ENERGY EFFICIENT AND LOSS RESILIENT WIRELESS CAMERA SENSOR NETWORKS

By

ASHISH VIJAY WANI

Bachelor of Engineering in Electronics and
Telecommunication
University of Pune
Pune, Maharashtra, INDIA
2009

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December 2011

ENERGY EFFICIENT AND LOSS RESILIENT WIRELESS CAMERA SENSOR NETWORKS

Thesis Approved:

Dr. Nazanin Rahnavard

Thesis Advisor

Dr. Weihua Sheng

Dr. Gouliang Fan

Dr. Sheryl A. Tucker

Dean of the Graduate College

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

The future of technology has been envisioned as that of a Real World Web, analogical to the World Wide Web or the Internet as we know it. A Real World Web, will be an era of ubiquitous computing; an intelligent environment where everything in the surrounding is capable of being monitored through devices as ubiquitous as the current form of dust, that will be called *smart dust.* They will act as electronic nerve ends of the planet ; a large scale network of small devices capable of harvesting information from the physical environment, processing it and transmitting it to a remote location. The monitoring might include locations as routine as our homes, cars, furniture, to keep a check on their operational condition; to places as inaccessible and hard-to-reach as international borders for monitoring tress-passers or even our own bodies, monitoring our physical condition such as the heart rates and blood sugar levels and capable of communicating this information to an appropriate source. This looks like a very promising goal for technology, but before these systems are assigned to perform many of the critical tasks, first number of issues and challenges need to be dealt with.

Wireless Sensor Networks (WSNs) have been considered as the precursors of this future. WSNs are battery powered devices that integrate the task of sensing, processing and communicating the data. The structure of sensor networks is such that they can be deployed over the area to be monitored, either manually or they can be strewn over large stretches of lands from aeroplanes. Traditional WSNs measure scalar data such as temperature, pressure, humidity, lighting conditions etc, and over a few years have found applications in precision agriculture [1], industrial monitoring [2], Ecology monitoring [3], and more recently for applications like parking space monitoring [4]. A detail discussion on the research history and trends is given in [5] . A detail survey on the different technical and architecture related aspects of WSNs has been covered in [6].

Even though designing and implementing WSNs is a challenging task, most of its applications up till now in general have been on low bandwidth data, such as temperature, pressures, humidity sensing etc. But if sensor networks are to be developed for true ubiquitous sensing, the devices should be made capable of capturing and processing high level data, such as images, videos, sounds etc. This high bandwidth data creates many new challenges, like availability of low cost sensor hardware for cameras and microphones and larger bandwidth requirement for data transmission. It will also require better and more powerful processors and larger memory units to perform in-node processing of the data. Still, a more difficult challenge would be to find a trade-off between the energy consumed for transmission and the quality of service (QoS). For high bandwidth data, the consumption increases due to increase in amount of processing to some extent but largely due to the increase in the amount of transmission. The QoS is important because it determines the quality of information obtained.

With the advance in semiconductor technology, the challenges in hardware have been moderated, with ample production of many low-cost, good quality CMOS cameras, microphones and memory units. This has given impetus to the development of Wireless Multimedia Sensor Networks (WMSNs) , which are devices capable of capturing, processing and communicating high level multimedia data, which includes different forms of audio-visual data. The ability of WMSNs to process such data makes them useful for applications such as surveillance and monitoring especially in difficult terrains, wildlife monitoring, etc. As mentioned earlier, the real utility of WMSNs would be when they are capable of operating for a long period of time with a reasonable of quality of service.

Wireless Camera Sensor Networks (WCSNs) are a category of WMSNs, which deal with snapshot data or streaming videos. One of the biggest applications of WCSNs is in surveillance and monitoring applications. The main requirement for this application would be that the sensor nodes operate for a long period of time, providing images with reasonable quality, so that adequate information about the area being monitored can be obtained. Their environment of operation is also likely to be harsh and given the limited power source, packet loss may occur in considerable amounts which can be a big problem, resulting in a deteriorated image quality and may convey no useful information as a result. Forward Error Correction (FEC) and Automatic Repeat Request (ARQ) based schemes are available to combat erasures, but they are complex and impose large processing overheads. Also, retransmission takes more power resulting in more battery energy consumption.

As part of this thesis a scheme for energy efficient and loss resilient camera sensor networks has been implemented. The idea has been developed by keeping the application for surveillance and monitoring in mind. The efficiency of the system has been improved by a two-staged conservation scheme. The first is at the sensing stage, where the energy spent idle-sensing is conserved by a trigger-driven wake-up of the sensor node. The event trigger is provided by a passive infrared sensor (PIR), which wakes up the main sensor board which is in a sleep mode. Considering the sparseness of activity expected by the WCSNs in surveillance application, the trigger driven scheme is advantageous over a schedule driven scheme. The second level of conservation is in transmission, where *Compressive Sampling* has been used to develop an efficient transmission scheme. Compressed Sampling (CS) provides an error resilient mechanism and still manages to be efficient in terms of transmission. This work of implementing CS on WCSNs forms a prominent part of this thesis.

CS turns out to be a good solution for the packet erasure and efficient transmission, but there are many challenges in implementing it on the sensor node as well as in the recovery at the base-station. The thesis has a detailed discussion about the concept of CS, its implementation on the sensor node hardware and the recovery at the base-station. CS recovery is done by convex optimization schemes, which is time consuming and also takes a lot of processor memory. Modifications have been made in the conventional CS scheme, to make it feasible to reconstruct on the available processors and also in a acceptable time-frame. There have been a few previous works which mainly talked about *why* CS should be used, but this work focusses on *how* to use CS on WCSNs to make it practically feasible.

In the next two chapters both the power saving strategies will be discussed for their relevance, their theoretical background and related works, implementation and results. The implementations have been devised to be close to the practical requirements and as mentioned earlier, the application of the system has been envisioned for surveillance applications and the operational assumptions have been made accordingly.

CHAPTER 2

Event Driven Camera Sensor Network

2.1 Importance of Event based wake-up

Sleep scheduling is a standard strategy of energy conservation employed on embedded systems and is even more relevant in WSNs. WCSNs have been envisioned to be a vital asset in surveillance and monitoring applications especially in hard-to-reach, infrastructure less terrains, such as unfenced international borders, forest areas, mountainous regions or places of strategic importance. All these places are expected to monitor activities such as unwanted intrusions, which are temporally *sparse* and *critical*. Considering this application, most of the time the sensor nodes may remain idle, but whenever an event occurs, it must be detected and captured (here an image) with a very high probability . In order to do that, keeping the nodes just idly sensing would be a gross wastage of the already energy constrained devices.

Depending on the application, different network architectures can be employed [7]. A single tier homogenous architecture can have all sensors equipped with a camera, while multi-tier, heterogenous architecture , there can be a combination of camera sensors and intermediate sensors acting as scalar and transceiving units. In any case, the current condition of the sentry nodes, i.e. their position, orientation, field of view etc are important in determining the information captured by these node.

Keeping the sentry (sensing nodes, here the camera sensors) nodes on a simple time-scheduled wake-up cycle is not reliable as the events occurring in the aforementioned places are not deterministic and there is a probability of the event being missed in the defined time schedule. There are better collaborative time based scheduling mechanisms which can provide better system performances. Most of the techniques used were developed as MAC protocols for channel sharing, which can also be used in this case. Noteworthy among them are the SMAC [8] ,TMAC [?], LEACH [9]. The problem here is, as previously mentioned; since these are not scalar sensors, the current position and orientation information about the camera sensors is necessary for co-ordinating the sleep cycles. This makes the in-network camera calibration extremely critical. Thus, a large and accurately calibrated network would be required which is a difficult task. The sparse and intermittent nature of activity experienced in surveillance applications makes it even more difficult to set the duty cycle for each sensor node.

An event-driven wake-up operation, where the sensor nodes wake-up whenever an event is detected, is a better choice especially for the proposed application. Event based operations have proven to be useful in implemented projects such as the Vigilnet project [10] and ExScal [11]. The implementations in these projects proves the effectiveness of event-based wake-up as a power-saving mechanism. But, these projects have been traditional scalar sensor networks. WCSNs have different and more expensive consumption characteristics, in sensing and especially in communication, hence effective experimentation should be done before implementing any specific scheme. For traditional sensor networks, sensing has been a more significant consumer of energy than transmission because the number of bits transmitted per event are small because of low-bandwidth data like temperature, pressure or humidity. This is not the case with multimedia data which requires large number of bits per transmission. Hence for camera sensors the significance of transmission energy goes on increasing with larger images. Thus the role of transmission also needs to be analyzed while studying wireless camera sensor networks.

As a part of this thesis, implementation of an event based wake-up on the Imote2 based WCSN platform has been discussed. The hardware (Imote2 including the multimedia sensor board) and the software (TinyOS-2.x implemented in nesC) are very suitable for an event-driven application. In this part it is shown how to judiciously use these available resources to implement an event triggered snapshot capture and transmission. All these efforts are towards developing a practically functional camera sensor node. A lifetime analysis of the system has been performed to evaluate the effectiveness of the event-based operation on the Imote2 platform, as compared to a schedule driven one.

2.2 Related Work

Event-based wake-up is an intuitive choice for the surveillance and monitoring application. The event-based wake-up requires some low-power sensor to be in an on-state in order to monitor and generate a wake-up event. Thus, the WSN cannot be put in the deep-sleep mode, it needs to be in some higher state so that the low-power sensing can be done. This has been used in projects like Exscal [11], where the activity of the sensed field is intermittent. The SensEye [12] has been a very good project using both high and low level sensors for event detection in WCSNs. It uses a low resolution camera, to detect an object of interest. The image is captured and processed on-board, and if an object of interest is detected, then a high resolution camera is activated which takes better quality image, which is sent to the base-station for further processing. This idea is very good, and is a model for future work that can be done in WCSNs. A few drawbacks are that, first because of the absence of a power budget analysis, it is difficult to identify its operational capacity. The knowledge of operational capacity is necessary to identify the probable application for the platform. Since, this has one camera remaining on for a considerable time, it may not keep the node alive for a long time. Also, the hardware requirements would not be cost-effective for a large-scale deployment. The presence of also a bulky structure may also make it difficult for random deployment.

In a timer based schedule, more energy can be saved by putting the node into deep-sleep or long term deep-sleep modes. The waking up in such mechanisms is done by a supervisor Real Time Clock (RTC), which is implemented using an independent oscillator powered separately by the batteries. This allows the entire sensor board to be in sleep. Using such mechanisms [13] have provided very long term operations for sensor nodes, extending as long as 552 days. This type of scheduling can be employed for applications like environmental data gathering, where there is no need of an event-based scheme and the overall the application is delay tolerant. Delay tolerance is relevant since, much more time is needed by the processor to come out of deep-sleep as compared to any other power saving mode.

There are also applications where sleep-based scheduling would end up using more resource as compared to a continuous awake strategy. Example for crowd monitoring, there will be more energy consumed, in putting the node into sleep and then waking it up. Also, the delay will result in some of the events being missed. The better approach for this application would be to keep the camera active all the time and use some other power-saving strategies to conserve energy [14].

As seen from the previous examples the *selection* of the mode of operation for sensor networks mostly depends on application. Once the mode has been decided, the *quantity* of energy saved depends on the consumption characteristics of the hardware along with the transitional characteristics of the selected mode. There have not been many works that have performed judicious power-budget analysis of the hardware and the operation mode. ExScal has been one of the biggest real sensor node implementations. This has been on primitive sensors and developed for the purpose of surveillance. A lifetime analysis of ExScal has been performed for an always-on sensor network (where at least one sensor needs to be awake for monitoring and not a complete deep-sleep operation) application. The analysis is thorough and perfect for the specific application, but its main limitation is that since it has been implemented on traditional scalar sensors it has assumptions which cannot be extended to multimedia sensors. For example, the active period for the sensors has been considered to be 10 seconds, as the target is expected to be in the sensing range for that time. This cannot be assumed for say camera sensors, where this value may change with respect to the field of view of the camera, its position with respect to the target. Also, even if the sensing period is low, WMSNs take maximum time in *transmitting* the captured data to the next hop. Similarly other assumptions for packet loss, packet sizes are not consistent with those of WMSNs.

D. Jung et al [15] have presented models for the two modes of operation i.e. trigger (event) driven and duty cycle (schedule) driven modes. The models are constructed using Semi-Markov models for considering the power-transitions. The proposed models use a set of hardware parameters such as power consumption per task, state transition overheads and communication costs to compute the average lifetime for a given event-arrival rate. The advantage of these models is that they consider both the hardware used and the mode of operation, hence they can be used for deployment analysis or as a tool to determine the best mode of operations for a given application and hardware.

In this work, an attempt has been made to implement an event-driven mechanism on the commercially available Imote2 sensor node and the IMB400 multimedia board. The IMB400 has an on-board PIR motion sensor which can be used to generate a trigger to wake the Imote2 up from a sleep mode. The software, TinyOS-2.x and NesC, also have specific properties that can be very suitable for this application. A number of different hardware and software parameters need to be known and implemented correctly, and through this work, a framework has been provided for implementing a correct event-triggered mechanism on the camera sensor. The consumption analysis of the camera sensors has been simulated to evaluate the role of transmission in the total consumption and also to prove the effectiveness of event triggering for a surveillance application using lifetime analysis methods proposed by [15].

2.3 Salient features of hardware and software

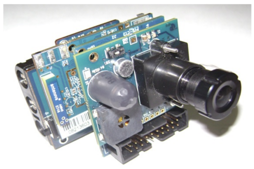


Figure 2.1: Imote2 camera sensor mote

2.3.1 Hardware

IPR2400 Processor/Radio board

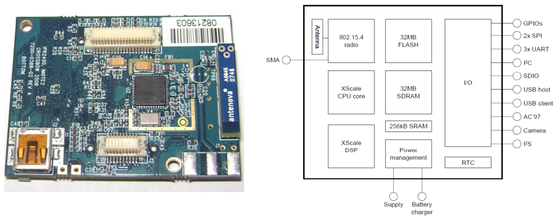


Figure 2.2: IPR2400 board

The IPR2400 mainly comprises of the **Intel Xscale PXA271 processor board** and the **CC2420 radio chip**.

The PXA271 is a low power platform with the lowest active power can be obtained by operating the processor at the lowest supported voltage (0.9V) and current of 30 mA. It supports 6 power modes. It has a 32 bit processor with 256kB SRAM, 32MB FLASH and 32MB SDRAM. This amount of memory helps multimedia operations along with a built in DSP co-processor which supports, a high performance and low power multimedia operations.

The CC2420 radio chip integrates the 802.15.4 Radio with a built in 2.4 GHz antenna that supports 250 kb/s data rate with 16 channels. A typical range of 100 feet (30 m) is achieved. For longer range requirements an external antenna can be used via optimal SMA connector.

