Battery-free Connected Cameras

Saman Naderiparizi

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2018

Reading Committee:

Joshua R. Smith, Chair

Shwetak N. Patel

Matthai Philipose

Program Authorized to Offer Degree:
Electrical Engineering

University of Washington

Abstract

Battery-free Connected Cameras

Saman Naderiparizi

Chair of the Supervisory Committee:
Zeutschel Professor Joshua R. Smith
Allen School of Computer Science and Engineering & Department of Electrical Engineering

Wireless cameras have traditionally been considered extremely power-hungry devices. The battery life of wearable camera systems (such as Google Glass, or Snap Spectacles) is under an hour when capturing video. Similarly, due to power requirements, wireless cameras (such as surveillance and monitoring cameras) are either battery-powered and require very frequent recharge, or must be plugged in, which increases cost and decreases reliability (since the camera can be disabled by attacking its power supply). In addition, power wires limit wireless camera deployment to places that have access to power lines. Existing approaches to wireless camera design optimize the camera and communication modules individually to minimize their power consumption. However, designing a wireless camera device requires power consuming hardware components and computationally intensive compression blocks (CODECs) that interface the camera and the communication modules.

This thesis describes several projects that address these limitations of conventional wireless camera design. *WISPCam* enables battery-free image capture by harvesting energy from Radio Frequency signals. *WISPCam* relies on heavy duty-cycling, so there may be a long latency between consecutive images due to the camera's power requirements.

Glimpse camera, a low power programmable wearable camera architecture for continuous mobile vision, integrates low-power sensing modalities and novel detection algorithms to detect when something interesting is happening in the wearers field of view. This allows a traditionally power

hungry wireless camera system to remain in sleep mode when nothing of interest is occurring in front of the wearer. The *Glimpse* camera approach reduces overall power consumption of the wireless camera by more than one order of magnitude. Despite the significant reduction in wireless camera power consumption, power-wise, *Glimpse* is far from enabling battery-free wireless video streaming.

Finally, our most recent work shows how to redesign the traditional camera architecture in order to eliminate the power-hungry CODEC, ADC, AGC and communication module in their conventional form. This work shows that 30 fps 720p (HD) video streaming can be achieved while burning only $250\mu W$, five orders of magnitude less than available solutions. This work demonstrates that these high-resolution and high frame-rate wireless cameras can be powered by RF signals emitted from an FCC-compliant reader at a distance of up to 12 feet.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Number Page

ACKNOWLEDGMENTS

Firstly, I want to deeply thank my adviser Prof. Josh Smith for being my primary support and help during my PhD. Josh has always been a great guide for me in finding the right next steps in my research. Also, the freedom that Josh gave me truly helped me to thrive, be creative and most importantly enjoy my PhD.

In addition, I want to use this opportunity to thank all of the current and previous Sensor Systems Lab members. I learned from all of them and enjoyed working with them. In particular, I want to thank Dr. Aaron Parks who was my colleague at Sensor Systems Lab and also my initial mentor when I started my graduate studies at UW.

I also want to deeply thank Dr. Matthai Philipose who not only was my mentor during my internships at Microsoft but also became one of my good friends and his guidance and advice was always a good source of support throughout my PhD. Through working with him I was exposed to new line of research which opened up new career opportunities for me.

I also want to truly thank Prof. Shyam Gollakota. I have worked with him in two very exciting projects during my PhD. I learned from him how to dream big in my research. Shyam also thought me how to properly articulate ideas and present it in an understandable way to audiences from diverse backgrounds.

DEDICATION

to my dear wife, Mahshid and my parents, Robabeh and Mehdi

Chapter 1

INTRODUCTION

These days wireless cameras are around us everywhere; they are used in security and home automation, traffic and highway monitoring, wearable applications such as video blogging and life casting [33], elderly and baby care [63, 113] and etc. Cameras are also widely used in machine vision wearable applications such as Google Glass [81] and HoloLens [43].

Traditionally, wireless cameras are known as information-rich and power hungry sensors, thus do not last long on a single charge of their battery. Snap Spectacles which are wearable video recording cameras [24] can record up to only one hundred 10-second videos (effectively less than 20 minutes of total operating time) before they need to be recharged [24, 19]. Google Glass lasts about 40 minutes when capturing video before its battery is depleted [69], and Ring Home Security Doorbell battery charge lasts only a few month if activated very infrequently [28].

Over the past few years we have been investigating to find answer to the below three questions.

1. Why wireless cameras are power hungry devices?
2. How to reduce the wireless cameras' power consumption?
3. Can we enable battery-free image capture, and more importantly, battery-free video streaming?

The answers to the above three questions enables battery-free ubiquitous wireless monitoring cameras that can be deployed anywhere with minimal installment and maintenance costs. Finding those answers, also, turns light-weight, easy-to-use, wearable cameras into reality.

In the remainder of this chapter I start with presenting some background and introduction for the questions above. Also, I provide an agenda for the rest of this dissertation and summarize what each chapter is presenting.

1.1 Why wireless cameras are power hungry devices?

Wireless cameras consist of three main modules: 1) image sensor with integrated high-speed amplifiers, Analog-to-Digital converters (ADC) and Automatic Gain Control (AGC) 2) image or video compression module (CODEC) which is an inseparable part of any wireless streaming cameras and 3) wireless communication (radio) to transmit the image frame data to an access point or a reader. Traditionally all these three pieces are known to be power hungry modules. Recent work has shown feasibility of passive radios (backscatter) comparing to traditional active radios and demonstrated orders of magnitude reduction in power consumption. Despite these major advancements in radio modules, power consumption of CODECs and image sensors have yet remained relatively high.

1.2 How to reduce the wireless cameras' power consumption?

After identifying the major power consuming modules of a power hungry system we should either try to reduce their power consumption or eliminate them if possible in order to develop a power efficient system. Three different major approaches are presented and fully characterized in this dissertation:

- **Duty-cycling:** One naive approach to reduce the power consumption of a system without changing its architecture is to duty-cycle it. In other words, if the target is to achieve two orders of magnitude of power saving, instead of operating the device for 100% of the time, we can turn it on for only 1% of the time and keep it shut down for the remaining 99% of the time.

- **Gating:** Majority of videos collected by some of wireless cameras mentioned above do not have important/interesting contents. For example, a home security camera does not need to be active when there is no activity happening in its environment. We can use some low-power sensors in conjunction with detection algorithms to gate the operation of a wireless camera.

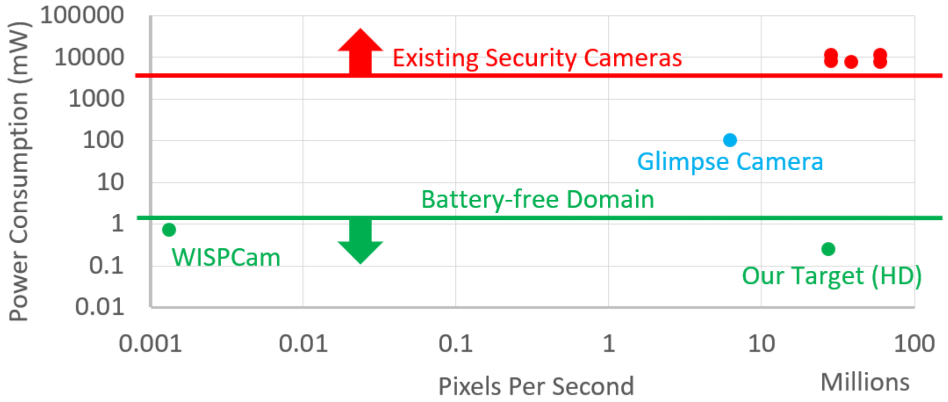


Figure 1.1: **Power versus pixel rate for existing wireless cameras and our work.** Existing wireless video streaming cameras burn orders of magnitude more than the available budget on battery-free platforms.

For example, an ultra-low power Passive Infrared (PIR) sensor can detect if there is any thermal activity in the background and based on that gates a power hungry wireless camera system. Effectively, *gating* can be viewed as a smart duty-cycling approach for reducing power consumption, in which no or negligible information of interest will be lost.

- **Camera architecture redesign:** A sophisticated approach to reduce the power consumption of a system is to redesign its architecture. For a particular example of wireless cameras, we can jointly re-design the camera architecture, from wireless communication, to video CODEC and image sensor. This enables a very power optimized system tailored around the widespread wireless camera application.

1.3 Can we enable battery-free image capture, and more importantly, battery-free video streaming?

Batteries have the recharging overhead and, due to use of chemicals in them, batteries have limited life-cycle period. Eliminating batteries from wireless systems and replacing them with capacitors as the charge reservoir enables zero-maintenance operations and lightweight systems that their life-cycle is as long as their electronics life-cycle.

Radio Frequency (RF) energy harvesting is an efficient mean of delivering power to wireless sensors. Wireless systems already have an antenna integrated with them which is used for communication and it can be re-purposed for energy harvesting as well. In addition, comparing to other well-known sources for energy harvesting such as light that are available at a particular time of the day, RF is more ubiquitous and is available at any time.

Power budget for battery-free systems that harvest energy from RF signals is up to a few milli-watts. This means developing a battery-free wireless video streaming system requires bringing down the traditional wireless camera power consumption to about milli-watts level or below. Fig. 1.1 shows the power consumption for already existing wireless video streaming cameras (red dots) and also the power consumption for the system we have developed (green and blue dots). The green horizontal line in this figure shows the boundary for a region that battery-free operation is realistic within. Clearly there is about 5 orders of magnitude power consumption gap between already existing wireless video streaming solutions and what is available on battery-free devices.

1.4 Dissertation Agenda

Here I summarize what each chapter of this thesis is aiming at.

In chapter §2 we discuss our efforts to develop *WISPCam*, the world's first battery-free wireless camera that harvests energy from RF signals. This chapter presents how we build upon already existing battery-free programmable Wireless Identification and Sensing Platform (WISP) to design and build the *WISPCam*. *WISPCam* leverages an off-the-shelf image sensor and integrates that with WISP. This chapter also discusses what are the fundamental challenges we face and our design decisions when enabling battery-free operation for a power hungry sensor such as a conventional image sensor. In this chapter we also show how we managed to transmit rich image data generated by the camera.

WISPCam, in fact, heavily **duty-cycles** a traditional image sensor to reduce its average power consumption, in the other words, comparing to traditional wireless video streaming cameras, *WIS*-

PCam does not stream live video, instead it captures and transmits still images at discrete timestamps. *WISPCam* is marked with a green dot in the bottom left corner of Fig. 1.1.

In chapter §3 we discuss how we can leverage *WISPCam* to perform tasks beyond image capture-only devices. We show how *WISPCam*s can localize themselves and enable location-aware operations. We also show, how we can go around the limitations *WISPCam* has such as low-resolution image capture capability and limited on-device computational power to enable applications that are traditionally known to be resource hungry such as face recognition.

In chapter §4 we discuss Glimpse camera, a low-power programmable wearable camera platform for continuous mobile vision. Glimpse system uses low-power sensors and feed their data into its on-board low-power Field Programmable Gate Array (FPGA) and microcontroller running custom designed low-power detection algorithms to detect if something interesting is happening in the wearer's field-of-view.

