sensor is:

$$E_{total} = E_{base} + E_{capture} + E_{encode} + E_{send} \qquad (5.4.2)$$

The x axis of Figure 5.18 is the frame distance and the x-axis of Figure 5.19 is the quantization level. The y-axes are the energy of medium motion video. The following figure shows the energy consumption for temporal scaling and quality scaling. Most energy consumption is still occupied by the base of the system, while the compression process consumes a large part of the energy.

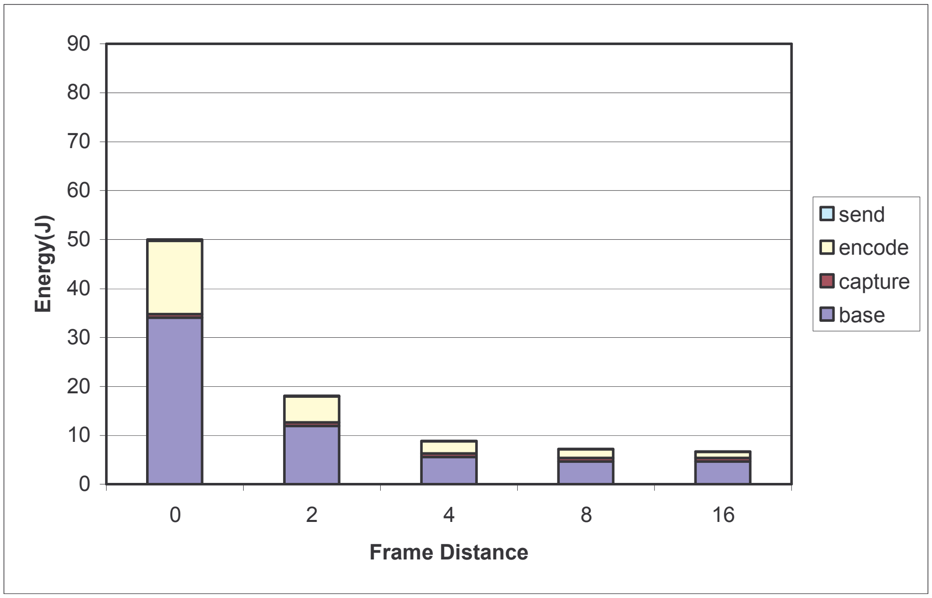


Figure 5.18 Energy for temporal scaling

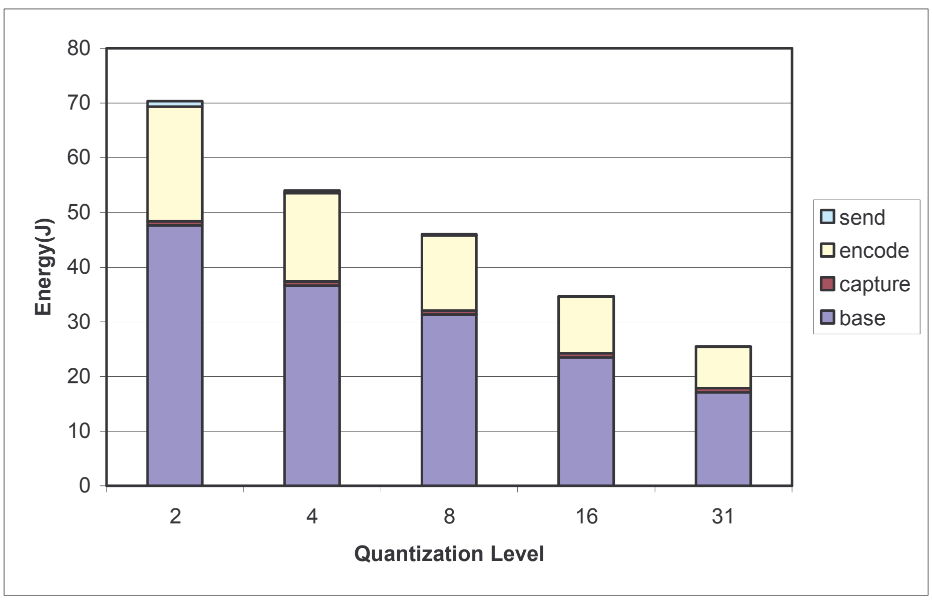


Figure 5.19 Energy for quality scaling

Now it is time to compare the energy consumption on the system with or without video compression technology. In the following table, if the raw YUV video without compression is sent over the wireless network, it costs more power than if the video is sent after compression. The main reason is the packets have to wait in a line for sending. It is a time consuming task to send the raw videos. Even though the compression process require significant power, the data in the tables 7, 8, 9, 10, 11 and 12 show that the system with video compression draws less power than the system without video compression. The second column is the total energy consumption with temporal scaling or quality scaling. The third column is the energy consumption without encoding, which is the model result. If the frame distance is 0, it costs 129 J. When the frame distance is 16, most frames are discarded and the energy consumption is only 9 J. Notice that the compression technology is sacrificing video quality for the power savings. A reasonable range of PSNR should be between 20 and 50 [FB02].