Power Source

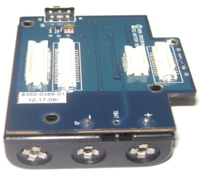


Figure 2.3: IIB2400 battery board

The IIB2400 battery board can hold 3 AAA batteries, with a combined energy of 18360 J. All the other boards are mounted on this board. There is a Power Management IC (PMIC) , which is a DA9030 chip on the IPR2400 board, which actually manages the power distribution. As will be seen later the DA 9030 along with the PXA271 have a number of power saving functionalities that can be used together to achieve very low power operations.

IMB400 sensor board

Figure 2.4: IMB400 multimedia board

The multimedia board has on-board, an Omnivision OV7670 camera, Panasonic Passive Infrared Sensor (PIR) and a miniature microphone and speaker. The camera has resolution 640x480 pixels with video capture capabilities. The PIR sensor is Panasonic sensor with a detection range of upto 5 m and detection angle of 80-100 degrees.

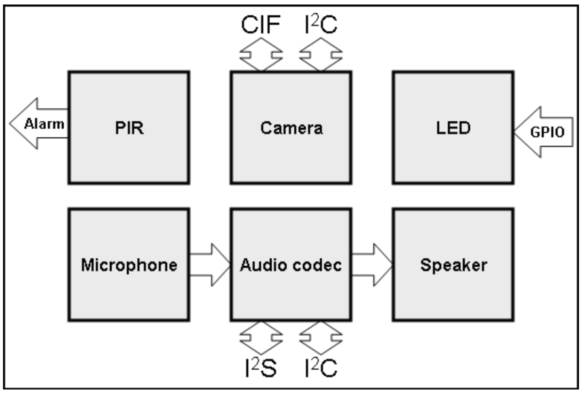


Figure 2.5: Block Diagram of the IMB400 sensor board

2.3.2 Software

The embedded OS that is used on Imote2 is TinyOS-2.x, which is programmed using NesC, a dialect of C. Both TinyOS and NesC are specifically made for WSNs. Some of the key features that make them very effective for WSNs are as follows. TinyOS has a very small memory footprint, with 400 bytes of code and data combined, making it ideal for the resource constrained sensor nodes. It has a *component* based architecture, where the components are re-usable software routines. Most of these components are software modules, while some are also wrappers around hardware. Each application can be *wired* to the component, customizing it to its requirement. One of the main features considered in the programming is the Event-driven nature. The idea here is that the motes should be fundamentally event-driven, i.e. unlike regular computing systems which are mainly interactive, the WSN software should react to the changes in the environment, for example message arrival, sensor acquisition etc. TinyOS was built to satisfy these critical requirements, and nesC was developed to implement TinyOS hence it also satisfies these requirements. Thus, along with the hardware available, the software provides the added impetus necessary for implementing an efficient event-driven wake-up.

A few disadvantages of these software include; it is not very easy to program and use. Owing to such component based architecture, it has very less library support for image processing functionalities, making on-board image processing programming very difficult.

2.4 Working

2.4.1 Role of software

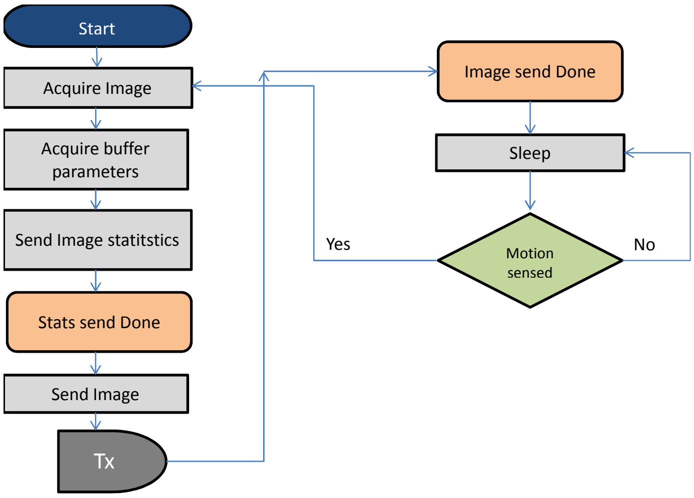


Figure 2.6: Flowchart for camera sensor operation

The image capturing process can be explained as a sequence of processes as shown in Fig.(2.6). The code consists of the *camera module* and the *payload transmission module* (also called as the big message sending module).

When image capturing initiates, i.e. the sensor node is turned on, the camera module is called first. The important task in this module is the *acquire* task, which acquires the image using the camera as per the specifications like resolution, format etc. The acquired image gets buffered in the memory. The next important task is to acquire the image parameters from the buffer, namely the image size and the starting address location. This information is passed on to the payload transmission module. The transmission is initiated by sending the image statistics such as the dimensions and size of the payload. This is done using the *send* interface. In the code this is given by,

```
call ImgStatSend.send(0, &img_stat_msg, sizeof(img_stat_t));
```

Listing 2.1: Sending image stats

In the code, the first input argument is the *next hop address*, the second the packet to be sent, and the third the length of the packet. If the header is sent successfully, then the component will signal the sendDone event in the future. This event triggers the call for transmission of the payload. This can be seen in the next part of the code, shown in Lis.(2.2). When the payload transmission is called, the inputs are passed on to another module called SendBigMsgM.

```
1  event void ImgStatSend.sendDone(message_t* bufPtr, error_t error)
2  {
3      call SendBigMsg.send(sendAddr,sendSize);
4  }
5  event void SendBigMsg.sendDone(error_t success)
6  {
7      call Leds.led0Off();
8      call Leds.led2On();
9      post sleepTask();
10 }
```

Listing 2.2: Sending image payload

The main task performed in this module is the *packetization* of the image data. Packetized data is advantageous for many reasons, but the simplest being that if lost, in case of packet data, only some information is lost not all. Each packet consists of a header and payload. The payload carries the image information. Each payload is of 64 bytes. The transmission module keeps track of the number of packets transmitted. When all the packets are transmitted, a signal indicating that the transmission is completed is generated. This signal is returned to the camera module, as the sendDone event. It is this event that can be used to trigger the sleep interface, as can be seen on line 9 of Lis.(2.2). This is an example of the event-based operation of TinyOS and nesC.

A McuSleep interface is available, which is wired to the McuSleepC component. This component is capable of generating an interrupt,which can put the mote in a desired sleep state.

Thus, the operational capabilities of both TinyOS and nesC, such as a component based architecture, event-based task completion etc, are utilized to the fullest in implementing an event-triggered wake-up procedure in an efficient manner.

2.4.2 Role of hardware

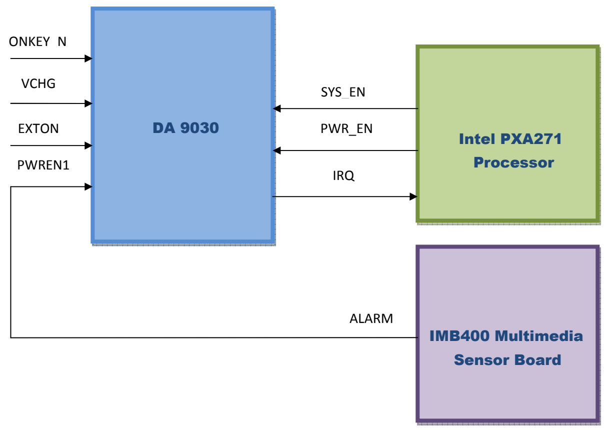


Figure 2.7: Operational Diagram of the sensor board

The PXA271 processor and the DA9030 Power Management IC (PMIC) together offer several power-save modes. The DA9030 PMIC was developed in close cooperation with Intel to achieve optimized power management for the mobile handset units that use the Intel communications processor along with the PXA27x application processor series. Because of this, the terminologies used will bear close association with the mobile handset applications.

The PXA271 provides the following power modes: run, turbo,idle,standby,sleep and deep-sleep. The **Idle** mode is the first level of reduced power consumption, in which the CPU clock is stopped. The DA9030 does not take any part in this mode. The **Sleep** mode significantly reduces the power consumption as the internal processor is not clocked and therefore not preserved either. Only the real time clock(RTC) and the power manager are clocked, resulting in significant reduction in consumption. The

Stand-by mode is similar to the sleep mode, just that the CPU state is preserved, the activity inside the processor stops. RTC and OS timer are optional. The **Deep-sleep** mode is one where all power domains except the VCC battery can be powered down.

The PMIC on the other hand provides assistance mainly for the two modes, sleep and deep-sleep. Entering into the power-save mode is initiated by the processor, during which the PMIC needs to be in the active state. The **PWREN** is an active high input from the PXA271 to the DA9030. When de-asserted, the PMIC is informed that the processor is entering the sleep mode and all the **low** voltage power supplies are to be shut down, which includes the Core, SRAM and the PLL. The other input is **SYSEN**, which when de-asserted disables the **high** voltage supplies that includes the I/O, LCD, memory, USIM and the USB.

As explained previously, the McuSleep is the interface which is called to put the mote to sleep and internally it acts as wrapper around these hardware and initiates the signals.

2.4.3 Implementing the Wake-up

As the power manager, the DA9030 is responsible for handling the switch-on or wake-up process of the processor. There are four different ways in which this can be done. The most obvious way is the ONKEY. It is also powered if the external adapter is detected. An external peripheral device can also wake-up the device by generating a input high signal. The important one here is the **ALARM** baseband signal generated at the PWREN1 of the PMIC. The alarm signal gets its name from the mobile handset application. When an alarm is set on a mobile handset it is triggered, irrespective of which power-saving state it is in. The same idea is used in this case. The PIR sensor on the IMB400 multimedia board is connected to this ALARM pin, as shown in Fig.(2.7). As a result whenever an event is detected by the PIR motion sensor, it generates a baseband signal resulting an *event based wake-up*.

2.5 Camera Sensor Lifetime Analysis

Event driven operation of the WCSNs is justified only if a significant improvement can be observed over the other operational mode, i.e. a schedule driven operation. The consumption of sensor network system depends on a number of factors such as consumption per task, state transition overhead, communication costs etc. Thus, the mode which consumes less energy for a given application, which largely depends on the event arrival rate, should be selected for operation. The idea here is to perform the lifetime analysis of the available hardware using the model-based design proposed by D.Jung et al [15]. Through the results of this analysis it can be determined whether this or a set of similar hardware can be useful for the proposed surveillance application.

As discussed earlier the available system is well suited for event based operation with the availability of an on-board PIR sensor. But even a schedule based scheme can be well supported on the Imote2, with the processor having an in-built RTC. The PMIC too has an internal RTC which remains on all the time and sinks very little current. Thus, the available hardware also has good support for a schedule based operation.

With a capable hardware for implementing both type of models, the main variable parameter which makes a difference is the *event arrival rate*. The event detection probability will thus depend on the arrival rate and the model used. For an event-driven model, the event sensor is always on, keeping the event detection probability very high, almost one. But this requires the sensor to be always on and also the event arrival rate will be crucial, as putting the node to sleep and then waking it frequently might expend more energy. Thus, the power saving mode is efficient only if time spent in that mode is greater than a certain threshold. This event rate will largely depend on the application. For example, if surveillance has to be done over a rough and inaccessible terrain the expected activity will also be low and sparse. In applications with medium and high activity, schedule based strategy might be better.

The first analysis is a life-time analysis of the given hardware, taking the power consumption characteristics for each operation, event-arrival rate (for event driven), duty cycle (for schedule driven) as inputs. This will indicate which mode of operation will serve better for a surveillance application.

The second important parameter that needs to be evaluated through simulation is the achievable lifetime for different power-states for a fixed event-arrival rate. This analysis predicts the duration for which the given set of hardware can operate.

The third analysis is simple but significant, which analyzes the role of the data size in the total power consumption. Traditionally, the power spent in transmission has been considered insignificant as compared to the one expended in monitoring especially in case of scalar sensors. The main reason being that the time for monitoring is very long as compared to the transmission time. But in case of camera sensors, the data transmitted per event is large, resulting in more consumption in transmission. This has been simulated in the third analysis.

2.5.1 Event driven vs Schedule driven

In Fig.(2.8) the lifetime plot for a very long inter arrival rate has been shown. The experience activity frequency is 1 event per hour to 1 event per day. This is quite sparse though a typical monitoring application may experience even sparser activities. In the plot it can be seen that the event-driven scheme is the clear choice. For some values the lifetime of schedule driven comes close to that of the event-driven, but in that case the detection probability for the schedule driven operation is very low. Thus, it can be concluded that for high inter-arrival times the event-driven mechanism is the best choice.

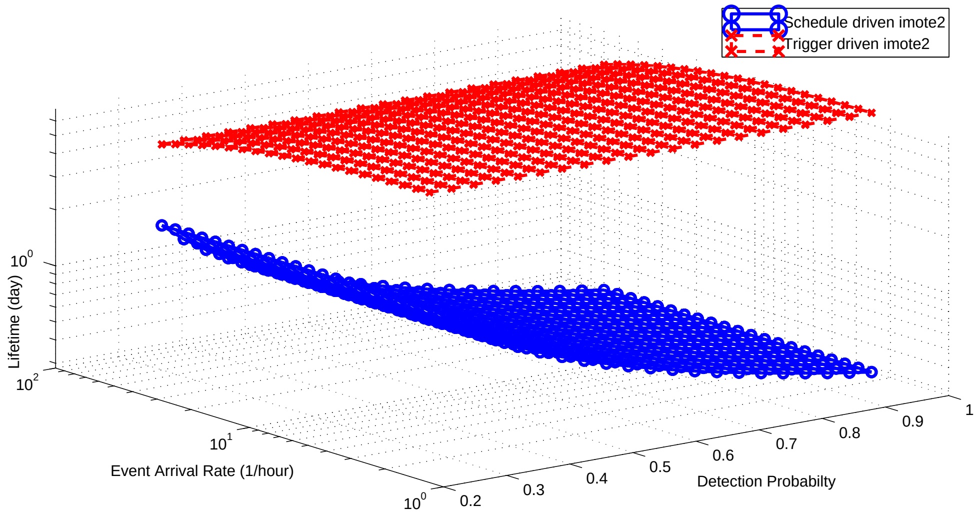


Figure 2.8: Lifetime analysis plot for large event interval

This may not be the case with a lesser event intervals, which can be observed in the analysis shown in Fig.(2.9) Here the expected range of events is between an event per minute and to one per hour. In this case there is a trade-off on the models that can be employed. It can be seen that for event arrival values less than 10^2 events/hour, which is equivalent to an event every second, the schedule based scheme has a better lifetime performance. Such small event-intervals are unlikely in a surveillance application, but may occur in some other applications.

Thus, lifetime analysis plots can be used for purposes such as selecting the models of operation for a particular application, test the operational capability of new hardware designs, predict the accuracy of the system for detection for different arrival rates, especially for schedule based schemes.