Glimpse explores how **gating** reduces the power consumption and bandwidth utilization of a traditional wearable wireless camera system. Glimpse is marked with a blue dot in Fig.1.1. Despite more than one order of magnitude in power reduction using Glimpse comparing to conventional wireless camera systems, its power consumption by far exceeds battery-free power budget.

In chapter §5 we present a new camera architecture pipeline to enable battery-free HD video streaming. This chapter explains what have been the limiting factors avoiding us from having wireless video streaming cameras that burn as low as sub-milli-watt of power. This chapter explains how we **redesign the camera architecture** to eliminate the need for power-hungry CODECs, amplifiers, ADC, AGC, and wireless communication to develop the world's lowest power 30 fps HD video streaming camera that burns only $\sim 250\mu W$ of power. This new camera system is shown with a green dot in bottom right corner of the Fig. 1.1 and falls within the power budget for battery-free devices.

Chapter 2

WISPCAM: A BATTERY-FREE RFID CAMERA

2.1 Introduction

Recent research has shown that battery-free sensing is feasible with passive RFID technology. Early research-grade systems describing battery-free sensing first targeted sensing of simple quantities, such as temperature or light intensity[104], or highly targeted phenomena, such as neural signals[131] or specific motions[36]. In recent years, commercial sensing systems based on RFID technology have become available, targeting applications ranging from strain monitoring to pick-to-light systems to tire pressure monitoring for aircraft[7, 22].

A commonality with all existing RFID-based sensors is the simplicity of the sensing they support. Existing systems perform sensing either by using extremely low-power sensor modules operated at a low output data rate to conserve power, or by cleverly employing varying electrical properties of the antenna that depend on environmental quantities[107]. This simplicity is largely attributable to limited on-board energy storage. Applications requiring a high data rate, such as neural recording[131], have used small data packets that can be computed and sent just before a small on-board capacitor runs out of energy. More complex sensing reduces the effective data rate; commercial products performing sensing of simple quantities have a 1 Hz advertised output data rate and operate up to 1.5 m from an RFID reader[7].

A naïve approach to overcoming the limited sensing capabilities of battery-free platforms is to increase on-board energy storage by merely substituting a larger storage capacitor. Greater capacitance enables longer-lasting or more energy-intensive sensing tasks, such as temperature sensing outside the range of an RFID reader[132], but naïvely integrating additional capacitance brings a significant drawback: parasitic leakage current in charge-storage devices typically *increases* with capacitance, threatening to consume an unacceptable share of an energy-scavenging platform's

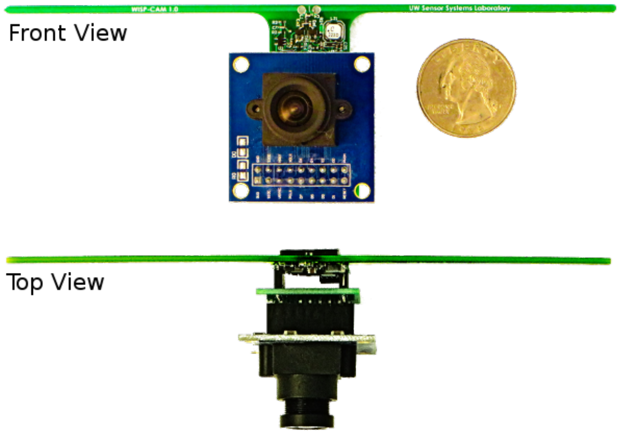


Figure 2.1: The prototyped *WISPCam*, a wireless and battery-free (passive) UHF RFID camera tag. The *WISPCam* is currently implemented as a stack of three PCBs: a modified WISP 5.0[25], an Omnivision OV7670 camera module development board, and a pin-compatibility adapter.

tight power budget.

A second challenge for richer sensing is that complex sensors typically produce large amounts of data during operation. When coupled with low-power microcontrollers that constrain RAM size to limit static power consumption, a sensor (e.g., a camera module) may produce more data than can fit in RAM. Storing to on-board *nonvolatile* memory has traditionally not been an option because writing to flash memory is prohibitively slow and energy intensive. Past systems have resorted to offloading data storage over the low-throughput backscatter link to avoid flash memory[103].

Taking these challenges into account, an energy-harvesting platform supporting complex sensing must balance charging time, application responsiveness, and richness of sensor data to meet the needs of the target application.

This chapter proposes, designs and evaluates the *WISPCam*, a battery-free camera based on the Wireless Identification and Sensing Platform (WISP)[111]. In comparison with prior RFID-attached sensors, a camera is a relatively complex and power-hungry load (millijoules, not microjoules, per sensing task), and it produces a much larger amount of data (tens of KB, versus a few bytes). To meet the needs of its off-the-shelf camera component, the *WISPCam* focuses on maximizing the *efficiency* of large-scale energy storage and data transmission with parameters chosen to balance all of the aforementioned concerns. To support the camera's large (about 25 KB per capture for a gray scale 176*144 QCIF image frame) data size, the *WISPCam* takes advantage of newly available nonvolatile FRAM memory technology that enables quick buffering of image data for later transmission to an RFID reader. The *WISPCam* reliably transmits an image over a sequence of many smaller transmissions within the EPC Class 1 Gen 2 protocol. The *WISPCam* makes battery-free image sensing feasible for monitoring and detection applications.

Contributions. This chapter makes two primary contributions in addition to the design and implementation of the *WISPCam*: (1) a charge-storage model that takes leakage into account to optimize the choice of storage capacitor, and (2) an image capture and transmission scheme tailored to battery-free sensing that leverages fast nonvolatile memory and backscatter communication.

Section 2.2 describes the design space for the system and further explores the power and data handling challenges outlined above. Section 2.3 describes specifics of the *WISPCam* prototype implementation. Section 2.4 evaluates the prototype with respect to power consumption, energy harvesting and storage efficiency, and general performance. Section 2.5 discusses two potential applications for the *WISPCam* and presents a preliminary proof-of-concept for these two applications.

2.2 Design of the WISPCam

This section describes two major challenges in order to build a battery-free RFID camera tag like the *WISPCam*: efficiently collecting enough energy to run a rich sensor, and handling storage and transmission of a large data structure that does not fit in RAM.

2.2.1 Challenge: Efficient and Effective Charge Storage

Prior WISP-based sensing devices required tens of microjoules per sample when performing sensing tasks[104, 132]. In contrast, capturing a frame from a modern camera module requires on

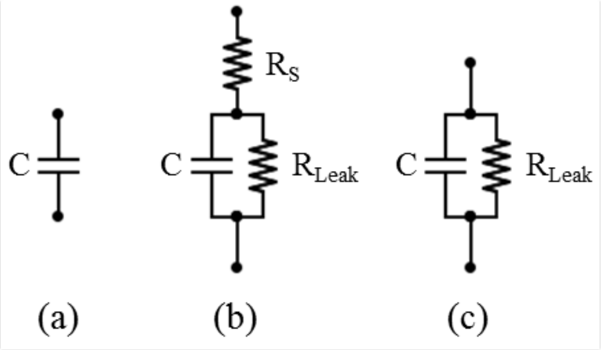


Figure 2.2: Ideal supercapacitor model (a), model with parasitics (b), and simplified model neglecting R_S (c). For the employed supercapacitors, R_S is less than 1Ω[3].

the order of tens of *milli*joules per image capture. Capturing an image requires a brief period of high current, resulting in power consumption far greater than the RF harvester's maximum output. Supplying enough energy for this operation is the *WISPCam*'s primary energy-storage goal.

Because a large on-board supply may take significant time to charge, minimizing leakage current is an important design goal. Excessive leakage current would hinder the platform's basic operation—even low-power operations would become high-power operations—and potentially consume energy that should be devoted to image capture.

An additional consideration is the longevity of the energy-storage element; applications in need of an RFID camera tag presumably seek a solution that does not require frequent maintenance. Supercapacitors and rechargeable batteries seem to be the best candidates for the *WISPCam*'s charge storage. However, rechargeable batteries do not last as long as supercapacitors; studies of supercapacitor longevity estimate that well over a century of operating life is achievable at room temperature and at low current[62]. Further, rechargeable batteries have the added inefficiencies of both electrical-to-chemical (charge cycle) and chemical-to-electrical (discharge cycle) energy conversion.

Unfortunately, supercapacitors are also not ideal. They typically exhibit disproportionately large leakage currents when compared to batteries with the same energy capacity, and these para-

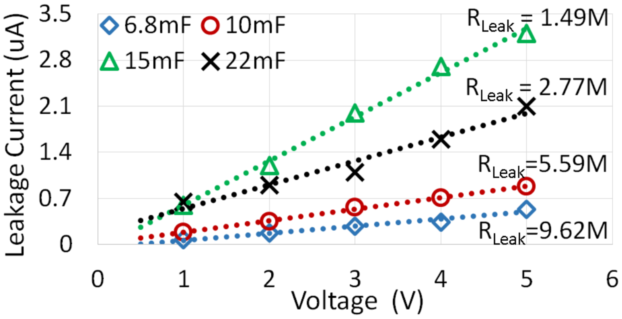


Figure 2.3: Leakage currents of tested supercapacitors as a function of voltage. Leakage can be closely approximated by a simple resistive model, and extracted resistance values (R_{Leak}) are given for each device.

sitic currents reduce the efficiency of charge storage.

Given the above concerns, the main question is how to most efficiently accumulate a high level of charge on leaky capacitors via potentially slow RF harvesting.

The first step in designing with leaky capacitors is to understand the system's power budget to evaluate the impact of leakage current. We expect the *WISPCam* to be capable of harvesting hundreds of microwatts of power from the RFID reader's carrier, and to be consuming tens of microwatts to tens of milliwatts depending on the operation taking place. The next step is to find a supercapacitor that has an appropriately small parasitic conductivity. Figure 2.3 shows the effective parallel DC resistance and leakage power consumption of several candidate supercapacitors we measured. while the voltage across supercapacitor is below 4V, each exhibits leakage power consumption of under $12\,\mu$W, two orders of magnitude smaller than the gap between expected harvester average output power and sleep-mode power consumption of the WISP's microcontroller, and are therefore reasonable choice for the energy storage component in our design.

To power its camera module, the *WISPCam*'s capacitor must charge to an upper threshold V_{max} via RF harvesting, then discharges until it reaches a minimum operating voltage V_{min}. The amount of energy transferred by the capacitor to the system during the discharge cycle—in this case, the energy required to capture and transmit a photo—must match the energy requirement of the load.