Temporal scaling	Total energy (E_{total})	Total energy without encode(J)	PSNR	VQM
0	31.62	129	40.37	0.20
2	7.19	43	26.51	0.23
4	4.81	26	24.24	0.28
8	5.81	15	23.30	0.33
16	2.33	9	20.25	0.56

Table 7. Encode low motion video with temporal scaling

Quality scaling	Total energy (E_{total})	Total energy without encode(J)	PSNR	VQM
2	44.25	129	45.75	0.07
4	40.74	129	42.53	0.14
8	29.73	129	38.88	0.23
16	27.44	129	35.30	0.41
31	21.52	129	32.09	0.55

Table 8. Encode low motion video with quality scaling

Temporal scaling	Total energy (E_{total})	Total energy without encode(J)	PSNR	VQM
0	49.99	129	36.05	0.16
2	18.07	43	23.40	0.33
4	8.89	26	21.07	0.48
8	7.21	15	19.18	0.64
16	6.71	9	16.25	0.78

Table 9. Encode medium motion video with temporal scaling

Quality scaling	Total energy (E_{total})	Total energy without encode(J)	PSNR	VQM
2	70.19	129	43.44	0.04
4	53.89	129	38.59	0.09
8	46.00	129	34.53	0.21
16	34.68	129	31.39	0.40
31	25.45	129	28.98	0.56

Table 10. Encode medium motion video with quality scaling

Temporal scaling	Total energy (E_{total})	Total energy without encode(J)	PSNR	VQM
0	69.40	129	36.26	0.06
2	28.28	43	17.23	0.62
4	15.42	26	16.20	0.73
8	9.33	15	15.19	0.82
16	4.77	9	14.49	0.88

Table 11. Encode high motion video with temporal scaling

Quality scaling	Total energy (E_{total})	Total energy without encode(J)	PSNR	VQM
2	136.95	129	43.73	0.03
4	93.97	129	39.10	0.04
8	60.63	129	34.39	0.08
16	46.26	129	30.10	0.19
31	35.25	129	26.57	0.33

Table 12. Encode high motion video with quality scaling

The x axis of Figure 5.20 is the frame distance and the y axis is the energy consumption for temporal scaling (J). The two trend lines compare the energy consumption with or without compression in temporal scaling. In temporal scaling, some frames are discarded. So using less power than the original videos. Figure 5.20 shows that video compression still can reduce even more energy consumption.

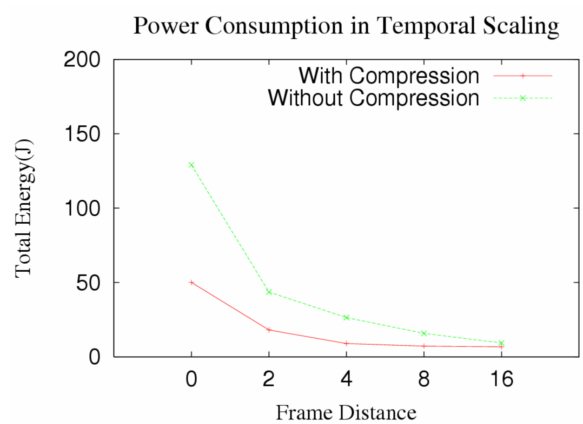


Figure 5.20 Energy consumption for temporal scaling

To trade off the energy consumption and video quality, two curves, PSNR/Energy and (1-VQM)/Energy are generated to explore the relationship. The PSNR value increases with the media scaling levels, whereas the VQM value decreases with the media scaling levels. In order to get the same trend of PSNR, we use 1-VQM to discuss the relationship between media scaling levels and video quality. The x axis of Figure 5.21 is the media scaling level and the y axis is PSNR/Energy. The x axis of Figure 5.22 is the media scaling level and the y axis is (1-VQM)/Energy. PSNR/Energy and (1-VQM)/Energy curves indicates the video quality produced by each J energy. Obviously the low quality compression provides more video quality than the high quality compression for the same amount of energy. Meanwhile the compression of low motion video produces higher video quality for the same amount of energy. Thus temporal scaling and quality scaling can more effectively improve the power in the low motion scenarios. PSNR of low motion videos drops quickly in temporal scaling. That's why the value of PSNR/Energy drops quickly at the frame distance of 8.