2.5.2 Achievable lifetimes on the Imote2 hardware platform

As mentioned previously, together with the PMIC and the PXA271 a number of power-saving modes can be implemented for the Imote2. But each power-saving mode has a peculiar characteristic which can affect the ability of wake-up. The most

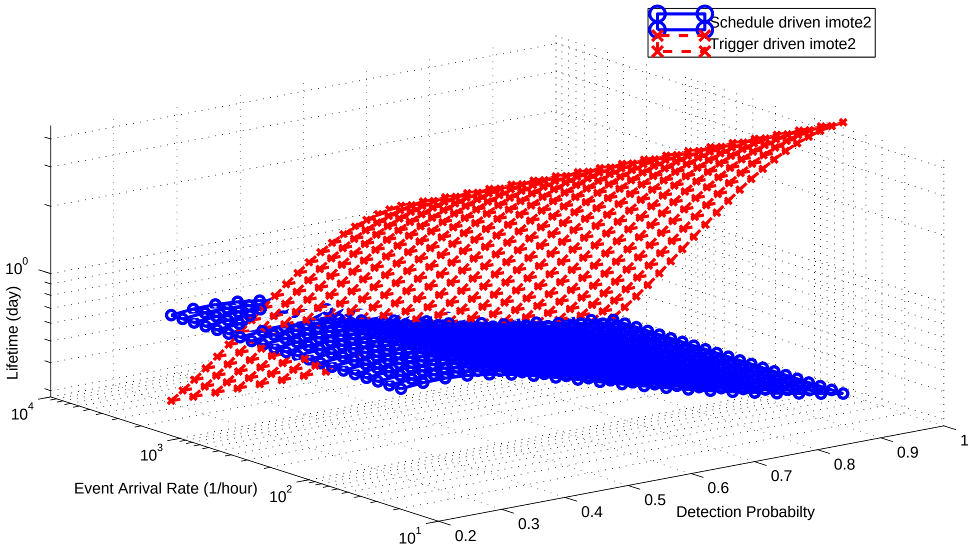


Figure 2.9: Lifetime analysis plot for small event interval

important one here is the Deep-sleep. In this mode, all the high voltage system power supplies are also turned off. As a result the system cannot be triggered using an internal on-board sensor, like the PIR. The Sleep, Stand-by modes are possible to wake-up. A good indicator to observe the benefits of sleep operations is to compare the lifetime with respect to an Idle mode sensing. The event-arrival rate considered is 20 events per day.

At this rate, the Idle mode can support the Imote2 for 2 days. For the same in the Stand-by mode the mote will last for 11.22 days and for the Sleep mode it will be 15.22 days. This shows the effectiveness of the power-saving modes in prolonging the lifetime. Previous projects on scalar sensors had discussed and proved the role of such mechanisms, but an analysis for this set of hardware was not available.

2.5.3 Effect of data-size on the consumption characteristics

As mentioned previously the data-size could be an important factor in the consumption. This can be seen in the graph shown in Fig.(2.10)

With the increase in the number of packets, the prominence of monitoring power

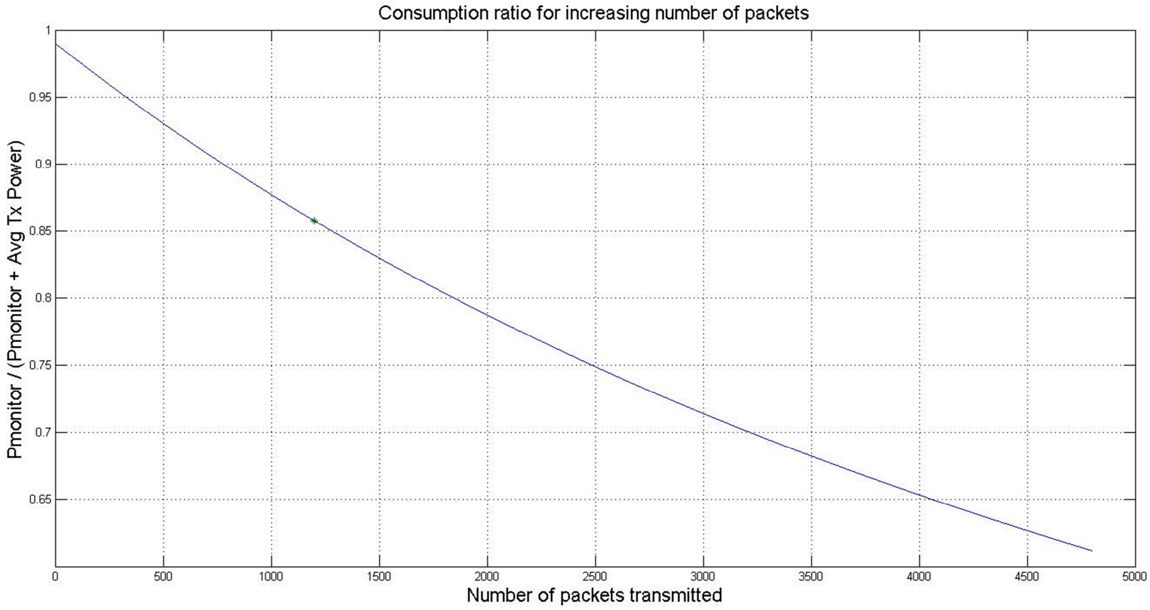


Figure 2.10: Decreasing influence of monitoring on the total consumption with increasing packets

in the total power consumption has decreased, indicating the increased role of transmission.

This fact can also be proven through lifetimes for variable image sizes. At 20 events per day, in the Sleep state, the lifetime when capturing and transmitting a QVGA image is 15.22 days. The same combination will only last for 11.08 days for a VGA image.

These analysis provide enough evidence that in WCSNs both monitoring and transmission are equally important in terms of consumption. In this section it is shown how to implement an effective scheme for saving up on monitoring. In the next section, a scheme for an energy efficient transmission will be discussed. In traditional transmission schemes, overhead in terms of parity bits are used to enable error correction or loss recovery. This excess overhead can be avoided by using a new technique signal sampling called Compressed Sampling. The next section discusses its relevance, theory, implementation and the results.

CHAPTER 3

Compressive Sampling for Energy-Efficient and Loss-Resilient Camera Sensor Networks

3.1 Relevance of CS

As mentioned earlier WCSNs are embedded systems capable of capturing and transmitting snapshot or streaming multimedia over a multi-hop network, making them suitable for surveillance and monitoring; application which also demands longevity of operation while maintaining a certain degree of quality of service.

Different energy consumption models [16],[17] show that the *communication* process takes much larger power as compared to the processing and sensing units of the WSNs. In the previous section, an attempt to conserve energy was made on the sensor node level, which reduces the idle sensing period of the sentry nodes, but the Tx/Rx, where the maximum energy gets consumed needs to be more efficient. Thus, efficient transmission schemes can help in operating WCSNs for long durations.

Data loss has been a notorious problem in the wireless domain and it gets aggravated in sensor networks because of the harsh environments and a limited power source. Excessive losses can result in most of the image being lost, even making a complete transmission useless. This results in wastage of the already energy constrained devices. Forward Error Correction (FEC) and Automatic Repeat Request (ARQ) schemes are available to combat erasures, but they are complex and with high packet overheads making them not very energy efficient schemes.

Thus an ideal requirement in WCSNs is a scheme which is loss tolerant and energy efficient at the same time. CS turns out to be an effective solution to both these problems and also suitable for implementation in the sensor networks domain. The theory of Compressive Sampling enables us to generate samples of a sparse signal through linear random projections such that the compressed signal is smaller than the original, reducing the number of packets that need to be transmitted. This signal can be recovered using the fact that the L1 minimization of a CS signal provides the sparsest solution. This helps to mitigate the challenge of reducing the transmission costs and because of the inherent randomness in the sampling stage, random erasures make little difference to the overall signal statistics making the CS data resilient to losses.

Thus CS proves to be a valuable asset for WCSNs, but using CS in WCSNs has some practical difficulties. WCSNs are being developed with prospective applications in surveillance and monitoring. This requires the cameras to capture images over a wide field of view, which also requires large images to be processed. It is known that the recovery algorithms of CS data such as the Basis Pursuit have a complexity of $O(N^3)$, where N is the length of the data. This makes the large images computationally intensive with long recovery times. These factors hinder the use of CS in practical applications and hence need to be minimized.

As a part of this thesis a CS based solution to the aforementioned problems of WCSNs has been proposed and implemented on the Imote2 sensor node. This includes using CS parameters suitable for sensor node implementation. The implementation of CS has been validated through extensive testing over an actual sensor node test-bed. Another important contribution has been in making CS computationally more feasible, by modifying the basic scheme to block-based schemes and extremely sparse binary matrix, in order to improve the decrease time and reduce the processor memory requirements.

3.2 Theoretical Background

Data sampling has been governed by the Nyquist theorem, which gave a formulation of sampling a given signal at a rate at least twice its frequency in order to recover the samples correctly. This conventional belief is a *sufficient* condition, and not a necessary onw. This is because, the Nyquist theorem does not consider the properties of the signals being sampled. There are a certain class of natural signals, which can be recovered with high probability, from samples obtained at much lesser rate than given by the Nyquist theorem. Compressed Sampling (aka Compressive Sensing) deals with this newly developed sampling theory. The theory of Compressed Sampling gives a detailed mathematical review of the signals on which such type of *under-sampling* can be performed, the methods of sampling and their recovery.

Compressed Sampling (CS) derives its name from the fact that even in case of conventional sampling schemes, the data is first sampled at the Nyquist rate, important samples are retained and the less important samples are discarded. This results in most of the sampling resources going unused. The best example for this is the JPEG compression scheme. First the image is captured, then it is transformed in the DCT domain and components are ordered in a descending order. Of these only a few components are kept while most of them are discarded. Thus, sampling resources are not efficiently used in the traditional sampling. But, this does not hurt much for applications like JPEG used in conventional cameras, as the sensors have become very cheap. But in fields like Infrared imaging, sensor costs are very high and if by some method the number of sensors required can be reduced, then it can be economically very beneficial. Another probable application for CS is MRI scanning, where the *time* taken by the complete scan is very long. If the time taken for scanning can be reduced then it will be more convenient for patients and could even help to serve more patients. Thus, CS can find many other applications where conventional compression schemes worked.

The question is how to collect enough data with lesser sensors or in lesser time than usual. The theory of Compressed Sampling attempts to answer these questions. The ideology of CS is very simple , Take only what you need. i.e. sampling only those components that are going to be actually required rather than taking all and then discarding some. The theory of Compressed Sampling tries to answer questions such as, the type of signals on which such sampling can be performed ? How to perform this selective sampling ? How to recover the complete information from the limited samples etc. In the next two parts, the concepts of CS sampling and recovery have been discussed in detail.

3.2.1 Sampling

The first question was the type of signals on which CS works. As mentioned earlier there a certain class of natural signals on which the CS strategy works the best. These signals are called *sparse signals*. A signal is called sparse if the most of it components are zero or of a very small value, which can be approximated to zero. Actually very few natural signals are sparse in their natural form, but signals do exhibit sparsity in their transform domains. For example, images are sparse in the DCT or Wavelet domain, sound signals are known to be sparse in the Fourier domain. Thus, CS works on sparse signals or signals that can be expressed in a sparse form in some transform domain. The reason behind this would be clear in the recovery section, but since sparsity in a transform domain is allowed, most of the signals can be operated under the purview of CS.

The problem as to how to sample the data such that only the important samples are picked during sampling appears to be more complicated, because determining which samples are important even before they are observed seems to be paradoxical. Using some prior knowledge will not always be possible, as the information to be captured is mostly random in nature. For example, in an image it is impossible to decide before hand which part is important which is not, mainly because for every image the important part is different and random.

But ironically the solution to this problem has been found to be Random sampling. Random sampling is a process of generating *linear random projections*. It can be given as random selection or random weighting and addition of the signal elements to generate a set of measurements. This type of sampling ensures a measurement contains information from multiple signal components, and thus *less* measurements inherently carry *more* signal information, resulting in a *Compressed Sample*. Random sampling just ensures that a a set of measurement has information from all parts of the signal, making each measurement statistically same.

Mathematically, sampling is given by

$$\underline{y} = \Phi \underline{x} \tag{3.1}$$

, where $\underline{x}$ is a *sparse signal* of length n. $\underline{y}$ is the set of *measurements* of length m, $(m << n)$. Φ is the *sampling matrix* of size m x n.

The important conditions that need to be satisfied by the sampling matrix is the Restricted Isometric Property (RIP), which has been discussed in [18]. Different types of random sampling matrices have been found to satisfy this property and work well as sampling matrices, like the Gaussian matrix, Vandermonde matrix, Scrambled Fourier matrix and Binary Sparse matrices.

3.2.2 Recovery

After compressively sampling the data, the question is how to recover the original signal from the measurements. As can be seen from Eq(3.1), signal x_n has to be recovered from the measurements y_m, where $m << n$. This is an ill-posed problem resulting in an under-determined linear system with infinitely many solutions. It is the solution to this problem that has given a huge impetus to the work in CS

[19],[18],[20],[21]. The solution of the problem states that the *l1*-norm minimization of the Eq(3.1), results in the sparsest solution.

$$\underline{\hat{x}} = \arg\min \parallel \underline{x} \parallel_{l_1} \ s.t. \qquad \underline{y} = \Phi \underline{x}. \tag{3.2}$$

As seen in Eq(3.1) and Eq(3.2), the signal $\underline{\mathrm{x}}$ should be sparse and as discussed previously most natural signals would not be sparse in the original (canonical) basis and they have to be transformed. Now, when sensing a physical signal, i.e. a process when it is being acquired, there is no actual signal available that can be transformed, rather it is being acquired.

It is here that the Basis Pursuit(l_1 minimization) provides an advantage. Suppose the transform domain of sparsity of the signal is known, i.e. $\underline{x} = \Psi * \underline{s}$, where Ψ here is the matrix corresponding to the ortho-normal basis. Then at the recovery step, the signal to be recovered can be written as, $\underline{y} = \Phi\Psi\underline{s}$, as shown in Fig(3.1). As stated earlier, the l_1 minimization produces the sparsest solution to the problem, which is the $\underline{s}$. The recovery equation when sampling has been performed in the dense domain is given by,

$$\underline{\hat{s}} = \arg\min \parallel \underline{s} \parallel_{l_1} \ s.t. \qquad \underline{y} = \Phi\Psi \underline{s}. \tag{3.3}$$

After finding $\underline{\hat{s}}$, the original signal can be obtained by simply using an inverse transform operator on $\underline{\hat{s}}$, i.e. $\underline{\hat{x}} = \Psi\underline{\hat{s}}$. This gives the freedom of sampling the signal in its original/dense domain as in Eq(3.1), and still perform successful recovery. It has been proven that for a sparse enough signal, basis pursuit provides exact recovery if the measurements are taken in the order of $klogn$, where k is the sparsity of the signal.