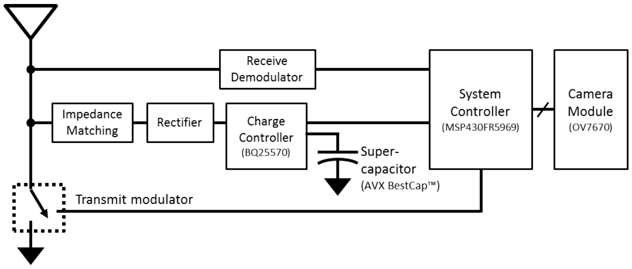


Figure 2.4: *WISPCam* block diagram. A Texas Instruments MSP430 microcontroller with 64K of integrated FRAM memory implements the system controller. DC-DC conversion, supercapacitor charging, and efficient switch-mode load regulation is handled by a TI BQ25570 power management IC. The backscatter uplink and R=>T downlink demodulator are based on those employed by the WISP 5.0[25].

The energy stored in a capacitor is related to its terminal voltage as well as its capacitance. Therefore, the same energy may be stored by choosing a larger capacitor and lower V_{max} threshold, or by choosing a smaller capacitor and a higher V_{max}. Because supercapacitor leakage is approximately resistive in nature, a higher V_{max} threshold voltage means more leakage power. However, higher capacitance *also* increases leakage current. Thus, the tradeoff between capacitance and voltage threshold must be carefully weighed to determine the optimal parameters for high charge-storage efficiency.

Simplified non-ideal supercapacitors can be modeled as in Figure 2.2[127]. In this work we use model (c), as the serial resistance is negligible at low current. Further, we assume that incoming power is less than required operating power, forcing the system to duty cycle. The appropriate supercapacitor is the one that minimizes energy dissipation per charging operation within the minimum and maximum voltage thresholds. The relevant parameters in our model are as follows:

- E_{load}: required energy per sensing operation;
- V_{min}: minimum voltage threshold, greater than or equal to the platform's minimum operating voltage;

- V_{max}: maximum voltage threshold, calculated using equation 2.1;
- C: capacitance of the supercapacitor;
- R_{Leak}: leakage parallel resistance of the supercapacitor, from datasheet or empirical measurement;
- P_{in}: input power after RF–DC and DC–DC conversion, assumed to be constant in each experiment.

In each operating cycle, the supercapacitor will discharge from voltage V$_{max}$ down to V$_{min}$, as shown in Figure 2.5. So V$_{max}$ can be calculated as follows:

$$E_{load} \approx \frac{1}{2}C(V_{max}^2 - V_{min}^2)$$

$$V_{max} = \sqrt{\frac{2E_{load}}{C} + V_{min}^2} \tag{2.1}$$

We then work towards computing the amount of energy dissipated by the parasitic leakage path in the supercapacitor over the entire charging cycle. This will be a function of the input power, as faster charging means less overall energy wastage:

$$P_{Leak} = \frac{v^2}{R_{Leak}} \Rightarrow E_{Leak} = \int_{t_{min}}^{t_{max}} P_{Leak}(t)\,dt. \tag{2.2}$$

Substituting dt with $\frac{C}{i}dv$,

$$E_{Leak} = \int_{V_{min}}^{V_{max}} P_{Leak}(v)\frac{C}{i_c}\,dv. \tag{2.3}$$

Assuming the input power after RF–DC and DC–DC conversion is a constant P$_{in}$, and using $i_c = \frac{P_{in}}{v} - \frac{v}{R_{Leak}}$,

$$E_{Leak} = \frac{C}{R_{Leak}} \int_{V_{min}}^{V_{max}} \frac{v^2}{\frac{P_{in}}{v} - \frac{v}{R_{Leak}}}\,dv. \tag{2.4}$$

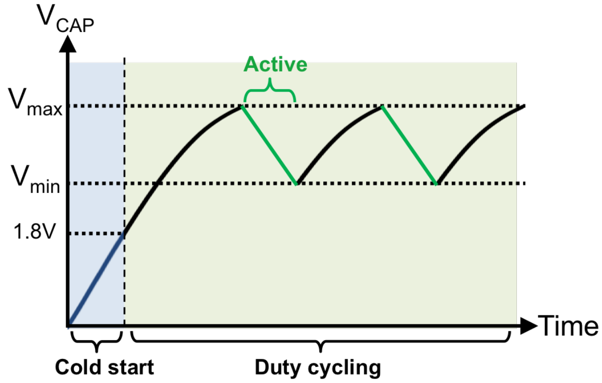


Figure 2.5: Like most energy harvesting systems, the *WISPCam* operates in a duty-cycled fashion, charging a storage capacitor up to a maximum voltage V_{max} before activating. The system remains active until the capacitor voltage falls to V_{min}. In this work, the leakage energy during the charging cycle is used as a cost function for optimization.

For the supercapacitor charge to increase, it is required that supercapacitor leakage power is less than the total input power: $P_{in} > \frac{v^2}{R_{Leak}}$. Finally, using a simplifying substitution $U = \frac{1}{P_{in}R_{Leak}-v^2}$ and defining $U_{min} = \frac{1}{P_{in}R_{Leak}-V_{min}^2}$ and $U_{max} = \frac{1}{P_{in}R_{Leak}-V_{max}^2}$ we will have:

$$E_{Leak} = \frac{C}{2}\left(P_{in}R_{Leak}\ln\left(\frac{U_{max}}{U_{min}}\right) + \frac{1}{U_{max}} - \frac{1}{U_{min}}\right). \tag{2.5}$$

Equation 2.1 assumes that energy dissipated by the load is far more than the energy dissipated by R_{Leak} while the capacitor is discharging. In other words, the leakage energy is negligible during discharging (since leakage power is small), thus approximately the total leakage energy dissipates just during the charging cycle. This will be valid in cases where the active period is much smaller than the charge period; in other words, most high-energy loads will behave in this way. Considering active mode capacitor leakage energy is difficult because it requires full time-domain characterization of the load power requirements and the capacitor's charge state during active periods.

In equation 2.4, t_{min} and t_{max} are the time the supercapacitor begins charging from V_{min} and the time its voltage reaches V_{max}, respectively.

Table 2.1: Comparing capacitor leakage energy during charge cycle

	6.08 mF	**11.24 mF**	**17.45 mF**	**21.98 mF**
Capacitor's Rated Voltage	15V	12V	5.5V	9V
Measured $\mathbf{R}_{Leak}$	9.62 MΩ	5.59 MΩ	1.49 MΩ	2.77 MΩ
Selected $\mathbf{V}_{max}$ **for** $\mathbf{E}_{load}$**=20 mJ**	3.80V	3.38V	3.18V	3.11V
$\mathbf{E}_{Leak}$ **@** $\mathbf{P}_{in}$ **= 10 uW**	2.63 mJ	4.18 mJ	31.03 mJ	9.26 mJ
$\mathbf{E}_{Leak}$ **@** $\mathbf{P}_{in}$ **= 100 uW**	0.23 mJ	0.35 mJ	1.28 mJ	0.65 mJ
$\mathbf{E}_{Leak}$ **@** $\mathbf{P}_{in}$ **= 1 mW**	0.023 mJ	0.034 mJ	0.121 mJ	0.063 mJ

To select the optimum supercapacitor, we calculate V_{max} and R_{Leak}, then use equation 2.4 to find E_{Leak}. The capacitor that minimizes E_{Leak} during the charging period is the most efficient one. Table 2.1 summarizes these data for our candidate supercapacitors, assuming $V_{min} = 2.8\,\text{V}$, $E_{Load} = 20\,\text{mJ}$, and all capacitors are charged with the same input power. The table 2.1 shows that if the charging power of the supercapacitor is high (around 1 mW), then the leakage current impact on the overall efficiency will be negligible (less than 1%). On the other hand, for low charging powers (around 10 uW) improper selection of supercapacitor may have a huge impact on the overall efficiency of the system and may turn into a limiting factor for minimum sensitivity. Our experimental results on the efficiency of power harvester (see 2.9) justify this table's results. Based on 2.9 the efficiency of the *WISPCam* when a 17.45 mF supercapacitor is connected to the output of its power harvester drops drastically once the input power is low. As a conclusion, the table shows that the 6.08 mF supercapacitor is the best candidate among those we tested.

RF power harvesting considerations

Additionally, it is desirable to have a power harvester that can dynamically track the maximum power point, since the impedance of the charge storage element is changing and charging simultaneously.

As discussed before, we decided to use a supercapacitor with millifarads capacitance as the charge storage element. The *WISPCam*'s microcontroller and camera require voltages above 2.5 V to operate properly, so the employed supercapacitor must charge up to at least this level, meaning the load connected to the power harvester is changing significantly during charging. In this case, in order to get the most out of the input RF signal, it is important to be able to adaptively change the current drawn from the rectifier's output to keep the input impedance of the system matched to the antenna as much as possible. Otherwise, energy-harvesting efficiency will be poor. To work around this challenge we used a TI boost charger that performs Maximum Power Point Tracking (MPPT) to increase the efficiency during charging.

2.2.2 Challenge: Large-Scale Data Capture and Transfer

Given a platform like the *WISPCam* that captures images on behalf of an RFID-based application, a primary design point is to ensure that image data can reach the application. Unfortunately, large-scale (tens of KB) storage and transmission are complicated for this class of devices.

To minimize static power consumption due to memory, microcontrollers suitable for energy-harvesting devices typically have at most a few KB of RAM[130, 111], not enough to hold a complete image larger than 100×100 pixels. Streaming image data directly to an RFID reader during capture is infeasible because of the poor performance of RFID protocols for large-scale streaming[122], and because energy shortfalls can cause the entire platform, including sensors, to lose power mid-transfer. The remaining option is nonvolatile memory on board the sensor platform. Unfortunately, the ubiquitous flash memory in microcontrollers is too slow ($\sim 1\,$s per 10 KB), too cumbersome to write (awkward erase-before-write requirements), too energy intensive (0.1μJ/byte[79]), and too limited in durability to be practically useful for buffering of large sensor data[14].

Figure 2.6 illustrates the importance of data-handling speed for the *WISPCam*. The figure shows the energy required by the camera to capture one image frame against the time required to buffer the frame to nonvolatile memory. Since the camera is the most power-hungry component of

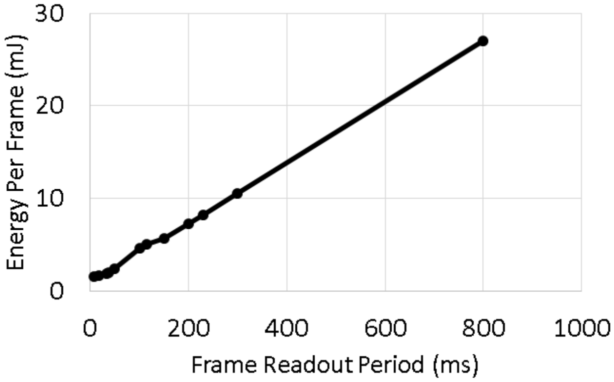


Figure 2.6: Camera per frame energy consumption vs frame readout period

the *WISPCam*, the optimal strategy is to turn on the camera, capture a frame, transmit the frame data to memory, and then put the camera in sleep mode before transmitting to a reader.

As an alternative to flash, the *WISPCam* is built on a microcontroller that includes ferroelectric RAM (FRAM). FRAM is superior to flash by orders of magnitude along all of the dimensions described above. Using FRAM instead of flash enables a 48 Mbps transfer speed on the *WISPCam*'s microcontroller—fast enough to transfer a camera image to nonvolatile memory before exhausting the capacitor's supply. Section 2.3 gives more details on the parts involved.