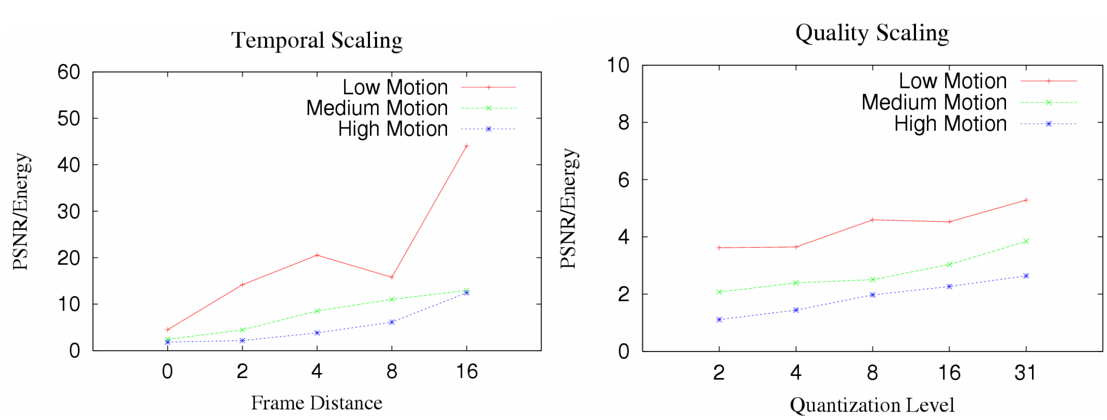


Figure 5.21 PSNR/ Energy (J) in media scaling

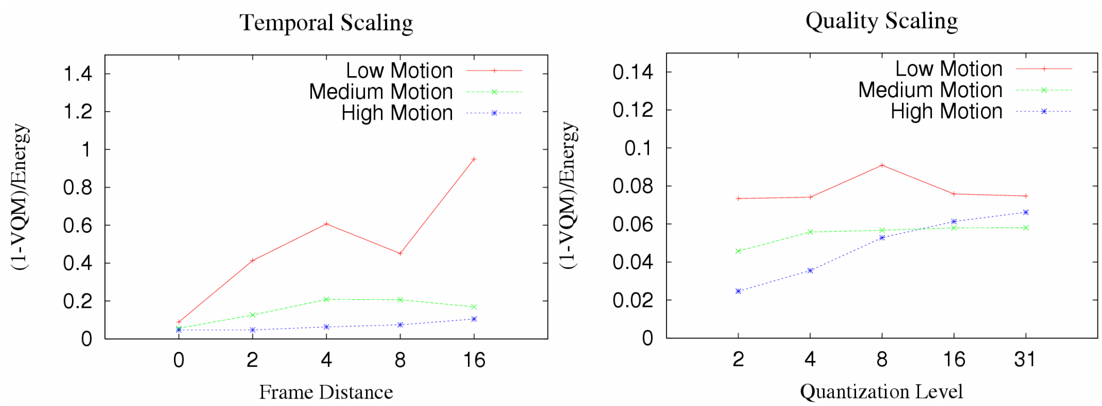


Figure 5.22 (1-VQM)/Energy (J) in media scaling

5.5 Validation

After building the energy consumption model, an actual system is designed to verify the accuracy of the model. FFmpeg captures the low motion video frames from the environment and pipes the frames into the UDPsender. The CPU utilization is between 60% and 80%. The video master receives the video from the sensor and saves it. The actual system can be described as follows:

Actual system= FFmpeg + UDPsender + Video Master

In FFmpeg, the camera always captures 10 frames per second. If we set frame rate less than 10, it will discard some frames in the camera. The x axis of Figure 5.23 is the capture time (seconds) and the y axis is the quantity of the frames. The trend line represents the number of the frames over capture time. In the following figure, the first 10 seconds only capture 90 frames, showing that the start up time of the camera is around 1 second.

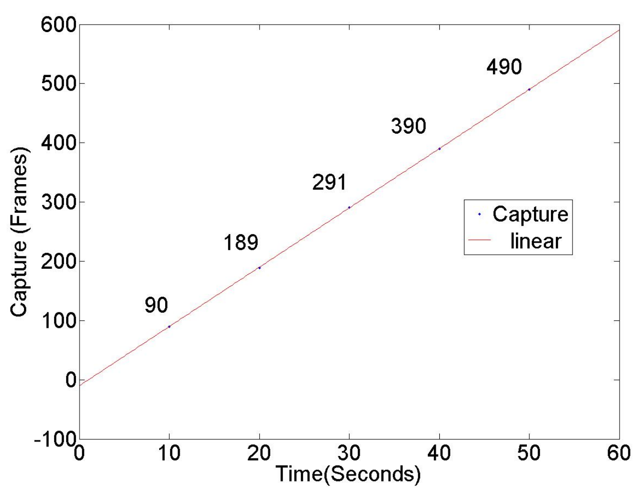


Figure 5.23. Capture frames vs. Time

Base on the measurement result of Figure 5.16, the encoder cannot encode at the camera speed if the capture rate is larger than 3. Now this fact is verified in the actual system. The x-axis of Figure 5.24 is the capture rate (frames/second) and the y-axis is the total power (watts). The two trend lines represent the power of the system for different capture rates. In the following figure, when the capture rate is over 3, the power is constant. If the CPU was faster, then the trendline would follow a course similar to the model curve.

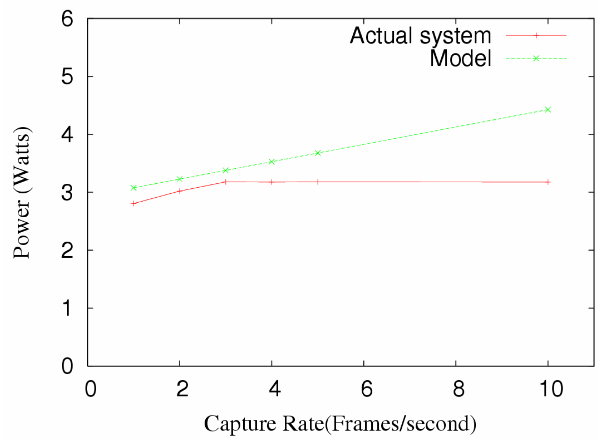


Figure 5.24 The power vs. Capture rate

The x axis of Figure 5.25 is the quantization level and the y axis is the power (watts). The camera captures 2 frames per second, of a low motion video (someone talking). The two trend lines represent the power on the different quantization level in the actual system and the energy consumption model. The difference between them is approximately 0.3 watts.