But there is one important condition that must be satisfied by a pair of sampling matrix (Φ) and the ortho-normal basis (Ψ). It is called the *mutual coherence*. It is a representation of the idea that signals that have a sparse representation in Ψ must spread out in the domain which they are acquired in. Mutual coherence between Phi

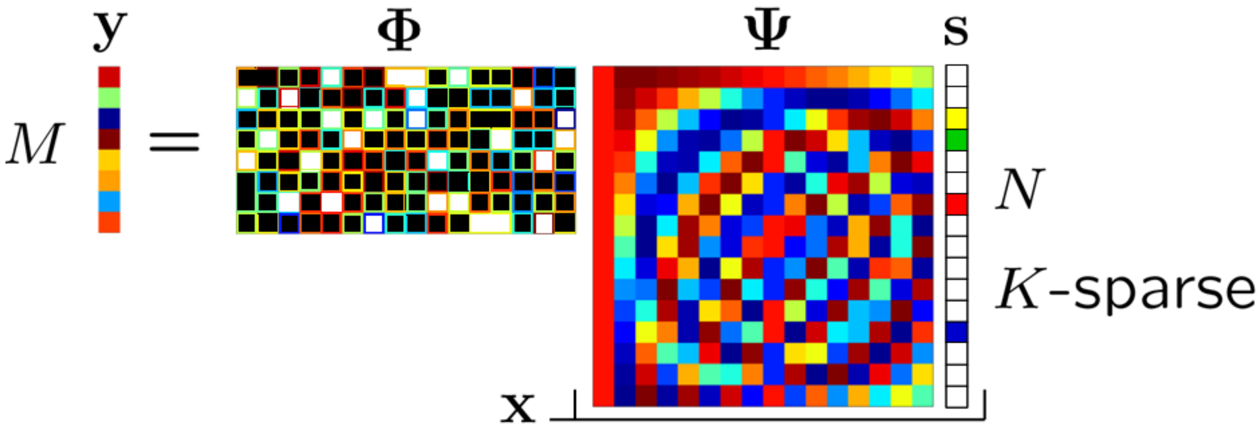


Figure 3.1: CS sampling process with sampling matrix Φ and Orthonormal matrix Ψ

and Psi ,

$$\mu(\Phi, \Psi) = \sqrt{n}_{1 \leq i \leq m, 1 \leq j \leq n} max \ |< \Phi_i, \Psi_j >|, \tag{3.4}$$

where Φ_i's are rows of Φ and Ψ_j's are columns of Ψ. It is the measure of the correlation between the rows of Φ and the columns of Ψ. The requirement for a good CS recovery is that the pairs of Φ and Ψ should have a *low coherence* or *high incoherence* [21]. The good thing about random sampling matrices are that most of them are found to be highly incoherent with most of the standard ortho-normal basis. Mutual coherence plays a very important role in selecting the CS parameters in this project and it will be discussed in the later sections.

Basis Pursuit or l_1 minimization is the most popular and holistic approaches for CS recovery as it operates on a wide variety of problems. One of the main drawbacks of this approach though is, since it is an *optimization* based approach, it takes a long recovery time and huge computational resource. Some other less intensive methods have been developed which are computationally less expensive. Prominent among them are *approximation* based approaches, where an iterative approximation based scheme is used to recover the signal. Popular algorithms in this category include Matching pursuit based approaches such as Orthogonal Matching Pursuit (OMP)

[22] and Compressive Sampling Matching Pursuit (CoSaMP) [23]. Methods such as Iterative Hard Thresholding have also been known to work for CS recovery [24]. The limitation of these approximation based methods is that they provide local minimums as compared to global minimization provided by the optimization approaches, hence the recovered signal quality may not be as good as compared to the optimization based approaches.

3.3 CS for WCSNs

After the theoretical background of the CS mechanism it will now be clear that how CS can be an effective tool in WCSNs.

3.3.1 Channel Erasure

One of the main advantages is that CS makes the data resilient to erasures. When an image is routed over an ad-hoc network, it is sent in form of packets, where a specific number of pixels are put into one packet and this packet is transmitted over a network. However, a transmission process is never perfect and a number of packets can get lost in the process of transmission. This loss of packets in a communication channel is called an *erasure*.

A channel is can be termed as an *erasure channel* if the a packet is either received perfectly or lost. Usually a channel is an AWGN channel (Additive White Gaussian Noise); however if packets with some error are dropped at the receiver, then the AWGN channel can be modeled as an erasure channel. In the available hardware platform the radio chip used, CC2420 has a CRC (Cyclic Redundancy Check) field. If the CRC fails, the packet is marked and the TinyOS module discards that packet. Thus, even if bit errors are present, the channel gets modeled as a Packet Erasure Channel.

Analyzing the performance of the WCSN system in presence of erasure is important because the *packet delivery performance* of the sensor network system is an important parameter. The delivery performance of WSNs varies with the environment it operates in. Zhao and Govindan [25] have tried to asses the performance of WSNs in various operating environments. They have attributed the delivery performance to different factors in two communication layers, namely the physical layer and the Medium Access Control (MAC) layer. In the physical layer the delivery performance is affected by the environmental characteristics such as multi-path propagation and signal attenuation. Multi-path can be a more prominent factor in a dense environments and a possibility of a dense surrounding in a WCSN application cannot be ruled out. The MAC layer deals with the arbitration of the channel for access, thus the traffic on the channel becomes critical. In case of WCSNs because of the bulky nature of data, transmission of a single image involves transmission of large number of packets, which proportionately increases the traffic on the channel and is one of the main factors which affects the packet delivery performance. The topology or the spatial relationship between the nodes also affects the number of nodes that might potentially contend for the channel at a given point of time which in turn affects the delivery performance of the sensor network.

A poor packet delivery performance in an image can lead to major portions of image not being received correctly, resulting in loss of information. This problem can be formulated as a CS problem. Suppose, $\underline{x}$ of size n (signal) is the image captured by the camera sensor node that has to be transmitted to the base-station. This signal is subjected to compressed sampling, i.e. linear random projections of the signal are obtained using the sampling matrix, to form a set of measurements $\underline{y}$ of size m , which are transmitted over a multi-hop network. Suppose t packets are lost, since it is an erasure channel, *(m-t)* packets are received correctly. The problem is to *recover n packets from m-t packets*. The only variation of this problem from the original CS problem eq(3.3 and 3.4) is that the Φ will be different at the transmitter and receiver, because of the packet loss. If the packet loss is accounted for while reconstructing the sampling matrix at the receiver (call it Φ'), then the signal can be recovered using basis pursuit. Note that Φ' is a sub-matrix of Φ.

Random sampling plays an important role in making this transmitted data *loss resilient*. As describer earlier the packet transmitted consists of CS measurements. Since each measurement is generated randomly, each packet carries statistically the same information. This makes the system immune to individual packet loss. Thus, we can reconstruct the signal as long as we receive enough number of packets, independent of which packets have been received.

3.3.2 Energy Efficient Transmission

As discussed earlier, CS provides the ability to recover complete information from far lesser number of packets than the ones required by a regular image. Continuing the previous analogy, if an image originally requires n packets and using CS if almost the same information can be recovered by transmitting only m packets, where $m << n$, then there is definite saving in the number of transmissions per image. Power required for transmission is very high as compared to the power required for processing. Especially, the Imote2 as seen in chapter 2, is specifically made for low power processing and equipped with a special co-processor which can compute mathematical instructions faster. This is better than the traditional error correction schemes, which use parity bits for error control which is more costlier because of the extra transmissions. The ARQ schemes where packet is re-transmitted does not work for WSNs at all, as it consumes even more energy.

Thus, Compressed Sampling is a powerful tool for transmitting images over WCSNs as it is a mechanism that ensures resilience to erasure and still proves efficient in terms of transmission. The challenge that remains is to implement a CS algorithm on an actual sensor node hardware. There are a number of CS parameters that need to be considered for selecting the best approach. The different parameters required for CS and their selection criteria have been discussed in the following sections.

3.4 Related Work

As mentioned before the inherent randomness and signal compression can solve both the limited power and data loss problems simultaneously, making CS an ideal choice for WSNs. Performance of CS for erasure coding has been discussed in [26]. In this work the authors propose a *compressive oversampling* approach to compensate for the expected erasure to maintain a target signal quality. This work has been extended in [27], where the oversampling is performed for expected loss due to bit-errors too, along with channel erasure, leading to some improvement in the reconstruction quality. The premise of this Oversampling approach is that the target signal quality is known. The problem here is that in a practical WCSN application, determining a target quality is not possible, because it is a no-reference system. Also, a standard compression ratio at the transmitter cannot be assumed because through erasure analysis we found that the number of samples required for a given reconstruction quality varies with the type of image captured. Thus, the compression at the camera will depend on the image being captured and hence will vary as per application.

Conventional CS schemes suggest a dense random projection matrix for signal sampling. However, it was shown later that binary and sparse random matrices have a good performance as well and are very convenient for implementation. [28] and [29] have a very good analysis on binary sparse random matrices and discuss the special properties of binary sparse projection matrices, especially the impact of the sparsity of the matrix (low column weight in their case), in the reconstruction as well as the recovery time. The findings in these studies can be related to our work for theoretical analysis. [30] proposes a *block compressed sensing* approach for improving the recovery time and memory storage. The approach in [30] involves using a sampling matrix similar to an FIR filter to generate CS measurements from the traversed portion of the image. For reconstruction, a minimum mean squared error estimation is used to obtain an initial linear estimation and then use some non-linear techniques for refinement. Using this approach in sensor networks is difficult because of the filter type implementation to generate samples, which increases the computation time and cost. Moreover, the performance of the linear estimation in presence of erasures is not known.

As a part of this thesis, solutions to aforementioned problems have been provided along with a CS framework that is suitable for implementation on an actual embedded sensor node. To the best of our knowledge, this work is unique in providing such a framework supported by an implementation on a real sensor node platform.

3.5 Selecting Appropriate CS Parameters

3.5.1 Measurement matrix (Φ)

From the theoretical background it is understood that the Measurement matrix or Sampling matrix is a matrix used for taking random linear projections of the original signal (hence also known as a projection matrix). The main independent condition that a measurement matrix needs to satisfy is the Restricted Isometric Property (RIP)[18]. There are a number of matrices which satisfy this property, some which are dense matrices such as the Gaussian Random matrices, constructed by selecting i.i.d random variables from a Gaussian distribution. Same is the case with the Bernoulli matrix. These matrices have a very good performance but they are costlier to work with on a resource constraint embedded processor. Hence, another class of matrices called Sparse Binary matrices are used.

A very thorough analysis about binary sparse matrices was done by Berinde et al [28], where they found the performance of sparse binary matrices comparable to Dense matrices. It was shown that such matrices satisfy a weaker form of the RIP property called the RIP-p property, where the l_2 norm is replaced by the l_p norm for $p \approx 1$. It was found that binary and sparse matrices provide advantage in terms of efficient update and encoding times, which also speeds-up the decoding. These matrices have recovery errors comparable to that of dense matrices but in much less recovery time.

The compression process of binary matrices comprises of random selection and addition of variables, hence they are easy and fast to compute.

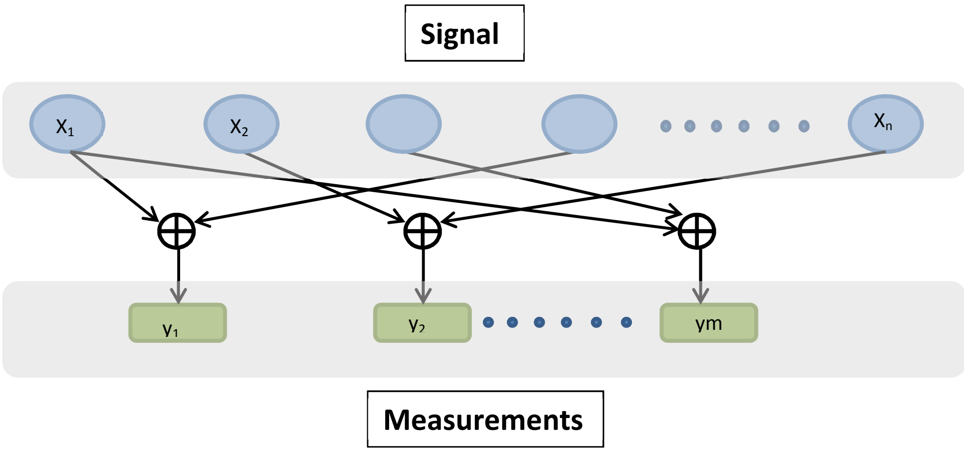


Figure 3.2: Sparse matrix representation of a bipartite graph

The operation of CS sampling using a binary and sparse matrix can be represented as a bipartite graph as shown in Fig.(3.2). It shows the operation of the matrix in the random selection and addition form. The signal co-efficients correspond to the columns of Φ and the measurements correspond to the rows of Φ. The number of signal elements combined to form one measurement is equal to the row/column weight of the sampling matrix, which is indicated by the edges between the signal and measurements. In this implementation a Constant Row Binary Matrix has been used.

There is a specific reason behind using a constant row matrix . For CS sampling implementation, as will be explained in the later section, signal elements are selected randomly and *sequentially*. Hence, having a constant row weight matrix ensures that each selection is made independently, since for each measurement generation there is only a constraint on the *number* of *signal elements* selected. This is not the case with a Constant Column matrix, where there is a constraint on the number of times *each* signal element can be chosen, making the current selection dependent on the previous. This can especially be a problem in case of an erasure channel operation, where because of the mutual dependence, an erasure can result in a wrong reconstruction of the Φ' matrix. Again, it is important to note that the dependence in selection arises because of the sequential selection procedure of the pseudo random generator.

3.5.2 Orthonormal Matrix (Ψ)

As mentioned earlier, the signal can be compressed in its dense domain, just that the domain of sparse transformation should be known at the receiver. But from Eq.(3.3) we can see that the ortho-normal (transform) basis needs to be available in its *matrix* form during the recovery. The problem with this additional constraint is that, even though higher dimensional wavelet transforms provide higher degree of sparsification of images, they are not available in a matrix form and hence cannot be used. The basis that can be represented in a matrix form are the 1D Haar, Discreet Cosine Transform (DCT) and Discreet Fourier Transform (DFT). Note that this constraint is there only because we are sampling in the dense domain. If the signal would have been first sparsified, then this constraint wouldn't have appeared.

Another property that was discussed previously was the Mutual Coherence. As explained before it is a property shared by a pair of orthonormal matrix and measurement matrix. Thus, while selecting an orthonormal matrix, it is pertinent that the use of a Binary Sparse matrix as a measurement matrix is taken into consideration and they should have a low mutual coherence. The Binary Sparse matrix is highly coherent with the 1D Haar matrix hence even though an image has a better sparse representation the Haar wavelet cannot be used. Between DCT and DFT the DCT performs better than DFT because of the better energy compaction. Hence, the DCT matrix is used as the ortho-normal basis.