2.3 Prototype implementation

We categorize the implementation of the whole system into two sections, hardware implementation and firmware/software implementation. Most of the control logic for each module of our design is implemented in firmware, as well as the EPC Class 1 Generation 2 (EPC C1G2) protocol for UHF RFID tags which allows communication with the reader. The hardware section focuses on the power subsystem, which provides the energy harvesting and storage necessary to operate the camera tag while creating a communication infrastructure.

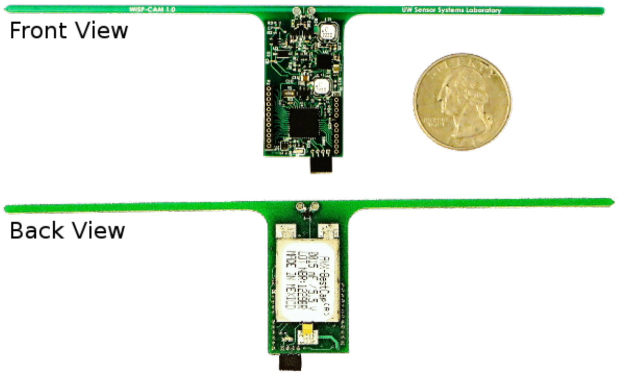


Figure 2.7: *WISPCam* main board, based heavily on the WISP 5.0[25]. An AVX BestCap series supercapacitor allows the *WISPCam* to accumulate the 20 mJ necessary for image capture and transfer.

2.3.1 Hardware implementation

Figure 2.4 shows the block diagram of the *WISPCam*. The harvesting system consists of an antenna to intercept RF signals produced by an RFID reader, a discrete impedance matching network, an RF-DC conversion stage utilizing the Avago HSMS-285C Schottky barrier diodes in a single-stage RF Dickson charge pump configuration, and a high-efficiency DC-DC boost conversion stage making use of a Texas Instruments BQ25570. Charge is stored on a supercapacitor chosen from the BestCap series by AVX.

Downlink communication (reader to tag) is accomplished through the use of an envelope detector and power-gated ASK demodulator circuit. The output of the demodulator is connected to the logic controller, a Texas Instruments MSP430FR5969 microcontroller. The microcontroller runs open source firmware[25] to interpret the EPC C1G2 protocol and issues responses to reader queries. Responses are transmitted via backscatter communication by modulating the impedance presented to the antenna through the transmit modulator switch shown.

Sensing load: Camera

The sensing load of this device is a camera, a very generally applicable sensor which requires a lot of energy per operation and is a source of a relatively large amount of data when compared with the typical byte-size quantities recorded by prior battery-free sensing loads.

We employ the OV7670 camera module due to its wide commercial availability. The nominal operating voltage for this camera is 2.8 Volts. Although the employed microcontroller, which is TI MSP430FR5969, can operate at as low as around 2 V, we choose to apply the same 2.8V to the microcontroller to eliminate the need for level shifting on the large parallel data bus which connects the microcontroller and camera.

Meeting energy requirements

The BQ25570 boost charger IC operates in a power cycling mode, in which charge is accumulated on the storage capacitor until and upper threshold voltage (V_{max}) is breached, then discharged through the output regulator until a lower threshold voltage is attained (V_{min}). The output regulator is also a switch-mode regulator designed for low-power loads, and maintains a far higher efficiency than the linear regulator often used in low power and energy harvesting systems[5].

Based on empirical data presented in Figure 2.10, we found that at least 10 mJ of energy is required to capture and store a 176*144 pixel (QCIF) gray scale image such as those seen in Figure 2.12 (Images with higher contrast or higher resolution will require expending more energy). We select a 6.08 mF supercapacitor and target 20 mJs of usable energy. This will ensure that the entire atomic imaging, parallel data read-out by the microcontroller, and storage operation can be finished in one active period, a requirement for successful image capture. We set V_{min} to 2.8V; to achieve 20 mJ of charge storage, by Equation 2.1 we select $V_{max} \approx 3.8$V.

Maximum power point tracking

As discussed in Section 2.2, because of changing load it is important to have a power harvester that uses Maximum Power Point Tracking (MPPT). MPPT effectively performs DC impedance matching of the RF–DC source to the DC–DC converter's input, even as the charge storage device's effective impedance changes throughout the charge cycle. This DC impedance match is reflected by a lowered S11 magnitude at the RF input side. The boost charger, with its built-in MPPT feature, resulted in an S11 magnitude better than $-25\,\text{dB}$ over the entire supercapacitor charging cycle for RF input powers smaller than -12 dBm.

2.3.2 Firmware implementation

Firmware running on the microcontroller initializes the camera, triggers the shutter, and receives image frame data through an eight-bit parallel bus (clocked at 6 MHz) and several control lines.

The *WISPCam*'s MSP430FR5969 microcontroller has a 64KB on-chip FRAM memory that supports 48 Mb/s data transfer from the camera to the microcontroller while burning only a few milliwatts of power. The *WISPCam* uses the DMA module of the MSP430 to perform this high-speed transfer of image data to FRAM without the intervention of CPU.

Host-side application

Communication between the *WISPCam* and a host computer takes place through a commercial-off-the-shelf RFID reader (Impinj Speedway R1000). We developed a host-side application that relies on an open-source Python library for interfacing with RFID readers[18]. This application implements a simple transfer protocol with the *WISPCam*, listening for image data and requesting missing or corrupted packets if errors occur during transfer. Image data is transferred through EPC modulation [104], in which certain fields of the electronic product code (EPC) of the RFID tag being emulated by the *WISPCam* contains the data payload from the camera.

Saving state

The *WISPCam* implements a hibernation state in which critical status information is saved to non-volatile memory prior to a power loss event. For instance, the *WISPCam* maintains the address of the next image pixel to be transmitted. If execution is interrupted, the *WISPCam* can resume

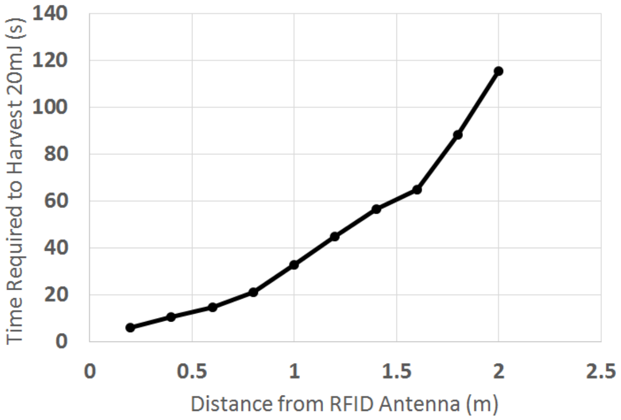


Figure 2.8: Time required to harvest 20 mJ versus the distance between the *WISPCam* and the RFID reader. A 9dBic circularly polarized antenna was used as the reader antenna and a 1 W power level was set at the reader, making for 4 W EIRP for the *WISPCam*'s linearly polarized antenna.

transmission from nonvolatile memory.

2.3.3 WISPCam Prototype

Figure 2.1 shows the actual prototyped *WISPCam* which is an Omnivision OV7670 VGA camera board on top of a modified WISP 5.0. Figure 2.7 illustrates the front and back view of the modified WISP 5.0 board. The employed 6.08 mF supercapacitor is soldered on the back side of the modified WISP 5.0 and takes up almost all the space on the bottom side.

2.4 Results

This section presents the results of three different experiments we performed to measure the performance and power consumption of the *WISPCam*.

The first experiment's goal is to evaluate efficiency of the power harvester while harvesting from a continuous RF 915 MHz signal. The *WISPCam*'s boost charger unfortunately exhibits poor cold-start performance. The boost charger will be in a cold-start state as long as the voltage across the supercapacitor is less than 1.8V. The minimum sensitivity of the system to do a cold start is about -12.1dBm, although this will improve due to MPPT once supercapacitor voltage is above 1.8V (past cold start). In this experiment, we measured the capacitance of four different supercapacitors from AVX by connecting them to a Source Measure Unit (SMU) and reading the slope of rising voltage while providing them with constant current. The nominal values are 6.8 mF, 10 mF, 15 mF and 22 mF whereas the measured capacitances are 6.08 mF, 11.24 mF, 17.45 mF and 21.98 mF accordingly. Then, we connected these four supercapacitors to the output of the power harvester one at a time. The RF input power changed from -12.1 dBm to 8.9 dBm and the supercapacitors in each of the four cases were charged from 2V to 3.8V. Using LabVIEW data acquisition, we logged time-domain charging information to a file and measured efficiency based on equation 2.6, where Δt is the total charge time (charging from 2V to 3.8V). Figure 2.9 shows the efficiency curves of the different supercapacitors. From this figure we can see that by using a 6.08 mF supercapacitor for lower input power, we can achieve more than 23% efficiency, and this trend will increase while the input power increases to some extent, further increase in the input power will result in decreasing the efficiency.

$$\eta(\%) = 100\frac{\frac{1}{2}C(3.8^2 - 2^2)}{P_{in}\Delta t} = 100\frac{5.22C}{P_{in}\Delta t} \tag{2.6}$$

The second experiment measures the *WISPCam*'s power consumption of the system in the time domain for both image capturing/storage (camera and microcontroller in active mode) and image transferring (camera in low-power mode and microcontroller in active mode). As Figure 2.10 shows, initially the system is not powered, and therefore the power consumption is zero. Once powered on, the system enters the image-capturing state, where power consumption spikes for 115 ms. During this phase the *WISPCam* burns about 8.51 mJ. After it is done with image capturing/storage, it enters the image-transfer state where the camera is no longer in active mode. This phase lasts about 3 s and the *WISPCam* consumes 11.34 mJ during that. The overall process takes about 3.2 s and the *WISPCam* burns 19.85 mJ over one entire image capturing, storing and transmission process.

The third experiment evaluates the performance of the *WISPCam* prototype in terms of distance

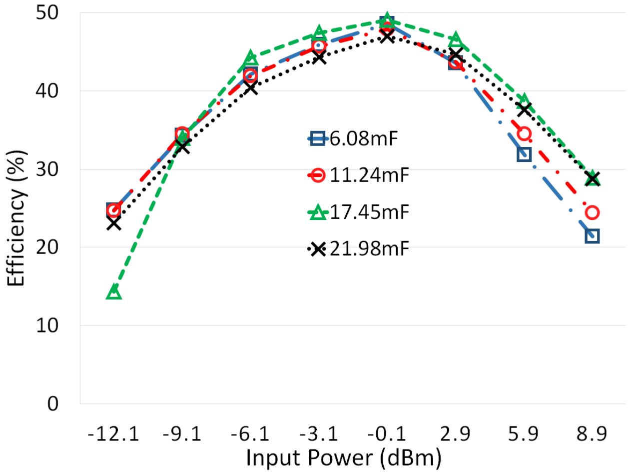


Figure 2.9: Energy efficiency of the power harvester when charging four different supercapacitors from 2V to 3.8V

from an RFID reader which transmits 1 W of RF power and is connected to a 9dBic circularly polarized antenna. Figure 2.8 shows that as the distance from the antenna increases, the power harvester requires more time to harvest 20 mJ, as expected. Based on this figure, if the *WISPCam* is 20 cm away from the RFID antenna in a single-tag environment, it will emit a new picture captured and transmitted approximately every 10 seconds. It is important to note that our experiments show that in a single tag environment, the *WISPCam* works up to 5 m away from the RFID antenna and captures and transmits an image approximately every 15 minutes.