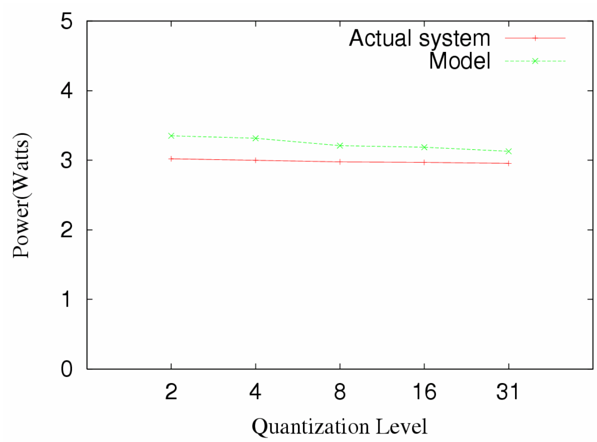


Figure 5.25. Power vs. Quantization level

The evaluation value from the model overestimates energy consumption approximately 10%. There are two possible reasons. The first reason is that some resources in the system are shared by the different components. When we combine the individual components of system in the model, the power of the shared resources are counted more than once. The second reason is that there may be some spikes in power when the states of the components are changing. Even though we use the average of the whole process in the model, the effect of the spikes can accumulate.

5.6 Power Optimization

The energy consumption model can help the system designer evaluate energy consumption before making an actual system. We discuss how the buffer technology can improve the power of a video sensor system and how improvement to the CPU helps the system save energy. The video compression could be implemented by the hardware, thus comparison of the energy consumption between the software and the hardware is discussed. Finally this section discusses the energy consumption when the wireless transmission ability is improved.

5.6.1 Buffering Technology

The investigation shows that sending faster uses less total energy. The energy consumption model can help analyze the relative power savings versus transmission bit rate. Two factors are considered for transmission: Packet Size and Sending Rate

1. Select Packet Size

The experiment shows the biggest packet (2262 Bytes) only costs 10% more power than the smallest packet (28 Bytes), but carries about 500 times more data. So the most efficient is buffering the data until it reaches 2262 bytes before sending.

2. Select Sending Rate

The x axis of Figure 26 is the media scaling level and the y axis is the sending energy for 60 frames. The two trend lines compare the sending energy with or without buffer. Most of time, the data from encoder is less than wireless network card's capability. If the data from the encoder is not buffered, the sending rate is determined by the encoding rate. Instead the data can be buffered from encoder until it reaches 83 packets (2262 bytes/packet), and then sent together. Figure 5.26 shows that the maximum send rate, 83 packets per second has a power saving advantage. So the largest packet size and the highest packet rate help save energy consumption in a video sensor.

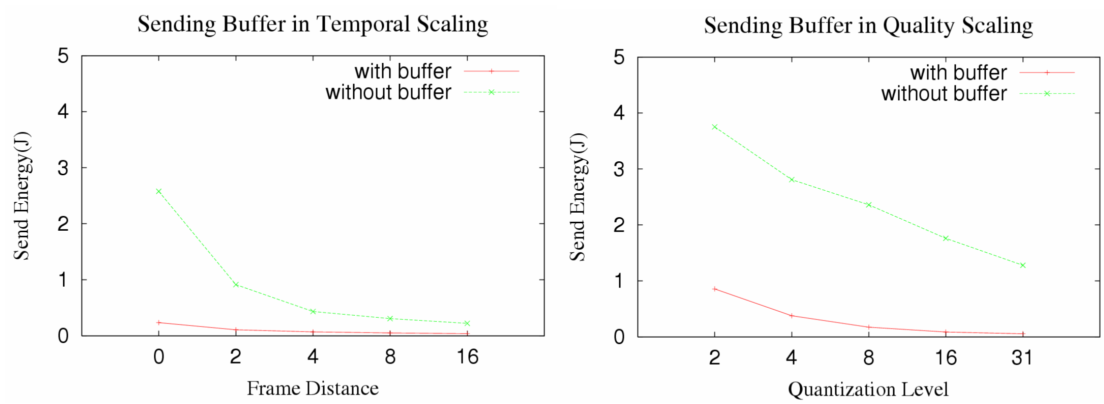


Figure 5.26. Sending energy with or without buffer

5.6.2 CPU Improvement

For the Stargate video sensor, the encoder often is the most time consuming component as in Figure 5.16. The key constraint is the CPU speed, which is unable to encode video in real time. In 10 years, the CPU speed may improve to be 2 to 5 times faster than it is now. The x axis of Figure 5.27 is the scaling level (one is Frame Distance and the other is Quantization Level) and the y axis is the energy consumption (J) of 2 second video (30 frames/sec). The three trend lines compare the energy consumption of the system when the CPU is 1, 2 or 5 times faster. According to the following result, the CPU improvement can greatly help save the power by reducing the compression time.