Mutual Coherence also helps in setting up another parameter, the Row Weight of the sparse matrix. As seen earlier the row weight will determine how many pixels are combined together to generate one sample. Intuitively, it seems like a higher row weight will ensure more original signal elements being combined and hence a better result; but this is not entirely correct. The results with a higher row weight are better, but as the row weight increases, so does the coherence between the DCT and Sparse sampling matrix as shown in Fig.(3.3). Because of these two opposing effects, the

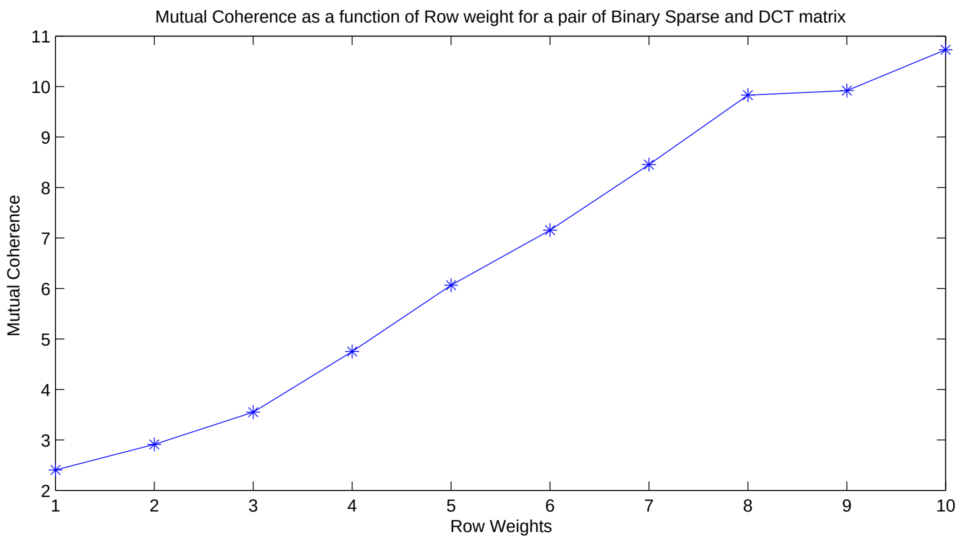


Figure 3.3: Mutual coherence vs row weight of the measurement matrix

increase in quality with row weight is not significant.

3.5.3 Sampling Strategy

As shown in the Fig.(3.2), sampling in case of a sparse binary matrix is performed by random selection and addition of the signal elements. The range over which the selection is made, determines the number of columns of the Φ matrix, which is also equal to the size of the Ψ matrix. From Eq.(3.3) it is clear that Ψ is required in its matrix form for exact recovery of the original signal. The Ψ being used here is a 1D DCT matrix. Suppose an image has a width of w and height h. The DCT matrix Ψ will be of size $w * h \times w * h$. In a double precision format, this matrix requires $w^2 * h^2 * 8$ bits of memory storage. For a QVGA (320x340) size image, the amount of Random Access Memory (RAM) storage would be around 6 Gigabytes, which is a huge requirement as compared to the amount of RAM available in regular computers. This too is only for a QVGA size image, which is the smallest of the standard image sizes used.

The recovery time of the basis pursuit problem holds a non-linear relationship with the vector size it is operating on. It is of the order of $O(N^3)$. As a result if large vector as big as a QVGA image is used, then it may take over 2-3 hours just to process one image.

Thus, the memory requirement and the recovery time are problems which make CS incompetent for practical implementation. Using a 2D DCT can alleviate the problem by some measure, but it puts a constraint of using square images for processing, which may not be always available.

A practical solution to this problem is a subtle change in the sampling strategy. A *block-wise* sampling scheme may solve both the aforementioned problems. The idea is to take independent measurements from each block, thus effectively reducing the vector size upon which the l_1 minimization operates on. As the effective length of the vector reduces, the problems of both memory and recovery time are solved. This also helps to make the mechanism independent of the image size. It is also know that the quality of reconstruction depends on the length of the vector, the larger the vector, better the quality [30]. Hence, there is a trade-off between the image quality and the recovery time. The optimum size can be found through experimentation, as it depends on the application; if an application requires a higher image quality with less constraints on time, then a longer vector can be used and vice-versa.

3.5.4 Implementation of CS for WCSNs

On the camera senor node (transmitter)

The previously explained compressed sampling strategy is implemented on an Imote2 sensor node coupled with an IMB400 multimedia sensor board, which bears the camera. The image used is gray-scale, hence a uint8 (8 bit) data type is sufficient to store a single pixel.Each packet of image data comprises of a header field and a 64 byte (64 pixel) payload.

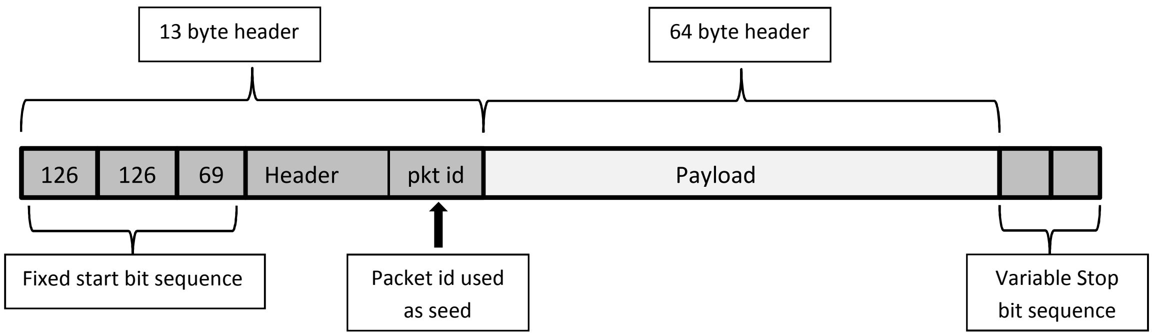


Figure 3.4: Sparse matrix representation of a bipartite graph

As discussed earlier, the sampling process in CS is of generating linear random projections of the original signal data. Since, a binary random matrix is used in this case, it basically has two operations; first is random selection of pixels and second adding together the selected pixels. Random selection is the more complex and important step.

A pseudo random generator (PRG) based on a multiplicative congruential generation (MCG) mechanism [31] is implemented on the sensor node. The default generator was not used because the default generator in TinyOS and the one in MATLAB may not be the same. Since, it is extremely critical to get exactly the same random sequence on the receiver side in order to get the CS reconstruction, a self -defined pseudo random generator is used. An MCG based mechanism was preferred because it is very simple to implement, which makes it suitable for implementation on a embedded platform , but still giving a very good random generation. The implementation of the MCG used is given by ,

$$s_i = as_{i-1} \ \mathbf{mod} \ d,$$
$$\mathit{offset} = s_i \mathbf{mod} \mathit{imagesize} \tag{3.5}$$

The PRG shown in Eq.(3.5) generates random numbers sequentially, i.e. in each step it generates one random number. It is a two step generation process, where a is a multiplier and d is a non-zero modulus. The values of a and d used in the code are 125 and 2796203 respectively. The question is how to use the random generator output to get a sample (here an image pixel). On the camera sensor, the only information available about the image is the starting address in the buffer where the image is stored. As shown in the equation, in the second step of the generator, an *offset* is generated. The range of this offset value is restricted by the maximum selection range (here the image size). This generated offset is added to the start-address of the image. The resulting address is the location of any random pixel of the image. The pixel at this location is selected and added to the measurement packet. The random selections are added to the same measurement packet until it reaches the row-weight.

It is extremely essential to have an ability generate exactly the same random combinations, since the same sampling matrix should be used for sampling and reconstruction Eq.(3.2)and(3.3). To ensure that this is possible the first step is, as explained earlier, to use a self-defined PRG both at the transmitter and the receiver. The second step taken is that the PRG is to initialize the generation sequence by the same *seed value*. To ensure uniqueness in the seed values and reduce an extra payload value, the *packet numbers* are used as seed values.

The exact reason for using the packet numbers will be clear in the next section. At the transmitter, the measurements are generated as described before. Each measurement packet is then packetized and transmitted over the network. Note, that the packet is the unit entity of communication, so in case if the data is lost in channel erasure, then a packet will be lost.

At the base-station

At the base-station the original signal is to be recovered from a set of CS measurements. In order to do that the information needed is, from Eq. (3.3), the measurements, the Ψ matrix and the Φ matrix, which conveys the correct random combination information of the received measurements. As mentioned previously, the same PRG is required to get the same set of random combinations, hence the same PRG is implemented in MATLAB as the one in NesC.

To get the correct combinations, packet numbers are used as seeds. Thus each packet is initialized with a known and new seed . This is done because due to the erasure channel, packets may get lost during transmission and if each packet carries its own seed, it basically carries the combination information about its measurements. Thus a matrix can be constructed as per the received seeds and in case of losses, may be different from the original sampling matrix. Hence, this matrix is called the sampling reconstruction matrix (Φ).

The matrix is constructed using the fact that each measurement received corresponds to a row and each randomly generated number corresponds to a column. Thus, for every packet received, a 1 is put in the column location generated by the PRG. The total number of rows of the matrix will be equal to the number of packets received (m'), making the matrix of size $m' \times n$. With the sampling matrix constructed and knowing the orthonormal matrix, the original signal can be recovered using l_1 minimization as long as m' is large enough.

Block-wise sampling scheme

The sampling process has to be modified on the transmitter in order to perform a block-wise sampling. The only information available about the image array is the starting address where the image is stored. The task is to take CS measurements independently from each block. Segmenting the image vector into an actual block-wise structure would be cumbersome and the process will have a high latency. A faster and simpler way is to modify the random selection Eq.(3.5). If the selection range is changed from the image size to block size and the addressing done with respect to the block starting address then an efficient block-wise sampling can be performed. It can be given by Eq.(3.6) and shown in Fig.(3.5)

$$
\begin{aligned}
s_i &= a s_{i-1} \bmod d, \\
\mathit{offset} &= s_i \bmod \mathit{blocksize}, \\
\mathit{offset} &= \mathit{offset} + \mathit{blockoffset}
\end{aligned}
\tag{3.6}
$$

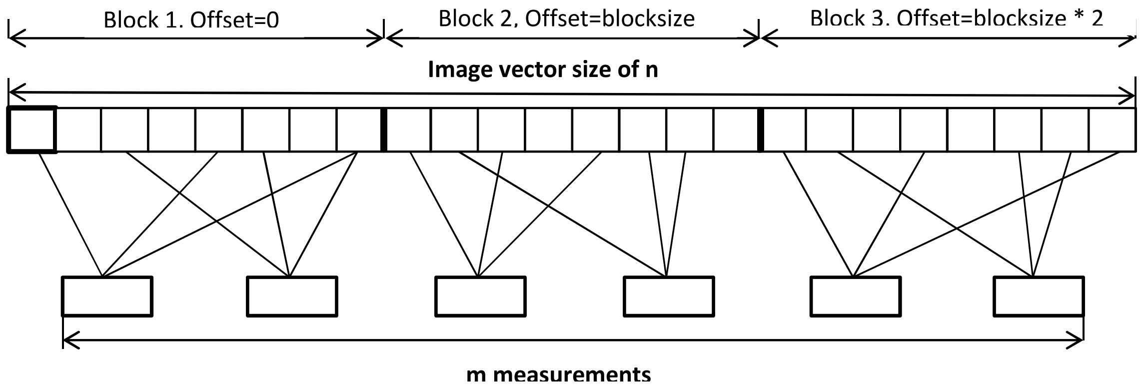


Figure 3.5: Schematic for block-sampling

This procedure ensures that only one block is sampled at a time, keeping the measurements from each block independent. The *block-offset* is incremented by an amount equal to the block size after measurements from one block are taken. This method provides an efficient, low latency mechanism for block-wise sampling by virtue of simple additive computations.

This process can be best explained through the part of the code where this operation in performed, which is the *task* that performs *random sampling* called *randsample*. The process starts with picking a certain packet id, *a*. In regular transmission, data stored in the memory is packetized sequentially, hence *a* or the *part id* increments sequentially. For random sampling this *a*, is generated randomly, as shown in Lis.(3.1), lines 16 and 17. This is the implementation of the MCG pseudo-random generator discussed earlier. This results in a randomly generated buffer offset, shown on line 10. The *memcpy* copies the data pointed by the buffer offset to the buffer. The first *if* loop generates random samples for one complete 64 byte payload *packet*. The *else* statement keeps a check on the block that is being processed. When a complete block is sampled, the block-offset is incremented, resulting in the selection been done from the next block, as shown on line 28.

```
1
2  task void randsample()
3  {
4  bigmsg_frame_part_t *msgData =
5  (bigmsg_frame_part_t *)
6  call FrameSend.getPayload(&tx_msg, sizeof(bigmsg_frame_part_t));
7  uint32_t buf_offset;
8
9          msgData->part_id = a;
10         buf_offset = a+block_offset;
11
12     memcpy(msgData->buf,&(buffer[buf_offset]),len);
13         if(i<64)
14         {
15                 arr[i]= msgData->buf[0];
16                 s=(s * 125)%2796203;
17                 a=(s % blksz)+ 1;
18                 i++;
19                 post randsample();
20                 call Leds.led1Toggle();
21         }
22         else
23     {
24                 if(blkctr<blkpkt) // No. of packets per block
25                         blkctr++;
26                 else
27                 {
28                         block_offset=(block_offset+blksz);
29                         blkctr=1;
30                 }
31         post send();
32         }
33 }
```

Listing 3.1: CS implementation

3.6 Experimentation and Analysis

As mentioned earlier, the sampling part has been implemented on an Imote2 sensor node. The first and second part of the experimentation analyze the sampling parameters that affect CS recovery quality and recovery time, done to validate the modifications made in standard CS sampling methodology through comparisons between reconstruction quality and time. The third part of the experiment has been conducted to observe the performance of CS recovery in presence of packet erasures. In the last part, the transmission efficiency of CS has been analyzed in terms of the number of bits transmitted and also in terms of the increase in number of days of lifetime.

We consider a multi-hop network that transmits data to the base-station. The received data is read into MATLAB using its serial reader interface. The received packet consists of a header and a payload. In this application, the header carries the important information, that of the packet number. As mentioned earlier, the packet number is used as the seed for reconstructing the sampling matrix at the receiver. The payload contains the CS measurements from which the signal is recovered using the Basis Pursuit. It is implemented in MATLAB 7.1(R2010a) on an Intel Xeon 3GHz processor. The SPGl1 package is used for l_1-norm minimization.