2.5 Applications

The *WISPCam* allows for battery-free imaging by harvesting wireless power and using low-power backscatter communication to both capture and transmit images. The major significance of the *WISPCam* is its ability to capture images without the limitations of batteries, excess wiring, or complex installation, while still remaining a low-power system. These functions and features introduce many promising real-world applications for the *WISPCam*.

The device can be used for applications such as monitoring inaccessible areas or general

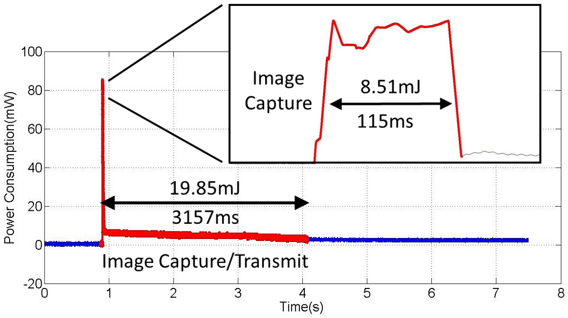


Figure 2.10: Time domain *WISPCam* power consumption while capturing, storing and transmitting the image

surveillance (e.g., home security). For instance, in the microelectronics industry, there are semiconductor fabrication plants that utilize numerous analog gauges. These analog gauges are often located in hard-to-reach places and require frequent monitoring. The *WISPCam* can be an effective solution for this application because it is completely battery-free and wireless, resulting in a very low-maintenance and cost-efficient system. Similarly, monitoring analog gauges on gas tanks can be useful—for example, surveying an analog pressure gauge on a helium tank to determine when a replacement is needed.

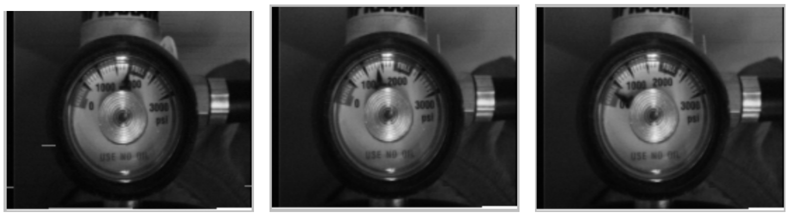
In order to demonstrate the potential for these two applications and many others, we developed a software interface to communicate with the *WISPCam* and display captured images in real time, as shown in Figure 2.13. We implemented the interface in Python using various software libraries, including a pure-Python LLRP library[18]. We mounted the camera in front of an analog gauge, shown in Figure 2.11, with an Impinj reader and an antenna nearby. The reader was connected to a PC with the software interface configured to display the captured images of the analog gauge. These images are shown in Figure 2.12.

Another application for the camera is in the field of security and surveillance. The *WISPCam* is advantageous compared to current surveillance systems because it is wireless and battery free. These are appealing features because they eliminate potential scenarios such as power disconnec-

Figure 2.11: The *WISPCam*, mounted on an industrial gas cylinder, monitors a pressure gauge.

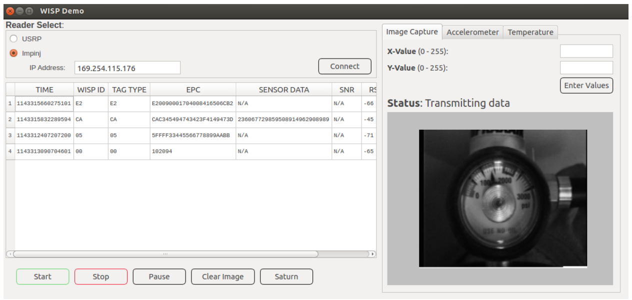
tion that would disable the system. Additionally, we integrated a Passive Infrared (PIR) motion sensor with the *WISPCam* to trigger the capture of an image only when movement is detected. To integrate the motion detector, we implemented a simple circuit that consumes a negligible amount of power (about 10 uW) using the *WISPCam*'s harvested energy and uses the output signal from the motion sensor as a trigger to enable the *WISPCam*. This feature can be useful in environments that require monitoring unexpected changes in the surroundings. To validate the integration of the PIR motion sensor, we used the software interface to collect captured images from the *WISPCam* triggered by the sensor, shown in Figure 2.14.

The *WISPCam* would also be beneficial in aircraft inspection. All airlines are required to have a maintenance plan that schedules the regular maintenance of each airplane in the fleet. Typically, these inspections occur monthly and involve opening areas of the aircraft that are somewhat inaccessible[27]. Many of these maintenance checks are overnight activities, however they can also have a duration of 20 to 30 days[27]. Installing the *WISPCam* in areas of an aircraft that require frequent and extensive inspection would allow for more constant monitoring and potentially improve the efficiency and safety of airline maintenance plans.



(a) (b) (c)

Figure 2.12: A mechanical pressure gauge as captured by the *WISPCam* using the setup in Figure 2.11. These images could be post-processed to extract the gauge reading, or displayed directly to a human operator.



Reader Select:
○ USRP
● Impinj
IP Address: 169.254.115.176

	TIME	WISP ID	TAG TYPE	EPC	SENSOR DATA	SNR	RS
1	1143315660275101	E2	E2	E2009000170400841 6506CB2	N/A	N/A	-66
2	1143315832289594	CA	CA	CAC345494743423F4149473D	236067729859508914962908989	N/A	-45
3	1143312407207200	05	05	5FFFF33445566778899AABB	N/A	N/A	-71
4	1143313090704601	00	00	102094	N/A	N/A	-65

X-Value (0 - 255):
Y-Value (0 - 255):
Status: Transmitting data

Figure 2.13: A *WISPCam* host application with GUI handles data transfer and image display.

2.6 Conclusion

This chapter presented the *WISPCam*, a battery-free camera based on a modified WISP 5.0, and addressed the challenges around harvesting large amounts of energy and handling large amounts of data on a transiently powered platform. We presented a method for selecting an optimal supercapacitor from a set of real supercapacitors given a load energy requirement and an input power. This method is generally applicable to systems operating high-energy loads in energy-harvesting scenarios.

The *WISPCam* was able to cold-start and operate at -12.1 dBm RF input power, storing 20 mJ of energy on a supercapacitor. Once started, the *WISPCam* was able to operate down to -15 dBm

(a) (b) (c)

Figure 2.14: A person passing through a doorway triggers a PIR sensor mounted on the *WISPCam*, which then captures an image of the scene.

input power. The *WISPCam* successfully and repeatedly captured and transmitted images to a host application via a commercial-off-the-shelf RFID reader, proving the feasibility of a battery-free camera system.

Example applications demonstrated include gauge reading and surveillance. However, a camera sensor is widely applicable to a variety of sensing applications, making this one of the first generally applicable battery-free sensing systems.

Chapter 3

RF POWERED FACE DETECTION

3.1 Introduction

The energy efficiency of computing has been improving on a steady exponential trajectory for the last 70 years at least, a trend that pre-dates integrated circuits. Today's microelectronics is about one trillion (10^{12}) times more energy efficient than early computers such as Eniac [64] [110]. Thanks to this energy efficiency scaling, it became possible in the early 2000s to power general purpose microcontrollers and low-power sensors using propagating radio waves as the only power source [98] [112] [106]. Most recently, it has become possible to power more demanding workloads, such as cameras, using only radio waves [84] [90] [115]. This article reviews our work on the WISPCam, an RF-powered camera that communicates via backscatter, and describes the challenges of mapping a machine vision application onto this highly constrained system.

3.2 RF Powered Mobile Camera

The *WISPCam* is a passive wireless RFID tag enhanced with a VGA camera that leverages RF signals for power and communication. A photo of the *WISPCam* is shown in Figure 3.1. The *WISPCam* has an on board low-power microcontroller enabling it to communicate with the camera to store image data as well as handling the RFID backscatter communication protocol. Additionally, the *WISPCam* utilizes a power harvester to accumulate energy from RF signals and then store the energy on a charge reservoir.

The major significance of the *WISPCam* is its ability to capture images, perform additional computation, and transmit data wirelessly without the limitations of batteries, excess wiring, or complex installation, yet still remaining a low-power system. These functionalities and features introduce many promising real-world applications for the *WISPCam*. A few of these applications

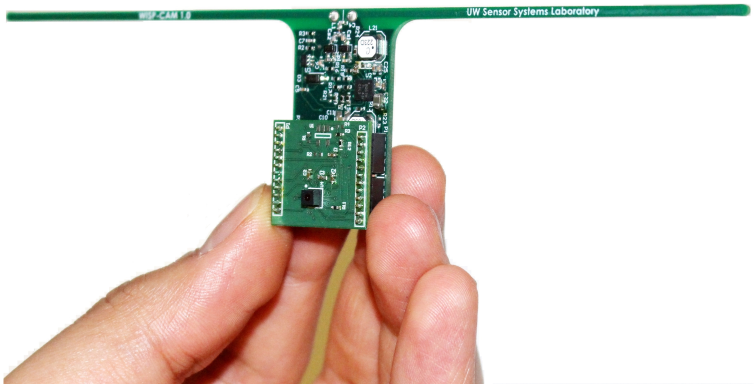


Figure 3.1: Latest prototype version of *WISPCam* with an off-the-shelf cellphone VGA camera

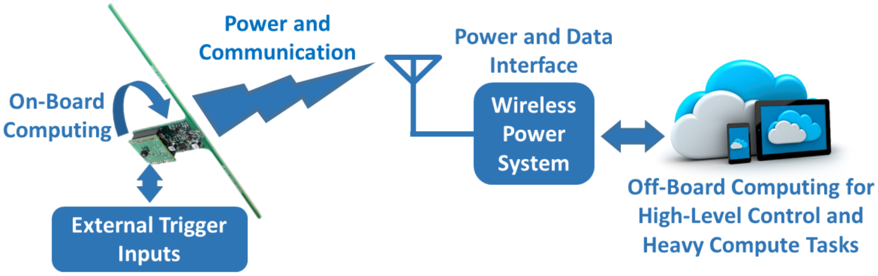


Figure 3.2: High-level block diagram of *WISPCam* deployment with its necessary components that will help eliminating its limitations

include, but are not limited to, monitoring inaccessible areas, surveillance (such as home security), or even monitoring parking availability. In fact, the *WISPCam* can be an effective solution to the aforementioned applications because it is completely battery-free and wireless, resulting in a very low-maintenance and cost-efficient mobile vision and computation platform. Despite enabling many applications, the *WISPCam* standalone has three primary constraints:

1. **Communication speed limitation**. The effective bit-rate using off-the-shelf RFID readers is less than 100kbps.