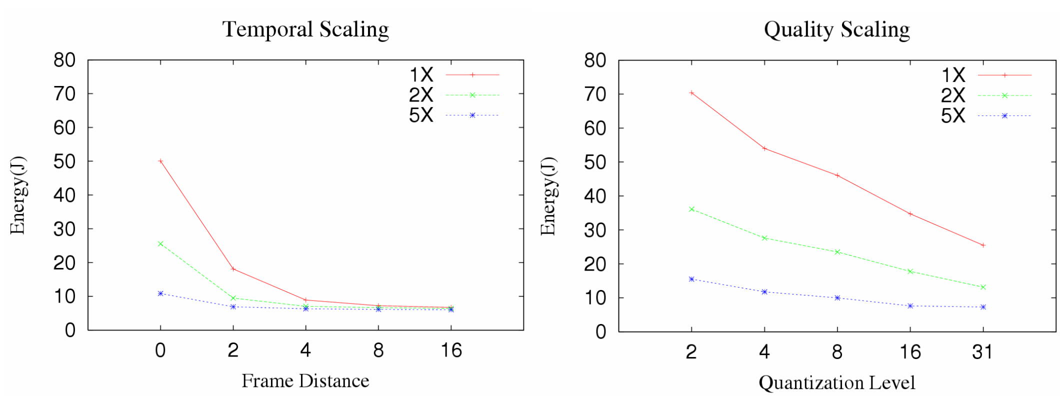


Figure 5.27. The total energy consumption when the CPU is 1, 2 or 5 times faster

5.6.3 Hardware Compression

Figures 5.18 and 5.19 have shown that the video compression has greatly improved the energy consumption of the video sensor, but compression occupies a high percentage of the total energy.

The video compression can be implemented by the hardware. As a lower bound we assume that the power of the video compression hardware is zero. The x axis of Figure 5.28 is the scaling level and the y axis is the total energy consumption (J) of 2 second video (30 frames/sec). The trend lines show that video compression implemented by hardware can improve the energy consumption in the video sensor.

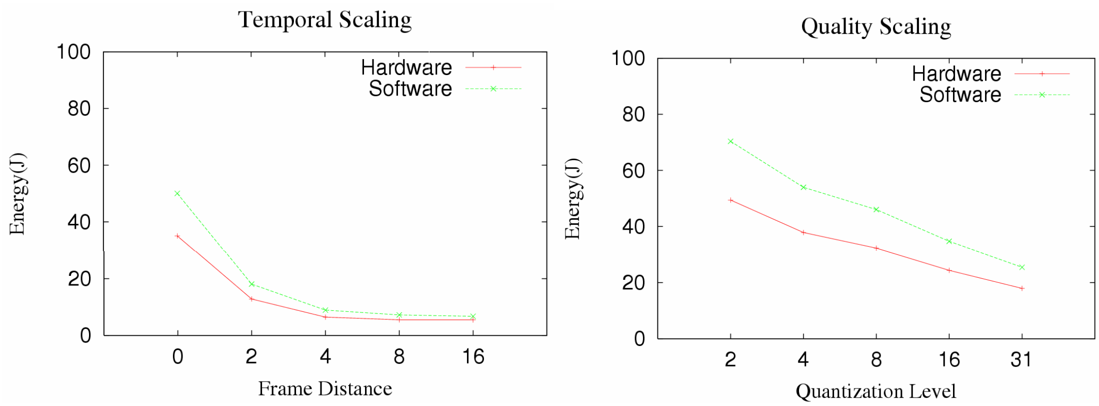


Figure 5.28 Video compression by software and hardware

5.6.4 Transmission Improvement

In Table 7, if the raw video is sent without compression, it costs 129 J. The transmission rate of wireless card can not keep up with the data rate from the camera. The raw videos have to be buffered in a queue. The data rate of the wireless network card may be improved to be 2 to 5 times in the next 10 years. The transmission power will be less, but the capture power and encoding power may be unchanged. The x axis of Figure 5.29 is the frame distance and the y axis is energy. The “with compression” trend line represents the energy consumption of the original system with temporal scaling. The other three lines represent the energy consumption without video compression when the wireless card has 1, 2 or 5 times of the current data rate. Some redundant video information is discarded. If the wireless network card is speed up 5 times in the future, the video compression has less advantage.

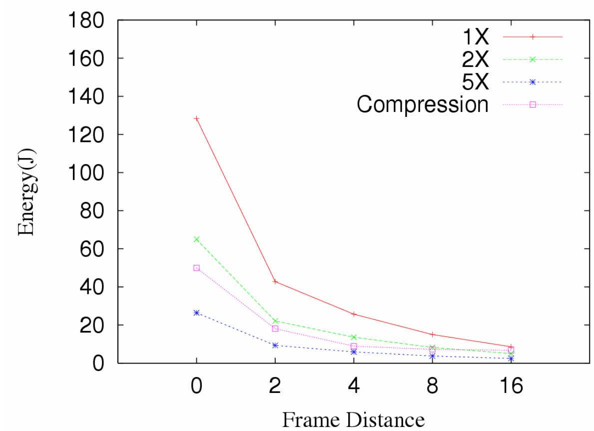


Figure 5.29. The power effect for transmission improvement

5.7 Use Case

The Stargate can be used in wireless sensor networks and robot systems. For example: The Personal Exploration Rover (PER) and the Environment Monitor (EM). The energy consumption model can help the system designer understand the power distribution in the target system. We use the energy consumption model to analyze two projects.