Another important part in the analysis is assessing the quality of reconstruction of the images. There are two general approaches in Image Quality Assessment (IMQ), reference based and non-reference based. In a reference based approach a reference image is available for comparison, while in case of no-reference the quality of the image is determined through its statistical properties. For the proposed application of WCSNs for surveillance and monitoring, ideally a no-reference system is required

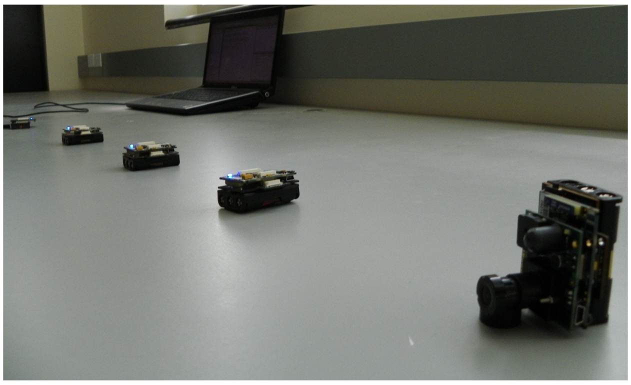


Figure 3.6: Experimental Setup

since in practical situations it is not possible to have reference images. But the research in no-reference IMQ is in a nascent stage and there is no standard and reliable technique available that can be used. Hence, reference based system has been opted for, which finds a great deal of research and many very good methods available for analysis. The Structural Similarity Index (SSIM) is a widely used reference based image quality assessment technique [32] , has been used in this work. To obtain a reference image, we first take an uncompressed no loss snapshot of the object under consideration. All the future reconstructed images are compared against this reference image.

3.6.1 Block size vs Quality and Recovery Time

The first experiment is designed to test the effect of the *block size* on the quality of reconstruction and recovery time. As mentioned earlier block-wise sampling was opted primarily because of the massive memory required to store the ortho-normal matrix for recovering large images. Also the fact that l_1 minimization holds a non-linear relationship with the length of the processed vector gives a much less recovery time for smaller blocks . The trade-off here is that a longer block gives better quality of reconstruction. Also shown is the effect of row-weight to show the significant reduction in recovery time for a low row weight and small block size .

The experiment has been performed on a 128x128 Lenna image, since this is the largest size of single image that could be computed without block-sampling, as more memory was required to store the large Ortho-normal matrix.

Table 3.1: Performance vs variable block size and row weight

No.of blocks	SSIM Rwt 10	SSIM Rwt 1	Time(s) Rwt 10	Time(s) Rwt 1
1	0.9221	0.9203	1600	625
2	0.9215	0.9168	1514	236
4	0.9234	0.9149	974	181
8	0.9164	0.9100	228	68
16	0.9168	0.9004	78	30

The number of samples taken are 50 percent of the original image. Through these results as shown in Table(3.1) it can be seen that the block size has a huge impact on the recovery time, due to the non-linear relationship. Similarly, the row weight is also a major factor in the recovery time. The important thing here is that the difference in quality between the best case result (block size 1, row weight 10) and the worst case (block size 16, row weight 1) is tolerable, but with significant gain in time.

The image results are shown in Fig.(3.7). It can be seen that, as the block size decreases the visual quality of the image deteriorates, so is the case with the row weight too. In the image with large row weight and large block size, the quality is good. In terms of information, many of the image details are retained, even in case of 50 percent compression. While in the image with the lowest row-weight and block size, the quality is not that good, but information wise, the image can still be identified as Lenna, with all the global features retained. It is the finer details that are lost. But, as discussed earlier and seen in the results table, the gain in recovery

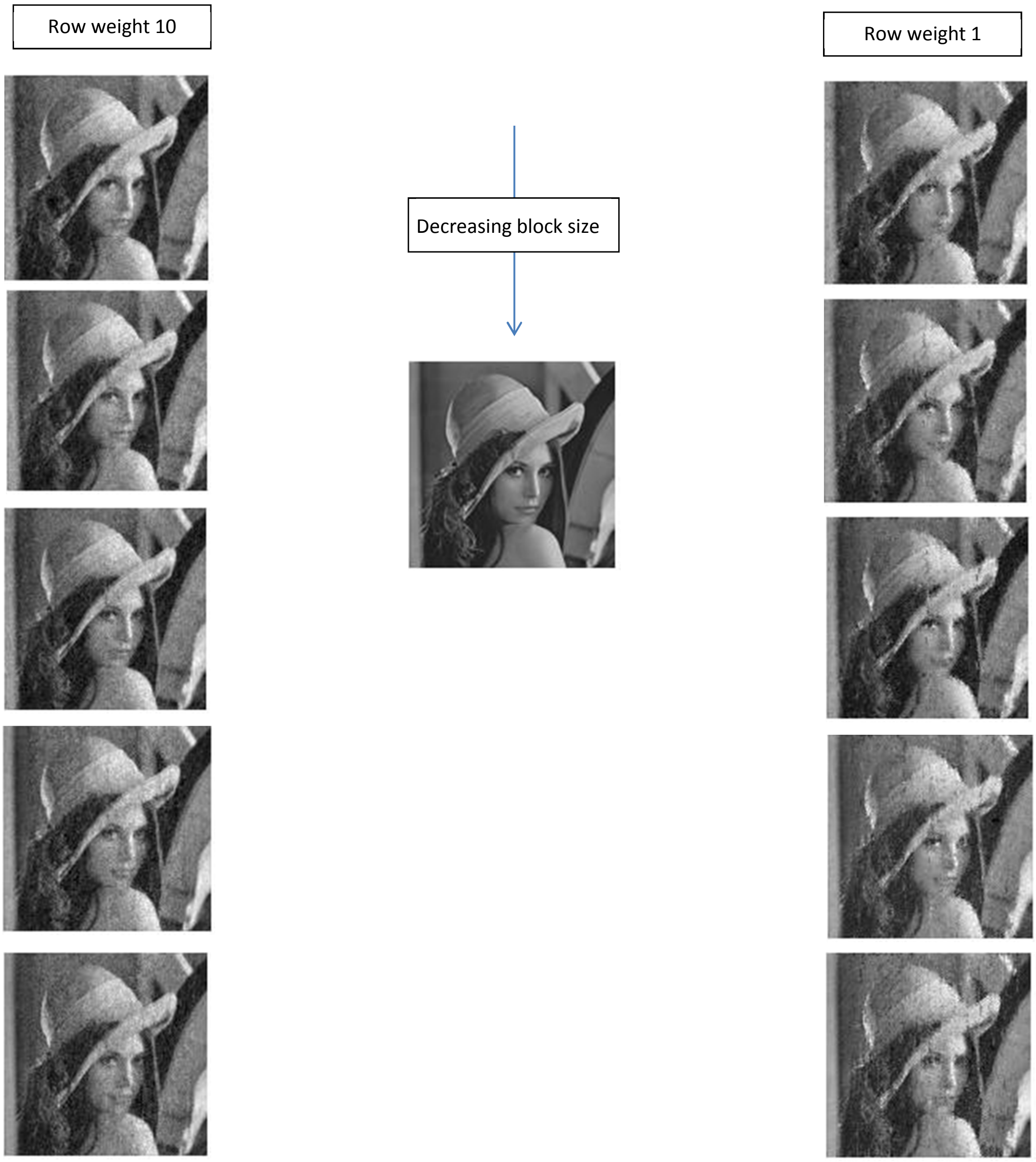


Figure 3.7: Reconstruction results for variable block-size

time is very very high.

A similar observation can be made on actual images captured using a camera sensor with variable row weigh 3.8. The image has testing parameters of, image size n=76800, measurements m= 53760, row-weight l=1. As seen in the image results, there is a small but gradual reduction in quality with decrease in block size. But,

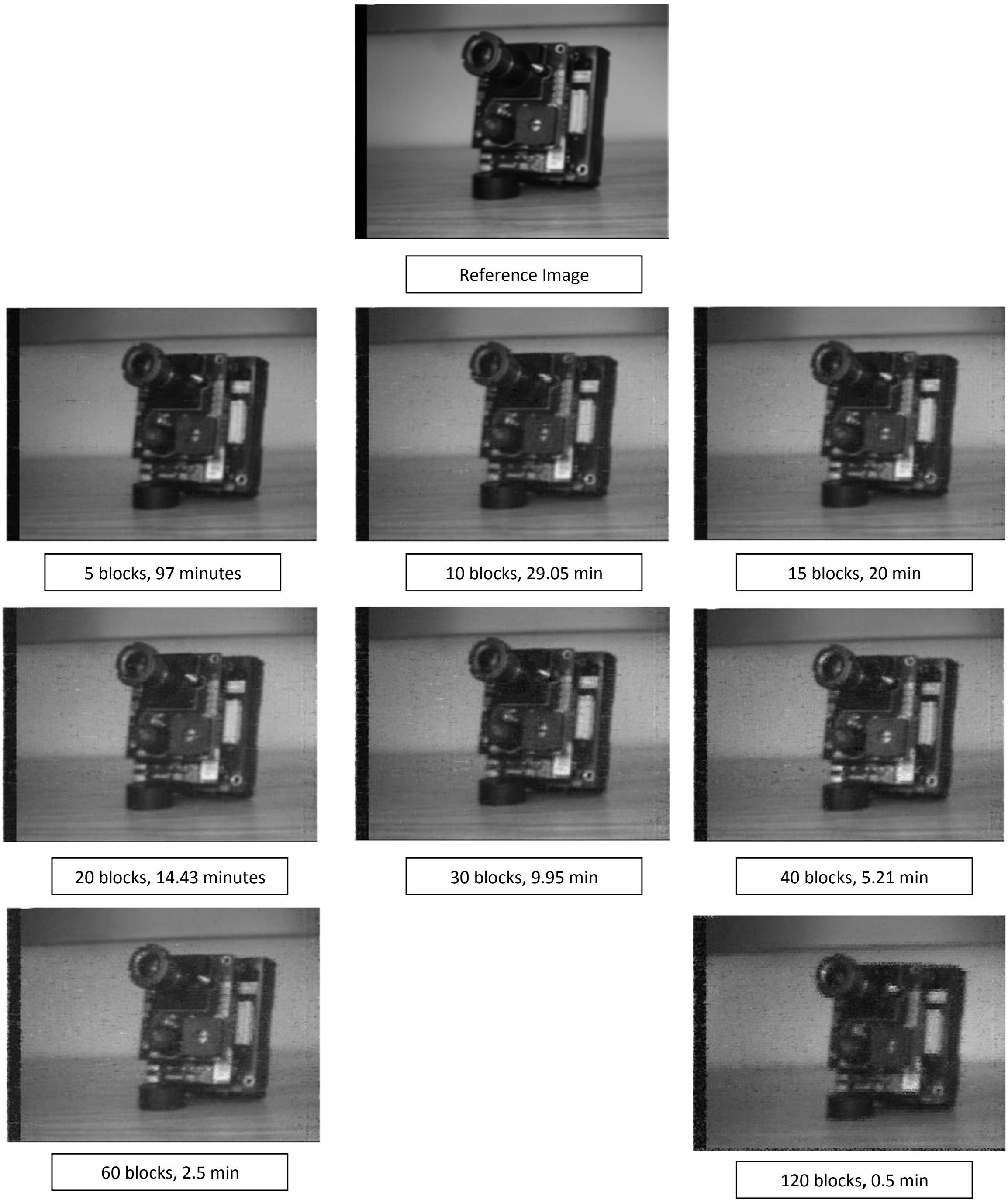


Figure 3.8: Camera sensor image results for variable block-size

the important requirement, especially for applications in surveillance is the retention of the information content of the image. For the images with lower block sizes, even though the quality, the loss in information has been in the minute details, i.e. the smaller features in the image, but the global features have been retained to high extent. Thus, block processing results only in decrease of very minute features, which for most applications may not be even required. But the biggest gain here is the reduction in recovery time. A reduction from 97 minutes to 2.5 minutes with nearly the same quality image is a very significant difference. This reduction in time actually brings CS in the realm of practical applications.

The last image has a significant quality loss, but a recovery time less than a minute. This shows that, it is possible to bring CS recovery to real-time requirements, with a sufficiently good quality of reconstruction. With some more better modifications it might be possible to get a better quality of reconstruction too.

Thus, depending on application and requirement these parameters can be selected. Applications requiring finer details can opt for longer blocks and higher row-weights, while with applications which require a general, global information about the observed scenes can opt for the faster combinations.

3.6.2 Reconstruction performance with variable row weight on real image

As discussed earlier, the row weight of the sampling matrix determines the degree of coherence it shares with the ortho-normal DCT matrix. From Fig.(3.3), it is seen that the coherence increases with increase in row-weight, which is undesirable. But, a row-weight basically signifies how many signal elements are being combined to generate one measurement and a higher row weight ensures more signal information per measurement.

The results of increasing row weight can also be seen for the images obtained from the sensor cameras. The reference image is shown in Fig.(3.9). Since the object is close to the camera, the image is dense, with a number of details being captured. The quality of the reconstructed image can be observed from the number of details preserved.