2. **Limited computational capabilities**. *WISPCam* is an energy scavenging device, thus for

it to operate properly cutting its computation corners in exchange of power requirement reduction is a vital factor.

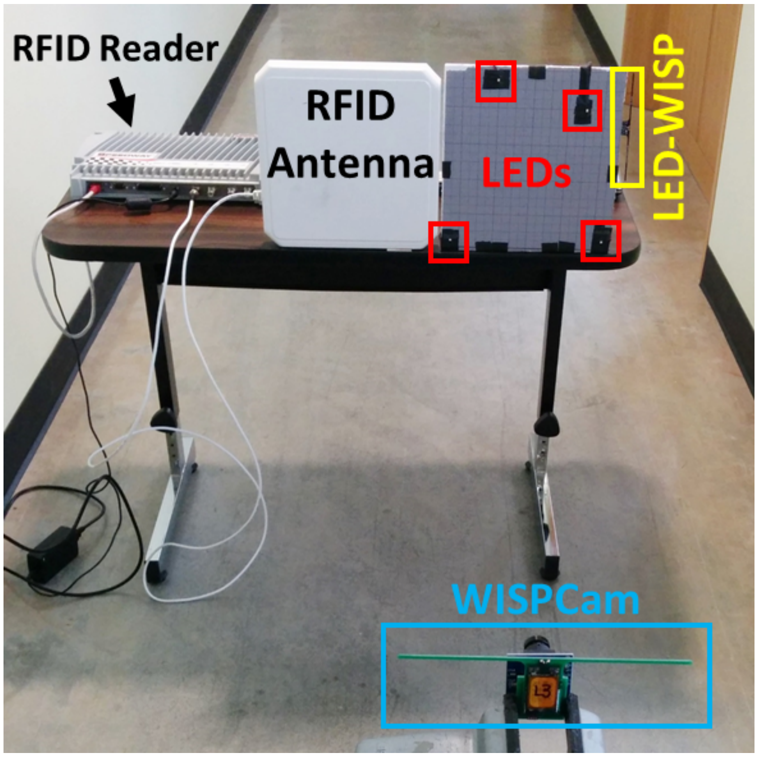
3. **Limited data/code storage space**. Its total on-chip low-power non-volatile memory (FRAM) is limited to about 64 kilobytes of data and code memory.

As a result, the *WISPCam* requires some intelligence to partition its operation efficiently. Essentially, the *WISPCam* should decide to work around its storage, communication, and computation limitations by spreading its workload to the on-board computer and host PC (cloud). Additionally, if a device like the *WISPCam* lacks the intelligence to capture images based on a target application, it is probable that it will capture and transmit a vast amount of data that contains insignificant information. For instance, in the case of the *WISPCam* as a surveillance camera, the *WISPCam* should only capture an image whenever some form of movement is occuring in its field of view.

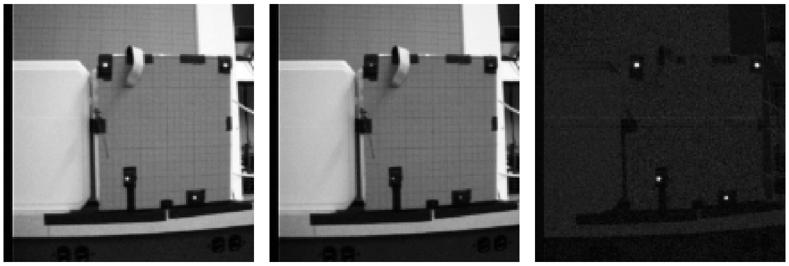
Figure 3.2 shows the high-level block diagram of *WISPCam* system deployment and how we can decompose and balance tasks on the cloud, allowing the *WISPCam* to overcome its limitations. In essence, the *WISPCam* can compress its backscatter data by either light computations or application based low-power triggering inputs. This will result in removing storage and communication barriers to some extent. Then in a higher level process, heavy computational tasks can be off-loaded to the cloud to overcome computation and storage limitations.

As another example, if a host PC wants to trigger the *WISPCam* and assign a task, based on the energy requirements of the task, the host should either distribute the work into smaller tasks or wait until the *WISPCam* harvests enough energy. To elaborate, image capturing is an atomic operation and triggering this operation when the total available energy is less than the minimum required energy will result to a failure and thus wasting the stored energy. This conveys that the host PC should have some notion of the *WISPCam*'s available energy. This can be achieved by having a low-power timer running at all times, which wakes the CPU on the *WISPCam* every few seconds to sample its charge level and send that over to the host PC. In the remaining of this article, we will demonstrate how we implemented the full-system deployment of the *WISPCam*.

3.3 On-Board Computation



(a) Localization setup



(b) Foreground Image (c) Background Image (d) Subtracted Image

Figure 3.3: Localization setup and a foreground, background and subtracted images sample of *WISPCam*. *WISPCam* will detect the four LED coordinates by subtracting background and will transmit only 4 coordinates instead of an entire image.

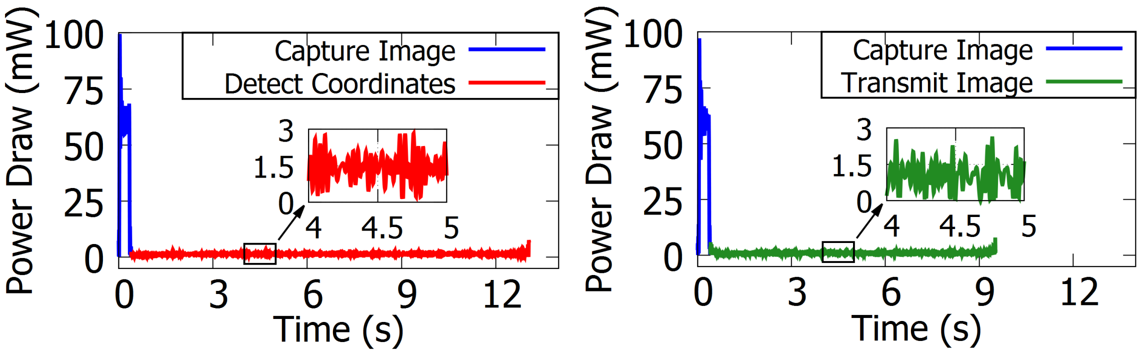
Localization is a key aspect that allows us to take more advantage of the tag. Specifically, in the case of a *WISPCam*, knowing the location from which an image was captured can enable different applications that are location dependent. A few examples include, environmental modeling, 3D object reconstruction, and location triggered image capturing that can be potentially enabled with a location-aware *WISPCam*.

To that end, we reuse the on-board camera of the *WISPCam* to optically localize itself by having an optical cue in the field of view. The idea is to detect four pixel coordinates in the image plane that correspond to four known locations in the field of view. To enable this, four LEDs that are powered and controlled by a battery-free RFID tag (LED-WISP) are placed in four known locations in the field of view of the *WISPCam*. The *WISPCam* will then captures two back-to-back images using its camera. The first image is called foreground, that has the four LEDs on, and the second image is called background, that has the four LEDs off.

The task of the LED-WISP is to listen to the communication between the *WISPCam* and an RFID reader and then synchronize itself to the *WISPCam* whenever a known secure key is transmitted between the RFID reader and the *WISPCam*. Detecting the occurrence moment of the secure key transmission will allow LED-WISP to know exactly when the *WISPCam* is going to capture the foreground and background images. Consequently, the *WISPCam* will be ensured to have a pair of proper foreground and background images, where their subtraction will simply give four bright corners that correspond to the four LEDs coordinates. The setup of this implementation is shown in Figure 3.3a.

The important point here is that the foreground and background images, as well as the detection of LEDs coordinates, is being done on the *WISPCam*. Thus, instead of sending an entire image to the RFID reader, only eight coordinates are being transmitted. This reduces data transmission to the RFID reader per WISPCam by a factor of 1680. Therefore, this will make our localization method practical for a very dense network of the *WISPCam*. Instances of $foreground$ and $foreground - background$ images are shown in Figure 3.3.

We explored the *WISPCam*'s power consumption while doing the aforementioned on-board computation. Figure 3.4a shows the power consumption of the *WISPCam* leveraging on-board computation to detect LEDs coordinates, while Figure 3.4b shows power consumption profile in a



(a) (b)

Figure 3.4: 3.4a shows detecting the LED coordinates leveraging on-board computation and 3.4b shows transmitting the both raw foreground and background images without any computations.

scenario where the *WISPCam* transmits its entire image frames to the host PC without doing any computation. The energy required for capturing foreground and background images, computing LED coordinates, and transmitting raw image data is $20.5mJ$, $17.47mJ$, and $8.19mJ$, respectively. Leveraging on-board computation to compute LED coordinates is a more power hungry task than transmitting the raw image, which is due to the fact that backscatter is a passive radio. On the other hand, paying that extra energy to compute LED coordinates on-board will introduce about 1700 times less traffic on the network.

3.4 Cloud Computing

Most vision algorithms are computationally intensive. Thus, it is not possible to implement them on an energy and computation limited platform like the *WISPCam*. For instance, the Viola-Jones face detection algorithm takes approximately $426ms$ on a $2.2GHz$ notebook to detect faces for a VGA image [126]. This means it is impossible to implement face detection in real-time on the *WISPCam*. Some other vision tasks, such as face recognition, are even more computationally intensive. Consequently, we require a combination of on-board and cloud computation to enable face recognition using the *WISPCam*.

(a) (b) (c)

Figure 3.5: *WISPCam* will capture a 160×120 low resolution image for face detection which is shown in 3.5a. Since this image does not have enough resolution extracting the face by zooming in will result in a poor quality image 3.5b. Next, *WISPCam* will capture a windowed high resolution image of the face according to the coordinates of the face 3.5c

On the other hand, face recognition will have a higher recall rate on higher resolution images up to a certain image resolution threshold [125]. However, the total amount of data memory available on the *WISPCam* is just enough for an image that has a resolution 10 times lower than a VGA image. Nevertheless, face detection is possible with lower image resolutions in comparison to face recognition [78]. With this in mind, our idea is to capture and send a subsampled low resolution 160×120 image. Then, a host PC will detect the coordinates of the windows that contain faces and report them back to the *WISPCam*. Finally, the *WISPCam* will capture windowed and high resolution images of the faces and transmit them back to the PC. The result of this process is shown in Figure 3.5. The face captured by *WISPCam* has four times more pixel density in each dimensions than the initial 160×120 image.

3.5 Summary

In this article we reviewed the various features implemented on the *WISPCam*, a wireless and battery-free mobile RF powered camera. Numerous applications were demonstrated to address the possible work arounds for the limitations that the *WISPCam* faces. We presented our approach on balancing computational tasks between a host PC and the *WISPCam*. Additionally, the feasiblity of some computationally demanding vision tasks were explored and evaluated. The key achievements on the *WISPCam* so far as an end-to-end system will provide new avenues for future applications, which in turn will make the *WISPCam* a more practical solution for real-life scenarios.