PER: The Personal Exploration Rover (PER) is developed by the CMU in 2004 [PER]. PER is used to look for signs of life on other planets. The rover's camera is mounted on a specially designed head that can create a panoramic image and can also detect obstacles using an optical rangefinder. Once people downloaded a panorama from PER, they can choose a rock for science testing, then estimate rover's heading and distance to reach the rock.

Figure 5.30 The personal exploration rover

During the exploring process, the camera has to continually check for unexpected obstacles enroute. If the real-time system can support 30 frames per second and high motion videos with best video quality (quantization level=2), the total power of the system is 69.7 watts, including base power (48 watts), capture power (0.4 watts), encode power (20 watts) and send power (0.9 watts). After reaching the designated location, PER illuminates the rock with ultraviolet light to look for signs of organ fluorescence, providing the mission scientist with science data. At this time the video sensor has two choices:
I. Temporal scaling: The capture rate can reduce to 2 frame per second and capture low motion videos with regular video quality (quantization level=6). The power of the system approximately is 0.875 watts, including base power (0.6 watts), capture power (0.03 watts), encode power (0.23 watts) and send power (0.015 watts).

II. Quality scaling: The Stargate video sensor can capture 30 frames per second and low motion videos with poor video quality (quantization level=31). The power of the system is approximately 10.4 Watts, including base power (7 watts), capture power (0.4 watts), encode power (3 watts) and send power (0.024 watts).

EM: An environment monitor in the zoo can help the animal feeders to take care of animals in the wild. Most animals are more active during the day. The real system can support 30 frames per second and medium motion videos with normal video quality (quantization level=6), the total power of the system is 25 watts, including base power (17 watts), capture power (0.4 watts), encode power (7.5 watts) and send power (0.1 watts). When the animals are sleeping during the night, the sensor can use temporal scaling and quality scaling to improve the power performance.

I. Temporal scaling: It can reduce the capture rate to 2 frame per second and capture medium motion videos with regular video quality (quantization level=6). The power of the system approximately is 2.96 Watts, including base power (2.3 watts), capture power (0.03 watts), encode power (0.6 watts) and send power (0.03 watts).

II. Quality scaling: The video sensor can capture 30 frames per second and medium motion videos with poor video quality (quantization level=31). The power of the system is approximately 12.1 watts, including base power (8.5 watts), capture power (0.4 watts), encode power (3.2 watts) and send power (0.024 watts).

Temporal scaling and quality scaling help the system save energy consumption. The energy consumption reduction of temporal scaling is better than quality scaling.

6) Conclusions

Most of surveillance video sensors, installed in the open air, are powered by batteries for convenience. The lifetime of the battery is constrained. Video data is usually large in storage size and affects the duration of the transmission. This study indicates that the wireless transmission of video sensor can cost a lot of energy. Even after compression, wireless multimedia streams can saturate the network. Media scaling has achieved significant improvements when used within an available bandwidth. This thesis explores the energy use of a video sensor with media scaling.

The goal of the thesis was to understand the power distribution on the sensor components and the ability of media scaling to improve the energy consumption of the wireless video sensor. To isolate the power on the components of the video sensor, power is measured under different sensor states. Then the energy consumption of media scaling is analyzed. In order to evaluate the power performance of media scaling from the view of the whole system, an energy consumption model is built to evaluate the total energy of the video sensor. Most video compression is lossy, so it is necessary to measure the video quality to truly understand the trade-offs. In order to validate the energy consumption model, an actual system was built to verify the accuracy of the model. Comparing the model to the system, the energy consumption model appears to estimate the power of system accurately with a difference of approximately 10%.

The analysis data in chapter 5 shows the power performance of the different components. The capture power grows in steps of the capture rate and the pixels. High capture rate and the high resolution require more energy consumption. Experiments of video compression by FFmpeg on the Stargate processor shows that the frame distance and the quantization level affect the energy consumption. The large frame distance and low quantization levels draw less energy. The measurements of the video quality show that quality scaling yields better video quality than temporal scaling as measured by PSNR and VQM. The measurement of transmission shows that sending the data as fast as possible can help reduce the energy consumption, but the Stargate video sensor can only support 83 packets per second for the largest sized packet.

The energy consumption model is used to analyze the energy consumption of the video sensor. The Stargate processor is not fast enough for real-time video compression. Since the encoder can only support 3 frames per second. If the processor can be made faster, the power performance of the system will be improved. Temporal scaling and quality scaling can help the video sensor improve power performance. Temporal scaling reduces the energy consumption up to 65% and quality scaling reduces the energy consumption up to 50%. The energy consumption model shows that temporal scaling has better power performance than quality scaling. However quality scaling preserves more video quality than temporal scaling.

Additionally, the energy consumption model is used to consider several power optimization methods. A buffer mechanism can guarantee the data is always sent at the fastest speed. A 10% increase in power provides 500 times the data rate, thus sending as fast as possible can improve total energy consumption. The speed improvement of a processor can help the video sensor save energy consumption. Another improvement is to implement the video compression in the hardware. This saves on processor resources and reduces the energy consumption. If the transmission speed of the Stargate sensor is improved, the data does not need to be buffered before transmission. Thus the energy consumption of the video sensor is improved.