The original size of these images is 320x240(QVGA), hence the length of the

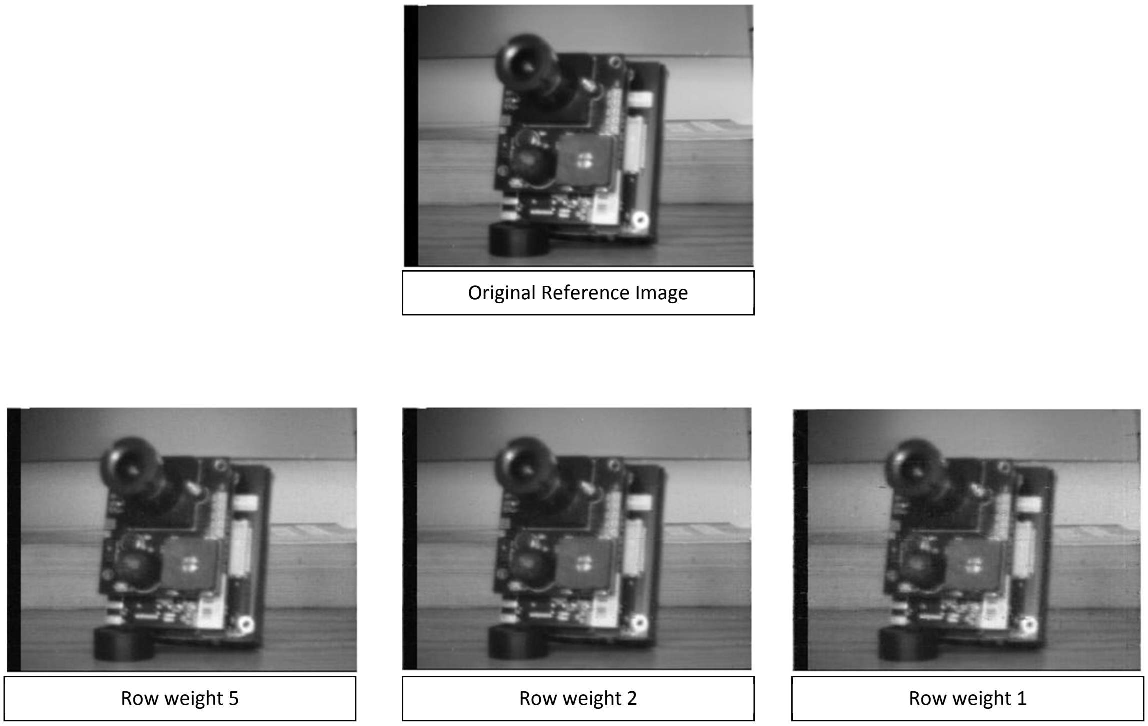


Figure 3.9: Reconstruction performance with variable row weights

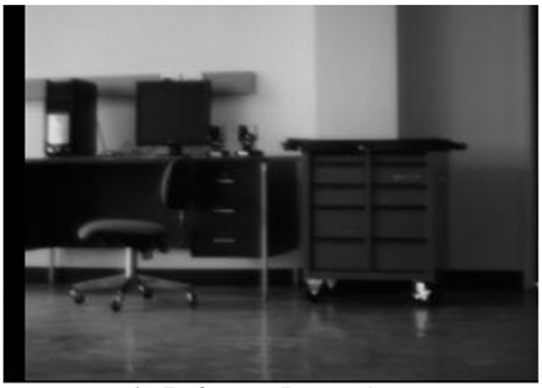
image vector is 76800 pixels. These images have been taken at 70 percent (53760) measurements of the total image size, with a block size of 7680 pixels per block. As seen in the reconstruction performance, Fig.(3.9), there is almost no perceptive difference between the images. Their SSIM values are 0.9512, 0.9396 and 0.9239 and the time taken in minutes is 145.9, 88.2 and 27.6 for row weights 5,2 and 1 respectively. From these results it is clear that with row weight, the quality does not improve much but there is significant gain in time. This makes a very sparse binary matrix (row weight=1) a very useful proposition. This improvement in the recovery time due to the decrease in row weight can be attributed to the fact that, since lesser signal elements are combined, the optimization takes lesser combinations of values, hence speeding up the process.

3.6.3 System performance in case of erasures

In this experiment we test the loss resilience of the sensor network system. For this we use one camera sensor taking snapshot data. It passes this data through a series of intermediate nodes to the base-station. The packets are routed using a simple address based scheme. The intermediate nodes act as transceivers, the variation in packet erasure was obtained by changing the number of hops and the distance between the nodes, keeping the transmission power constant. The degree of erasure is measured by the difference between the number of transmitted packets and the number of received packets.

Considering an image as a source of information , the experiment has been performed on two types of images, one image is a close-up snapshot of an object, termed as a spatially dense image Fig.(3.11) and another image is one capturing a wider area, referred as spatially sparse Fig.(3.12).

The close-up or dense image contains a number of observable and detailed features, and the quality is determined by ability of the image to convey the information about these minute features and conversely , the image quality will degrade as the visibility of these minute features reduces. On the other hand, for the wide area sparse image, it is the overall content of the scene that serves as information. Thus, the nature of the image captured needs to be considered while analyzing the results. The idea of image being used as a source of information is more clear from the images in Fig.(3.10). This is the sparse image used. As mentioned earlier the information needed from this image is the overall scene content, i.e. the number of object, their relative location, etc. Now, because of the packet erasures, about 30 percent packet loss has occurred. Even that has altered basic information such as the number of the objects. For example, there are two small objects besides the computer monitor, but because of the losses, one of them is barely visible and it might result in a false object count for the image. In case of CS even if the visual quality degrades, the information retention is more important.



(b) Reference Image- Sparse

(c) Erasure on image without compression

Figure 3.10: Erasure performance without compressed sampling

The image reconstruction results can be seen in Figures (3.11) and (3.12). These images have been taken at initial compression of 70 percent, so the received data will be much less than that. The important observations is the graceful degradation of the image quality because of CS. It can be seen that the image quality degrades very little for high degree of losses, for example,the effective percentage data received for 60 percent erasure, will be around 28 percent of the image size. Considering that such small amount of data is received the reconstruction quality is good, especially as compared to the degradation observed for the uncompressed image Fig. (3.10).

The role of the image type can be observed through the two image result sets and also through the erasure analysis plot shown in Fig.(3.13). For small degree of erasure, as compared to the sparse image, the dense image has a better quality and conveys decent information about the small observable features, hence its SSIM values are slightly better than the sparse image. But as the degree of erasure increases, the difference reduces and for higher degree of erasure (greater than 60 percent), the sparse image has a better SSIM index. This as discussed before, is because of the information conveyed. As the erasure increases, a denser image even though qualitatively looks better than the sparse image, it conveys lesser degree of information

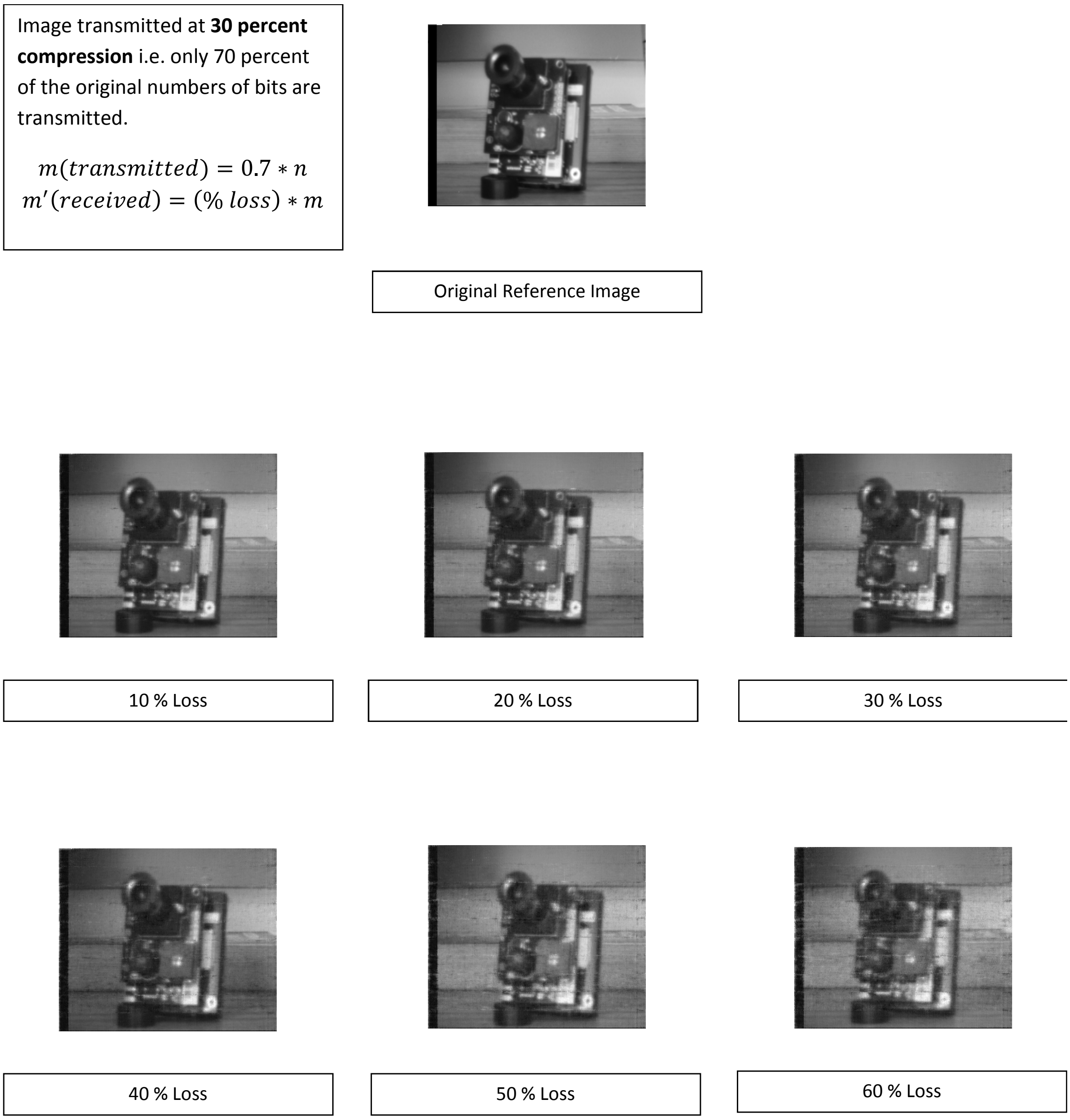


Figure 3.11: Erasure performance for dense image

as compared to the sparse image. Hence, factors such as initial compression, erasure tolerance will depend on the type of scene (which in turn decides type of image being captured) being monitored. Thus, the performance of the system depends on the type of images being captured.

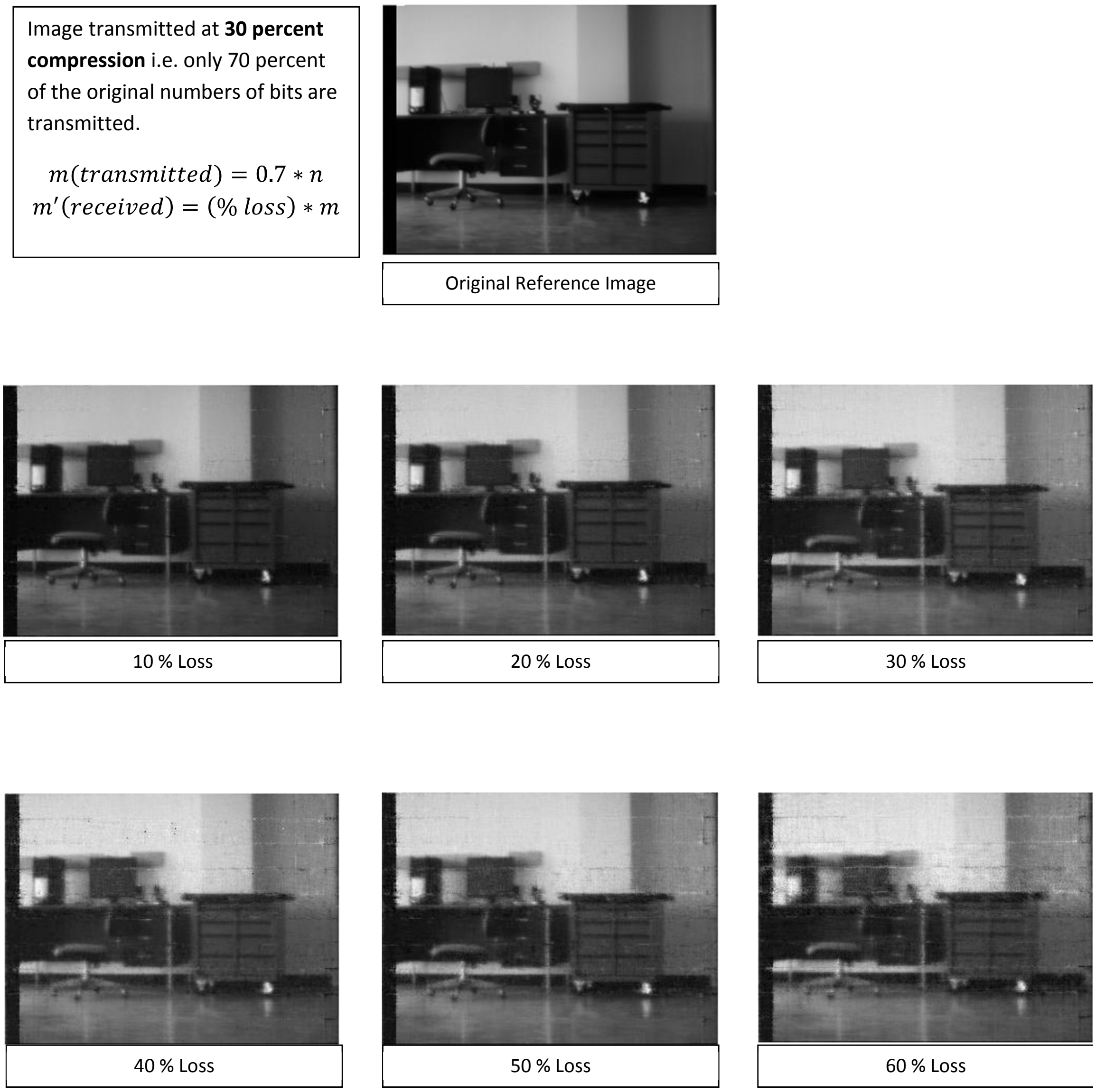


Figure 3.12: Erasure performance for sparse image

3.6.4 Energy saved using CS

We call CS an energy-efficient mechanism. In Imote2 if we transmit at -10dBm power level at which then energy/bit for transmission is approximately 134nJ/b [33].

One image packet comprises of a 64 byte payload and a 13 byte header. Thus each packet is of 77 bytes. A regular QVGA image takes 1200 packets for transmission. Thus, the total number of bytes transmitted are 1200∗77 = 92400. With 134 nJ/b the transmission energy for one complete image will be $92400 * 8 * 134 = 99mJ \approx 100mJ$.