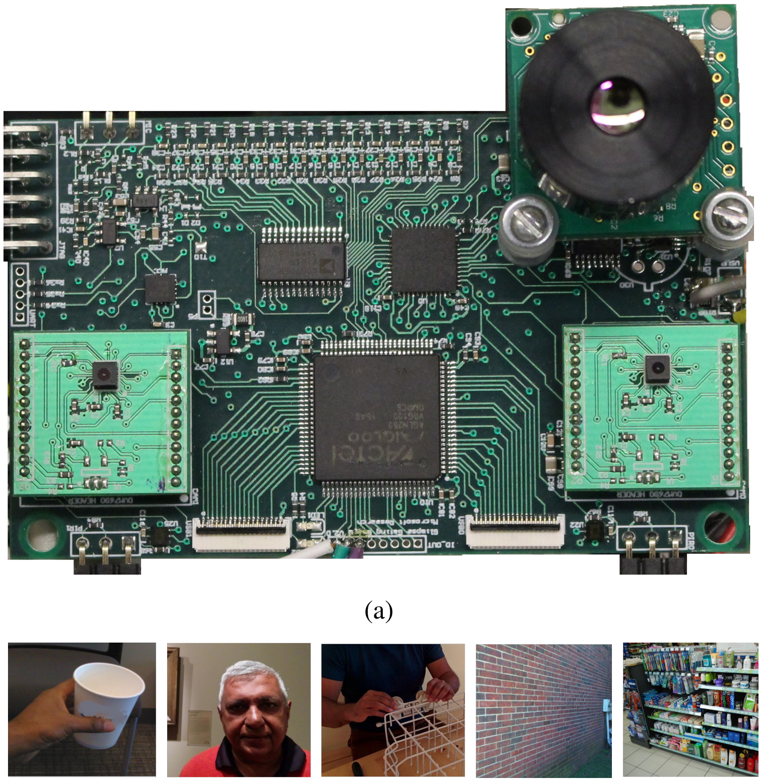
Chapter 4

GLIMPSE: AN EARLY DISCARD CAMERA ARCHITECTURE FOR CONTINUOUS MOBILE VISION

4.1 Introduction

Continuous visual analysis of video from wearable devices has the potential to enable many applications beyond those based on conventional wearable sensors such as accelerometers and GPS [57, 45, 29, 101, 71]. However, modern computer vision techniques, such as those based on Deep Neural Networks, have resource demands that require cloud-based servers rather than mobile processors [47]. The obvious solution, off-loading all computer vision to the cloud, drains the typical mobile battery within two hours, motivating a solution that can inexpensively select a few frames for cloud processing. To our knowledge, no mobile-battery-based solution to date is able to run continuously for a 10-hour day while allowing flexible selection of informative frames. In this chapter, we present Glimpse, a hardware-software add-on to the traditional mobile imaging pipeline that allows frames to be selected for processing *efficiently*, averaging under 100mW over a 10-hour day, and *flexibly*, allowing safe application-level extension.

For continuous functioning, a frame-selection subsystem must address the *resource-accuracy tradeoff* between resource usage, i.e., restricting the execution cost of rejection so it can run for extended periods, and accuracy, i.e., rejecting as many uninteresting frames as possible while preserving the interesting ones. A traditional solution to this problem is to use low-power sensors such as inertial, light [53], time [54] and even skin conductance [51] to trigger frame selection. These techniques score high on the resource-usage axis, since these sensors usually consume orders of magnitude less power than the image processing pipeline. However, in terms of accuracy, i.e. detecting all events of interest, they are limited: many situations of interest are simply not detectable using low-datarate sensors.



(a)

(b)

Figure 4.1: **Glimpse prototype and its applicability.**(a) Glimpse prototype board (size 5 × 8cm) (b) Entities of interest to applications: the wearer's hands [99], their conversational partners' faces [65], their partners' limbs [32], nearby walls [35, 121] and indoor places [102].

At the other end of the spectrum are approaches that use (less expensive) visual processing to trigger more expensive visual processing. Such vision-based triggers can be quite rich relative to sensors, and include, e.g., those based on pixel values (e.g., faces [65]), depth values (e.g., foreground [35]) or temporal variation (e.g., "change" detection [38]). However, because they re-use existing machinery (imagers, algorithms and processors) for visual processing, these approaches are limited by the design assumptions baked into these components:

- **Imagers** The quality of imaging necessary for selection is often much less then that for recognition. E.g., face detection works well at 25×25 grayscale [124] pixel resolution; recognition requires 200×200 color [95].

- **Algorithms** Particular selection tasks often require much less precision than general algorithms provide. E.g., detecting a camera-wearer's hand may only require checking if a pixel is within 1m, whereas a stereo depth system may return cm-level depth for every pixel.

- **Processors** Standard mobile application processors may be energy inefficient because of the mismatch between highly parallelizable image computations and the inherently sequential nature of CPUs. For example, temporal frame differencing for change detection can easily keep a mobile applications processor busy [38], impractical for day-long processing.

To avoid the overhead of reusing the standard vision pipeline while preserving the benefits of vision-based frame selection, we advocate that **vision-based frame selection be treated as a first-class design goal, with a corresponding dedicated hardware/software subsystem in a mobile device**. To this end, we present Glimpse, a system dedicated to discarding uninteresting frames by itself performing coarse visual processing at very low power. Glimpse combines several novel components in a novel way: An array of *gating sensors and imagers* (including a thermal imager and a stereo pair) draws substantially lower power than the primary imager, and is adequate for many selection (but possibly, not deeper analysis) tasks. A suite of *coarse image processing* algorithms is optimized for common early rejection sub-tasks using the gating imagers. A dedicated efficient *gating computation fabric*, comprising of a low-power microcontroller and an FPGA, with efficient access to the gating imagers, is designed to execute (coarse) vision algorithms more efficiently than the standard CPU/DRAM model. A simple programming model, based on *rejection cascades*, allows applications to combine the output of gating sensors into efficient, customized frame-selection classifiers.

Glimpse is fully implemented as a modular circuit board (Figure 4.1a) and embedded software. We tested its frame selection abilities by wearing it for 15 hours over 3 days and programming it to select frames where a new person entered the personal space of the wearer (as a front end for social

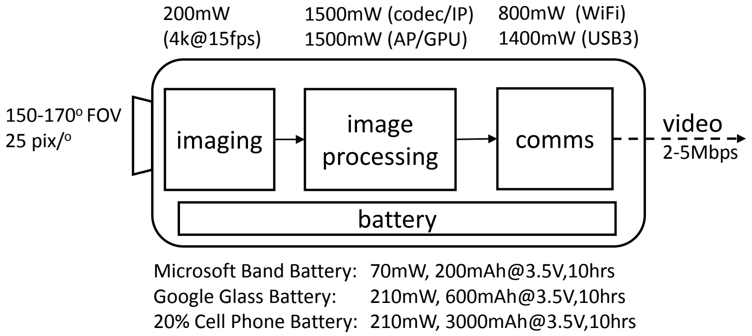


Figure 4.2: **Wearable camera pipeline.** Day-long continuous operation off a band/wearable-sized battery is infeasible with the conventional wearable camera systems pipeline.

assistance [77, 65]), and those where the wearer's hand came in the field of view (as a front end for activity tracking applications [128, 99]). Glimpse is able to detect frames representing events of interest over 87% (100%) of the time for visual events longer than 1s (3s) while drawing roughly 41-54mW for frame selection and 43-153mW in total. These represent a 10-20× improvement over state-of-the-art mobile application processor based implementations. We believe our work is a significant step toward day-long, programmable, continuous vision on wearable-device energy budgets.

4.2 Background and Motivation

Figure 4.2 illustrates the structure and key performance characteristics of a wearable camera pipeline comprised of state-of-the art components. The basic pipeline consists of a high-resolution, large field of view color imager[1] feeding into a compression and/or image processing subsystem. The resulting video stream is either stored on board (not shown) or transmitted using either wireless or wired communications ("comms") module. In the figure, we detail the power *consumption* of each component above the camera, and also specify the average power *budget* of the entire camera, under varying battery assumptions, below the camera.

We now consider two configurations. In the "full offload" model, we could compress the result-

---

[1] A large field of view is critical to wearable devices because interesting events occur in a wide area in front of the user.

ing frames into a video stream and offload it to, e.g., the cloud. Compression, via a codec/image processor (IP) combination, would require at least another 1.5W on top of about 200mW high-resolution camera power consumption [2], and the resulting stream would require a steady 800mW or more to offload via WiFi (and considerably more via WWAN). In the "selective offload" model, we could use an application processor (AP) to detect and transmit frames of interest. Keeping up with high-resolution 15fps would easily require multiple cores, including GPU support [69], and consume 1.5W or more. In this latter model, we would hopefully need to transmit frames only a small fraction of the time (e.g., 10% of the time), so that transmission power in this case would be lower, say 10% of 800mW, say 80mW. In either model, average power draw from the camera would exceed 2W. These numbers are in line with recent reported mobile vision systems [38].

This level of power consumption is problematic for two reasons. First, as the lower part of Figure 4.2 shows, most reasonable battery configurations will only yield at most roughly 200mW. Thus, if the wearable device is to be powered by either a compact battery (a la Google Glass) or by a fraction of a cell-phone battery (a generous 20% of the battery), a constant power draw of 2W is roughly 10× more than is sustainable. Second, for wearable devices of the Glass form factor, a reasonable rule of thumb is that each Watt of power dissipated raises the temperature of the device by 10°C [69] above its surroundings. The resulting 20°C (36°F) temperature rise is quite noticeable and potentially unpleasant for wearable devices. Thus even if it is feasible to power the entire camera via a wire connected to a large battery pack, e.g., police cameras (this is the "USB" transmit option in the figure), significantly reducing device power draw may be important.

4.3 System Architecture

We now present the hardware and software architecture of Glimpse.

4.3.1 Hardware Architecture

The primary imaging pipeline, as outlined in Section Section 5.2 and colored gray in Figure 4.3, will typically contain high-power imaging, computation and communications components that con-

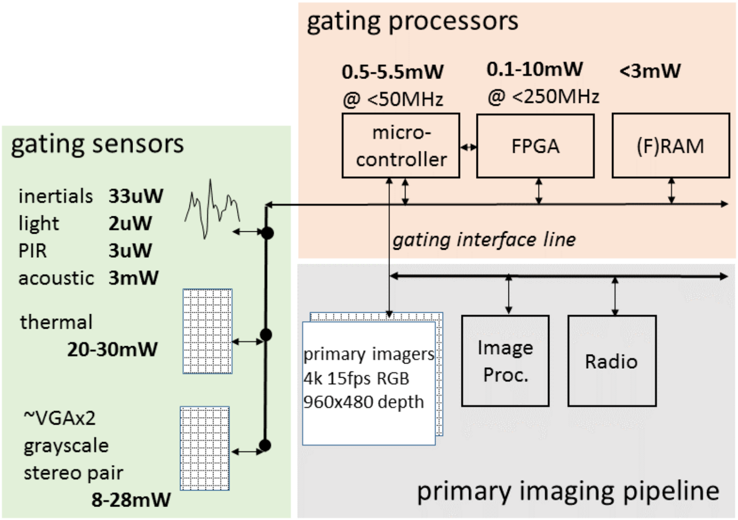


Figure 4.3: **The Glimpse hardware architecture.**

sume an order of magnitude more power than budgeted. Glimpse duty-cycles this pipeline so that aggregate power consumption is below budget, implying a duty cycle of 10% or less.