The energy consumption model can help analyze the energy consumption of the Stargate sensor when it is used for other applications. People can consider power in various situations and extend the lifetime of the battery. In two use cases, PER and EM, the energy consumption model gives the user an objective value for temporal scaling and quality scaling. It is easy for designer to apply special techniques according to the application environment.

The result of this thesis is useful for researchers to understand the performance of the different components in a video sensor. The energy consumption model can be used to explore the trade-off between power and video quality.

7) Future work

In the future, we can compare the energy consumption of the different compression technologies such as JPEG and H.261, on the video sensor. It would be interesting to investigate combining temporal scaling and quality scaling to improve the trade-off between the energy consumption and the video quality. Temporal scaling has better power performance than quality scaling, while quality scaling has better video quality performance than temporal scaling. So potentially temporal scaling and quality scaling can complement each other and improve the integrative performance of the video sensor.

The USB can only support 12 Mbites/s includes USB packet header overhead. For a typical web camera capturing 4:2:0 YUV data at 320x240 pixel resolution, the theoretical maximum frame rate sustainable is only 13 frames per second over USB. It might be feasible to implement temporal scaling and quality scaling inside the firmware of the camera. Sensors can control the capture rate and the video quality on the camera, without using too much CPU utilization. The energy consumption of the video sensor would be improved. On the other hand, the reduction of the data rate can help avoid the USB speed limitation.

The data transmission draws a lot of energy in Stargate video sensor. The MAC protocol determines the activities of the wireless card. It is feasible to improve the MAC protocol for the energy saving. The dynamic sleep scheme on the MAC protocol is one research direction. The wireless card can be turned off when no data is sent.

The video compression reduces the size of the video and save the energy for the video sensor, but the videos have to be waiting for the compression. The extended delay will affect the performance of the video sensor. Delay is important for a real time video sensor in the surveillance scenarios. The trade-off between the energy consumption and the delay has to be explored.

References:

[BD] M. Baldauf, S. Dustdar and F. Rosenberg. A Survey on Context-aware System. *International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, Vol. 2, No. 4, pp. 263-277, 2007.

[BLM+04] J. Boice, X. Lu, C. Margi, G. Stanek, G. Zhang, R. Manduchi, K. and Braczka. Meerkats: A Power–Aware, Self–Managing Wireless Camera Network for Wide Area Monitoring. Technical Report ucsc-crl-05-04, University of California, Santa Cruz, 2005.

[CH04] C.-K. Chang and J. Huang. Video Surveillance for Hazardous Conditions Using Sensor Networks. *Networking, Sensing and Control, 2004 IEEE International Conference* , Vol. 2, pp. 1008-1013 , 2004.

[CLJ01] C. Guo, L.C. Zhong, and J.M. Rabaey. Low-Power Distributed MAC for Ad Hoc Sensor Radio Networks. *Global Telecommunications Conference*, Vol. 5, pp. 2944-2948. Nov. 2001.

[CMUcam] The CMUcam Vision Sensors. http://www.cs.cmu.edu/~cmucam/.

[CPV03] S. Coleri Ergen, A. Puri and P. Varaiya. Power Efficient System for Sensor Networks. *IEEE International Symposium on Computers and Communication (ISCC03*), Kiris-Kemer, Turkey**,** Jul. 2003.

[CV06] S. Coleri Ergen, P Varaiya. Energy Efficient Routing with Delay Guarantee for Sensor Networks. University of California, Berkeley, CA, Jan. 2006.

[DCT] The Discrete Cosine Transform. http://cobweb.ecn.purdue.edu/~ace/jpeg-tut/jpgdct1.html.

[DWT] Discrete Wavelet Transform. http://en.wikipedia.org/wiki/Discrete wavelet_ transform.

[EC] Electric Circuit. http://en.wikipedia.org/wiki/Electric_current.

[FB02] Nick Feamster and Hari Balakrishnan. Packet Loss Recovery for Streaming Video*. The 12th International Packet Video Workshop(PV2002)*, Pittsburgh, PA, Apr. 2002.

[FBF05] W. Feng, N. Bulusu and W. Feng. Dissecting the Video Sensing Landscape. Porland State University, *Network and Operating System Support for Digital Audio and Video (NOSSDAV),* Skamania, Washington, Jun. 2005.

[FC] Fractal Compression. http://en.wikipedia.org/wiki/Fractal_compression.

[FCK+03] W. Feng, B. Code, E. Kaiser, M. Shea. W. Feng and L. Bavoil. Panoptes: Scalable Low-Power Video Sensor Networking Technologies. *ACM International Conference on Multimedia (MM'03),* Berkeley, CA, Nov. 2003.

[FFMPEG] FFmepg, Open Source Encoder, http://ffmpeg.mplayerhq.hu/.

[GK06] S. Gurun and C. Krintz. Energy Characterization of the Stargate Sensor Network Gateway. University of California, Santa Barbara, Aug. 2006.

[GW02] A. J. Goldsmith and S. B. Wicker. Design Challenges for Energy Constrained Ad Hoc Wireless Networks. *Wireless Communications, IEEE*. Vol. 9, No. 4, pp. 8–27, Aug. 2002.