At 70 percent compression, only 840 packets need to be transmitted. Therefore energy required will be, $840 * 77 * 8 * 134 = 69.3mJ \approx 70mJ$. Thus, using CS saves

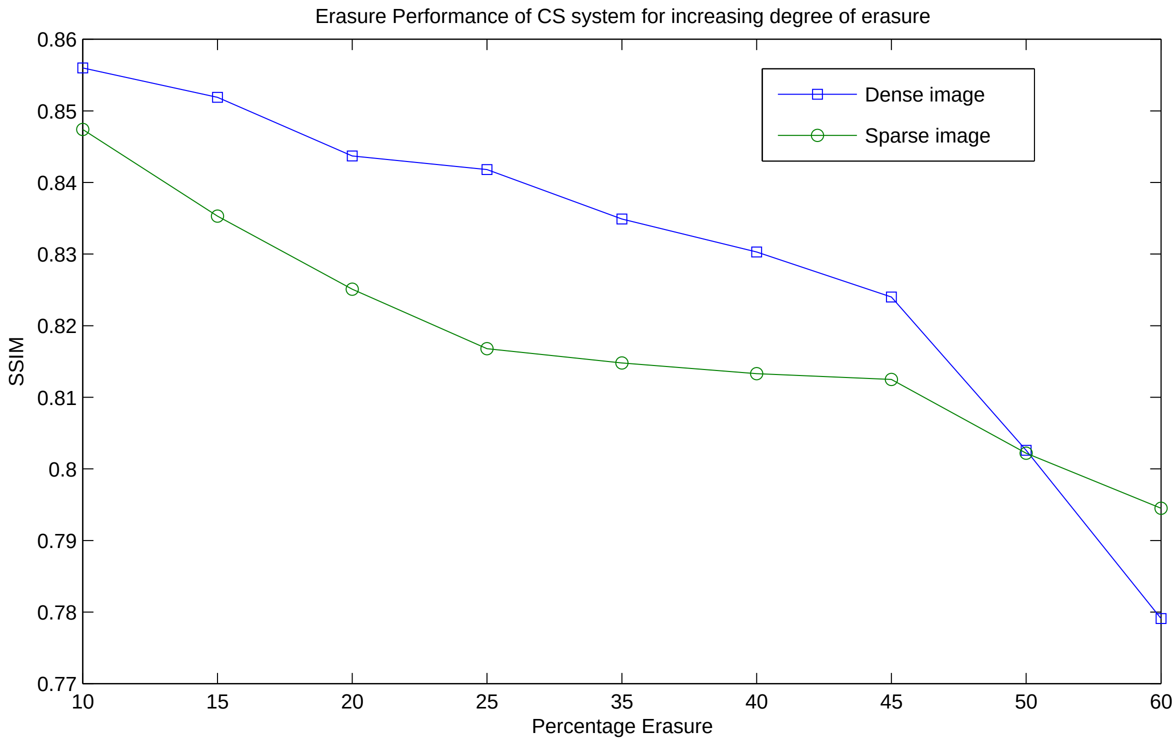


Figure 3.13: Performance of CS for increasing erasure

an average 30 mJ per transmission per node.

Extra energy might be expended in the increase in processing because of CS. The Imote2 has a DSP co-processor along with main processor, making it difficult to calculate the consumption per instruction cycle. But the instruction overhead imposed by the multiplicative congruential generator used for CS includes two extra modulus operations, two additions and 1 multiplication. The processors used are known for very low power operation hence these additional instructions should not consume much and a sizable energy conservation should be expected over a large network.

The number of days the lifetime increases because of efficient transmission are about 1.5-2 days, considering an event arrival rate of 20 events per day. This is clearly not a big increase. But, as discussed earlier the transmission consumption depends on the data size, and for higher resolution images, a better improvement

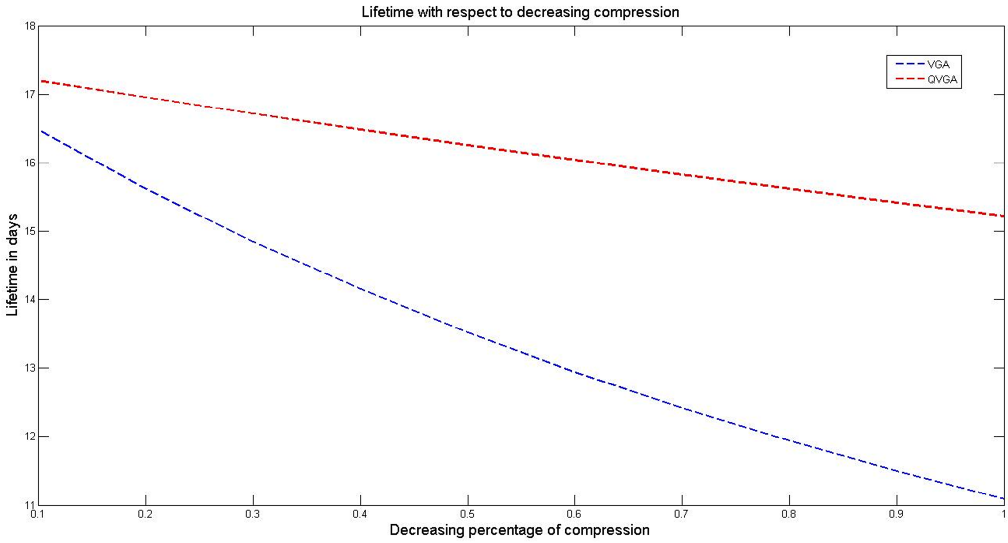


Figure 3.14: Lifetime for variable compression

can be observed, as shown in Fig.(3.14).A VGA image shows a steeper gradient, which means that the effect of CS will be felt more on a VGA image in terms of the lifetime. This proves the effectiveness of CS as mechanism for prolonging camera sensor lifetime.

CHAPTER 4

CONCLUSIONS

In this thesis, an energy efficient and loss resilient mechanism for operating wireless camera sensor networks has been proposed. A framework has been proposed to save up precious energy spent in idle sensing and transmission. The propositions have been supported by implementations on an actual sensor node platform. Considering that these devices may find their biggest application in surveillance and monitoring their operational surroundings have been assumed to be hard-to-reach areas and the assumptions about their working have been made accordingly.

In the first part of the thesis, the energy spent is idle-sensing is saved through the use of an event-driven sleep wake-up mechanism. There are basically two standard modes of operation in sensor networks, schedule-driven and event-driven. Both the modes find useful applications in their own capacity based on their working characteristics. For the application of camera sensors in surveillance, the event-driven mechanism turns out to be a better choice. In this part, an event-driven mechanism has been implemented on an actual wireless camera sensor node hardware, the Imote2. The hardware and software have special properties which, if appropriately harnessed, can be used to implement an efficient event-based wake-up scheme. These properties and the best ways to use them have been discussed. Finally, the significance of this mechanism on the life-time of a sensor node has been discussed through a lifetime analysis, where the consumption characteristics of the Imote2 have been used to determine its probable lifetime. Thus, in this part the relevance, implementation and analysis of an event-driven sleep wake-up mechanism for a wireless camera sensor node has been discussed.

The second part is the more prominent part of this thesis, where conservation has been attempted through efficient transmission. The new technique of data sampling, Compressed Sampling, has been used for this purpose. Compressed sampling also provides an added advantage of making the transmitted data resilient to losses, which is one of the biggest problems in ad-hoc networks. This too has been implemented on the Imote2. The traditional compressed sampling approach is not ideal for operating on sensor nodes and also not practical it terms of operation. In this thesis we suggest a framework suitable for implementation of camera sensors and also suggest modification in the traditional approach to improve the performance. Through the results, the effectiveness of compressed sampling for loss resilience can be seen. The other results show the improvements achieved through our proposed modifications. Finally, the energy saved per transmission is also provided.

Through this thesis we suggest schemes which can help to alleviate problems that might be faced by wireless camera sensor networks in the actual environment. Even though it may not solve all the problems, our work tries to improve some relevant ones. There is ample scope of extending this work, by still improving upon the parameters discussed. The block-wise sampling proposed opens the probability of parallel computing, which can further improve the results. Considering that parallel computing has lot of active research, our propositions can be tested for their optimum capabilities. Unequal compressive sampling strategies can be implemented to give prominence to target objects over the background. This will require practical mechanisms suitable for implementation on sensor hardware, but can definitely improve the results. Using a no-reference image quality analysis on the sensor node to judge the received signal quality in order to influence the future results through a feed-back centric mechanism can also improve the quality of results obtained.

This thesis can serve as a good platform for future research on wireless camera sensor networks, especially in terms of implementing compressed sampling on it and testing better ways on improving their quality of service.

BIBLIOGRAPHY

[1] A. Baggio, "Wireless sensor networks in precision agriculture," in *ACM Workshop on Real-World Wireless Sensor Networks (REALWSN 2005), Stockholm, Sweden*, Citeseer, 2005.

[2] X. Shen, Z. Wang, and Y. Sun, "Wireless sensor networks for industrial applications," in *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, vol. 4, pp. 3636 – 3640 Vol.4, june 2004.

[3] J. Porter, P. Arzberger, H. Braun, P. Bryant, S. Gage, T. Hansen, P. Hanson, C. Lin, F. Lin, T. Kratz, *et al.*, "Wireless sensor networks for ecology," *BioScience*, vol. 55, no. 7, pp. 561–572, 2005.

[4] J. V. Horn, "Los angeles moves to credit card readers and on-street sensors." `http://www.parkingtoday.com/s_article.php?id=358,`.

[5] C. Chong and S. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[6] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[7] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921 – 960, 2007.

[8] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Confer-*

ence of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 3, pp. 1567 – 1576 vol.3, 2002.

[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pp. 10–pp, IEEE, 2000.

[10] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. Stankovic, *et al.*, "Vigilnet: An integrated sensor network system for energy-efficient surveillance," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 1, pp. 1–38, 2006.

[11] A. Arora, R. Ramnath, E. Ertin, P. Sinha, S. Bapat, V. Naik, V. Kulathumani, H. Zhang, H. Cao, M. Sridharan, *et al.*, "Exscal: Elements of an extreme scale wireless sensor network," 2005.

[12] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu, "Senseye: a multi-tier camera sensor network," in *Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 229–238, ACM, 2005.

[13] D. Lymberopoulos and A. Savvides, "Xyz: a motion-enabled, power aware sensor node platform for distributed sensor network applications," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, pp. 63–es, IEEE Press, 2005.

[14] S. Kumar, A. Arora, and T. Lai, "On the lifetime analysis of always-on wireless sensor network applications," in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005.*, p. 188, IEEE, 2005.

[15] D. Jung, T. Teixeira, A. Barton-Sweeney, and A. Savvides, "Model-based design exploration of wireless sensor node lifetimes," *Wireless Sensor Networks*, pp. 277–292, 2007.

[16] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on*, vol. 1, pp. 286–295, IEEE, 2006.

[17] M. Halgamuge, M. Zukerman, K. Ramamohanarao, and H. Vu, "An estimation of sensor energy consumption," *Progress In Electromagnetics Research*, vol. 12, pp. 259–295, 2009.

[18] D. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, pp. 1289 –1306, april 2006.

[19] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, pp. 489 – 509, feb. 2006.

[20] R. Baraniuk, "Compressive sensing [lecture notes]," *Signal Processing Magazine, IEEE*, vol. 24, pp. 118 –121, july 2007.

[21] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," *INVERSE PROBLEMS*, vol. 23, pp. 969–985, JUN 2007.

[22] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *Information Theory, IEEE Transactions on*, vol. 50, no. 10, pp. 2231–2242, 2004.

[23] D. Needell and J. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[24] T. Blumensath and M. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[25] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 1–13, ACM, 2003.

[26] Z. Charbiwala, S. Chakraborty, S. Zahedi, Y. Kim, M. B. Srivastava, T. He, and C. Bisdikian, "Compressive oversampling for robust data transmission in sensor networks," in *Proceedings of the 29th conference on Information communications*, INFOCOM'10, (Piscataway, NJ, USA), pp. 1190–1198, IEEE Press, 2010.

[27] S. Pudlewski, A. Prasanna, and T. Melodia, "Resilient image sensor networks in lossy channels using compressed sensing," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pp. 62 –67, 29 2010-april 2 2010.

[28] R. Berinde, A. Gilbert, P. Indyk, H. Karloff, and M. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pp. 798 –805, sept. 2008.

[29] A. Gilbert and P. Indyk, "Sparse recovery using sparse matrices," *Proceedings of the IEEE*, vol. 98, pp. 937 –947, june 2010.

[30] L. Gan, "Block compressed sensing of natural images," in *Digital Signal Processing, 2007 15th International Conference on*, pp. 403 –406, july 2007.

[31] G. Dwyer, K. Williams, *et al.*, "Portable random number generators," *Journal of Economic Dynamics and Control*, vol. 27, no. 4, pp. 645–650, 2003.

[32] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *Image Processing, IEEE Transactions on*, vol. 13, pp. 600 –612, april 2004.

[33] N. Chohan, “Hardware assisted compression in wireless sensor networks abstract.”

[34]

[35] Crossbow, *Imote2, high performance wireless sensor network node.*

[36] Crossbow, *Imote2 hardware reference manual*, revision a ed., September 2007.

[37] Intel, *Intel PXA27x Processor reference platform user's guide*, version 1.4 ed., September 2004.

[38] M. Gillespie, *Power Management in the Intel PXA27x series Applications Processor.* Intel, April 2005.

[39] Intel, *Using Dialog Semiconductor DA9030x Advanced Power Managment Controllers with the Intel PXA27x Processor Family*, revision 1.0 ed., May 2004.

[40] Dialog Semiconductor, *Cellular Power Management and Audio*, preliminary ed.

[41] D. Jung, *A MATLAB WIreless Sensor Node Platform Lifetime Predictiona and Simulation Package.* ENALAB, Yale University,CT, version 2.0.0 ed., November 2007.

[42] Panasonic, *MP Motion Sensor, Passive Infrared type.*

VITA

Ashish Vijay Wani

Candidate for the Degree of

Master of Science

Thesis: ENERGY EFFICIENT CAMERA SENSOR NETWORKS

Major Field: Electrical Engineering

Biographical:

Education:

Received the B.S. degree from University of Pune, Pune, Maharashtra, INDIA, 2009, in Electronics and Telecommunication

Completed the requirements for the degree of Master of Science with a major in Electrical Engineering, from Oklahoma State University, Stillwater in December, 2011.

Name: Ashish Vijay Wani Date of Degree: December 2011

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: ENERGY EFFICIENT CAMERA SENSOR NETWORKS

Pages in Study: 64 Candidate for the Degree of Master of Science

Major Field: Electrical Engineering

Data loss during transmission and a limited energy source are two main challenges that need to be dealt with in embedded sensor networks. These problems are even more severe in wireless camera sensor networks (WCSNs), owing to the large data size. Energy spent in idle event monitoring and communication, turn out to be the two biggest sources of energy consumption.

An event-based sleep and wake-up mechanism is a suitable option for surveillance applications with long event arrival intervals. With proper use of different hardware and software functionalities an efficient event-based wake-up mechanism can be implemented.

Compressive Sampling (CS) turns out to be an effective solution in reducing the transmission costs and also provides a loss resilient mechanism. It involves under-sampling the data through linear random projections which allows transmission of lesser bits than the original. The randomness in sampling makes the system tolerant to losses without requiring transmission of redundant parity bits. Both these characteristics help us on saving up on energy. The original signal can be recovered from this compressively sampled measurements using l_1 optimization. However, using conventional CS on embedded WCSNs has some implementation related challenges. The processor memory and the recovery time of l_1 optimization, are non-linear with respect to the data size and hence large image sizes may hinder the applicability of CS in practical cases.

In this thesis, a framework for practical implementation of these energy saving strategies has been provided. Issues that affect the practical usability of CS, namely recovery time and memory usage have been discussed and the solutions have been provided, backed up by a number of experimental results. Significant improvements have been observed in the implemented schemes over traditional schemes in terms of recovery time. All the suggested schemes have been implemented on an actual Imote2 sensor node test-bed. This provides a platform for future research and testing of different aspects of WCSNs.

ADVISOR'S APPROVAL: ______________________________