[HDB04] B. Hohlt, L. Doherty and E Brewer. Flexible Power Scheduling for Sensor Networks. *Information Processing in Sensor Networks (IPSN)*, Berkeley, California, Apr. 2004.

[JWC04] A. Jain, Y.-F. Wang, and E. Y. Chang. A Collaborative Camera System for Surveillance. UCSB Technical Report, Nov. 2004.

[JWW+04] L. Jiao, Y. Wu, G. Wu, E. Y. Chang, and Y.-F. Wang. The Anatomy of A Multi-camera Security Surveillance System. *ACM Multimedia System Journal Special Issue*, Vol.10, No.2, Oct. 2004.

[KGS+05] P. Kulkarni,D. Ganesan and P. Shenoy. The Case for Multi-tier Camera Sensor Networks. *Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Stevenson, Washington, Jun. 2005.

[KWF03] C. Krasic, J. Walpole and W. Feng. Quality-Adaptive Media Streaming by Priority Drop. *Network and Operating System Support for Digital Audio and Video (NOSSDAV),* Monterey, California, Jun. 2003.

[LS05] D. Lymberopoulos and A. Savvides. XYZ: A Motion-Enabled, Power Aware Sensor Node Platform for Distributed Sensor Network Application. *Information Processing in Sensor Networks (IPSN)*, Los Angeles, California, Apr. 2005.

[MPEG] MPEG video. http://en.wikipedia.org/wiki/MPEG.

[NKA05] L. Nachman, R. Kling, R. Adler, J. Huang, and V. Hummel. The Intel Mote Platform: A Bluetooth-based Sensor Network for Industrial Monitoring. *Information Processing in Sensor Networks (IPSN),* Los Angeles, California, pp. 437–442. Apr. 2005.

[P06] V. Petkov. Using the DS2438 Battery Monitor on Crossbow's Stargate, Feb. 19, 2006.

[PER] The Personal Exploration Rover. http://www.cs.cmu.edu/~myrover/PER/.

[PHH98] C. Perkins, O. Hodson and V. Hardman. A Survey of Packet-Loss Recovery Techniques for Streaming Audio, *IEEE Network,* Vo. 12, No. 5*,* Sep. 1998.

[PSNR] PSNR. http://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio.

[RBI+05] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin and M. Srivastava. Cyclops: In Situ Image Sensing and Interpretationin Wireless Sensor Networks, *The 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys'05),* San Diego, California, Nov. 2005.

[Stargate] Crossbow Technology INC., http://www.xbow.com/.

[SFN+07] P. Sitbon, W. Feng, N. Bulusu and T. Dang. SenseTK: A Multimodal, Multimedia Sensor Networking Toolkit. *The Fourteenth Annual Multimedia Computing and Networking Conference (MMCN'07)*, San Jose, California, Jan. 2007.

[SP06] S. C. Ergen and P. Varaiya. PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks, *IEEE Transactions on Mobile Computing*, Vol. 5, No. 7, Jul. 2006.

[Throughput] The Throughput of the 802.11b, http://www.bbwexchange.com/wireless_inte rnet_access/802.11g_wireless_internet_access.asp.

[TinyOS] TinyOS for sensor networks. http://www.tinyos.net/.

[VQM] Video Quality Metric. http://www.its.bldrdoc.gov/n3/video/VQM_sofware.php

[Wireless] Wireless LAN Standard, http://en.wikipedia.org/wiki/802.11.

[W04] Y. Wen. Smart Dust Sensor Mote Characterization, Validation, Fusion and Actuation. University of California, Berkeley, Nov. 2004.

[W06] Y. Wang. Survey of Objective Video Quality Measurement. WPI-CS-TR-06-02*,* Worcester Polytechnic Institute, May 2006.

[WCK05] H. Wu, M. Claypool and R. Kinicki. Adjusting Forward Error Correction with Quality Scaling for Streaming MPEG, *Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Stevenson, Washington, Jun. 2005.

[WCK06] H. Wu, M. Claypool and R. Kinicki, On Combining Temporal scaling and quality scaling for Streaming MPEG, *Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Newport, Rhode Island, May 2006.

[WGC+] Y. Wen, S. Gurun, N. Chohan, R. Wolski and C. Krintz. SimGate:Full-system, Cycle-Close Simulation of the Stargate Sensor Network Intermediate Node.

Embedded Computer Systems: Architectures, Modeling and Simulation, 2006 International Conference, Samos, Greece, Jul. 2006.

[WIRELESS] Wireless LAN standard. http://en.wikipedia.org/wiki/802.11.

[WJD04] W. Ye, J. Heidemann, and D. Estrin. Medium Access Control with Coordinated Sleeping for Wireless Sensor Networks, *IEEE/ACM Trans. Networking*, Vol. 12, No. 3, Jun. 2004.

[YG03] D. Yang. H. Gonzalez-Banos, and L. Guibas. Counting People in Crowds with a Real-time Network of Image Sensors, *IEEE ICCV Workshop*, Nice, France, Oct. 2003.

[YN05] Z. Yang and K. Nahrstedt. A Bandwidth Management Framwork for Wireless Camera Array. *Network and Operating System Support for Digital Audio and Video (NOSSDAV),* Stevenson, Washington, Jun. 2005.