The design philosophy of Glimpse is to provide a variety of low-power sensing (crucially, including a variety of imaging options) and processing capabilities that can be combined in a flexible manner to select frames inexpensively for a given application. Glimpse therefore has two primary subsystems (green and orange in Figure 4.3). *Gating sensors* run at relatively high duty cycle compared to the primary imagers and must consume relatively little power. *Gating processors* initialize sensors, control communication with sensors, run application extensions on sensed data, and coordinate task execution with other processors.

Sensors typically run in the 1uW to tens of mW range. These sensors are selected such that, for many applications, they can (individually or collectively) predict whether the primary imagers are likely to detect events of interest to the application at any given time. Note that, crucially, gating sensors are not required to detect the interesting events themselves. Thus, for example:
The **inertial sensor** (accelerometer and gyroscope) may indicate that the velocity of the camera is high enough that motion blur will make it unlikely that faces, or other specific objects, can be detected or recognized.

The **light sensor** may indicate that light levels are not adequate for detection or recognition.

The **passive-infrared** sensor, which is triggered when the thermal signature of the field of view changes, may indicate that it is unlikely that people or their parts are close to the camera in the field of view.

The **microphone**, with accompanying custom analog spectral decomposer for sub-mW acoustic feature extraction, may indicate, based on acoustic activity, that the wearer is in an uninteresting activity state (e.g., driving), or social context (e.g., not in conversation).

The **thermal (or far-infrared/FIR) imager**, which reports the temperature of every pixel in the field of view at 9 fps, may identify the *parts* of the field of view that are unlikely to contain people, vehicles, lights, monitors, etc as per their temperature.

The **VGA grayscale imager pair**, a pair of synchronized low-resolution grayscale imagers, from which a depth map of the field of view may be derived, may indicate that a part or whole of the field of view is too far/close to be of interest.

In every case, these sensors act as gatekeepers to the primary imagers, hence the term "gating" sensors. Good gating sensors have a few key characteristics:

1. Low power draw relative to the primary imager.

2. High information content in terms of predicting, for a variety of applications, the required behavior of the primary imaging pipeline *with high recall*: the gating sensor must seldom falsely imply that the primary imager need not wake up.

3. Low latency in making the prediction, since the primary imagers then need to be turned on to analyze the detected image.

The gating calculations themselves, which determine if a certain application is likely to find the current primary-imager frame interesting, are performed on the *gating processors*. These calculations range from simple digital IO checks (e.g., on motion detection using passive IR sensors) to lightweight computer vision algorithms (e.g., depth from stereo on the grayscale pair and object tracking on a thermal image). Gating processors must run at substantially lower power than the

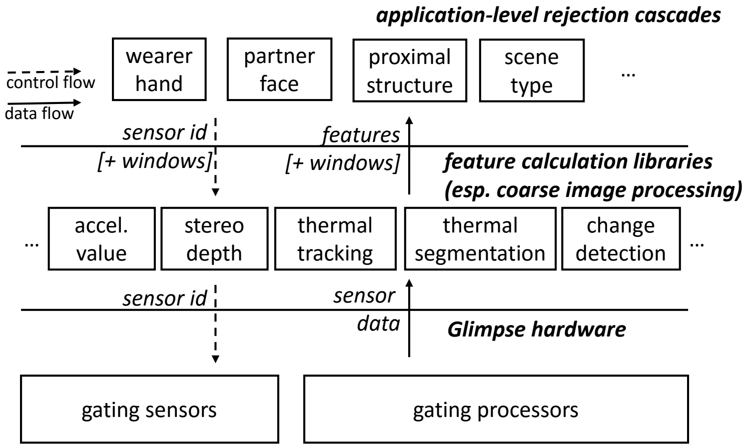


Figure 4.4: **The Glimpse software architecture.**

primary imaging pipeline. Glimpse achieves this by providing a very low-power microcontroller for simple calculations and user-friendly interface and a low power FPGA to handle the vision algorithms, with a small amount (e.g., 1 MB) of static RAM for buffering. The FPGA is also responsible for reading data from gating imagers, which allows FPGA-based vision algorithms to process pixels in a "streaming" manner as they are read off the imager *with no intermediate memory accesses*.

In summary, two aspects of the Glimpse hardware architecture are especially worth noting. First, it advocates the use of multiple possibly unconventional imaging technologies (e.g., thermal imager, low-power grayscale imager and stereo pair) dedicated to frame selection. Second, it incorporates dedicated programmable hardware (i.e., FPGAs) tightly coupled with these imagers for computations related to selection.

4.3.2 Software Architecture

The Glimpse software architecture is designed to allow applications to specify custom frame selection criteria in a simple, expressive and safe manner. Several applications should be able to simultaneously register interest in frames, based on a rich range of conditions without interfering with each other, and without having to write Glimpse microcontroller or FPGA code.

Glimpse takes the perspective that frame selectors are essentially *cost-sensitive classification* *cascades* [124, 129]. As such, they can be divided into relatively heavyweight *feature calculation libraries* and lightweight *rejection cascades* (Figure 4.4), which applications can safely chain together to make increasingly expressive but efficient classifiers as described below.

Feature calculation libraries

Feature calculation is performed exclusively by optimized native libraries. Features range from simple wrappers around sensors like accelerometers to versions of visual primitives such as depth estimation, tracking, segmentation and change detection. Only feature calculators are allowed to interact directly with sensors. Feature calculation is optimized both via the design of new "coarse" versions of vision algorithms (discussed in Section 4.4), and also via careful implementation of these algorithms that, for instance, exploits the underlying hardware such as the FPGA.

Each feature calculation function is required to take as input a set of windows partitioning the field of view of the primary imager and return a set of windows, each annotated with a vector of values. For instance the "accelerometer" value feature calculator may ignore the contents of its input window and simply annotate it with system acceleration. On the other hand, the thermal segmentation function, given a window, returns a set of sub-windows along with the temperature for each window. The depth estimator annotates each of its input windows with the (coarse) depth of that window.

Interpreting rejection cascades

Applications are allowed to define *rejection cascades*, small programs that define classifiers. Glimpse provides a simple embedded *cascade interpreter* to execute the cascade and notify the primary imaging pipeline.

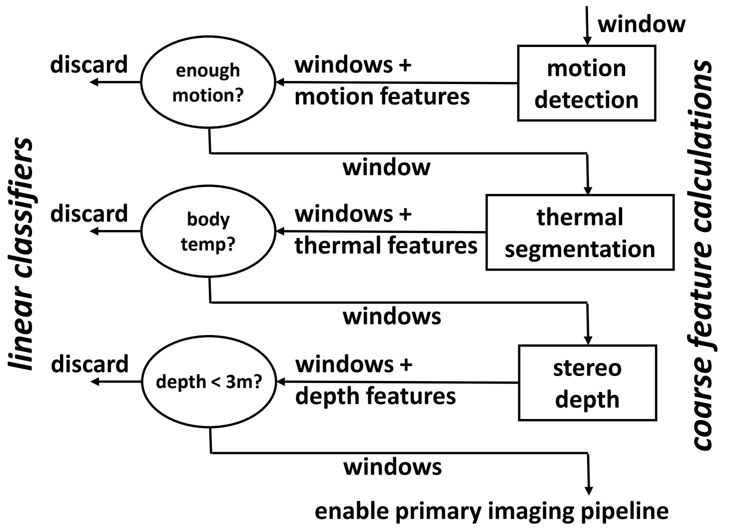
A cascade is a list of triples $(f_i, \hat{w}_i, t_i)$. Each triple consists of a reference f to a feature calculator, a vector $\hat{w}$ of linear-classification weights and a single scalar threshold value t. Given a set X_{i-1} of windows from the previous step of the cascade (the first step always starts with a single window that encompasses the entire field of view of the primary imager), the interpreter applies feature calculator f_i to produce a new set $X_i' = f(X_{i-1})$ of candidate windows and corresponding feature vectors V_i. For each candidate window $x_{ij}' \in X_i'$ and corresponding feature vector $\hat{v}_{ij} \in V_i$, the interpreter includes window x_{ij}' in set X_i for processing in the next cascade step if $t_i > \hat{v}_{ij}\hat{w}_i$, and rejects it otherwise. If, at any step, no windows remain to be forwarded, the interpreter stops. After the final stage of the cascade, if any (non-rejected) windows remain, the interpreter notifies the primary imaging pipeline to select a frame.

In Figure 4.4, solid arrows represent data flow, and dashed lines control flow, during interpreter execution.

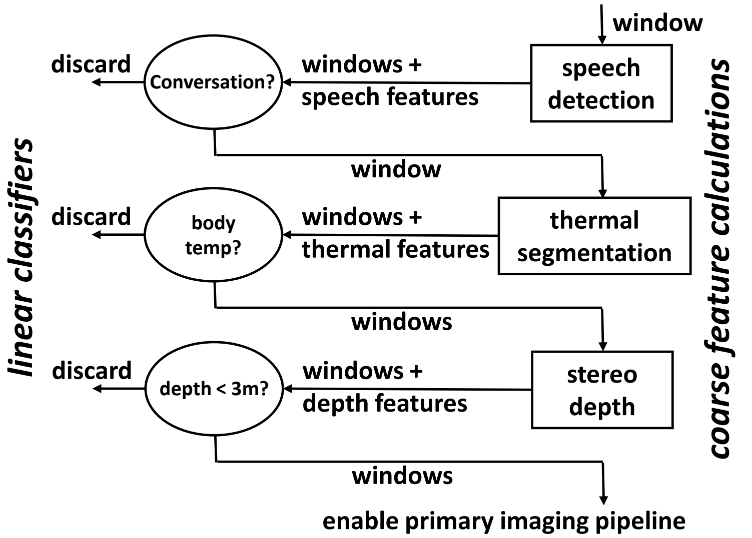
Sample cascades

Figure 4.5 illustrates two rejection cascades. Figure 4.5a shows a cascade that triggers whenever either the wearer is interacting with a human or an object in his hand. This requires existence of human body parts closer than a certain distance (here set to 3m) from the wearer. It checks a motion-detector (typically passive infrared based) to check for adequate thermal activity in the field of view (either due to background or the wearer motion). If not, it discards the frame. If so, it uses thermal imager output to look for sub-windows in the frame that are at body temperature; if none are found, it rejects the frame. Finally, it checks the depth of the surviving sub-windows, retains any within 3 meters of distance and rejects the rest. If any windows survive, the primary imaging pipeline is enabled. Feature extractors that are inexpensive are expected earlier in the cascade, and more informative, and usually more expensive ones, later. Changing the depth threshold from 3 meters to 1 meter will configure Glimpse to trigger only when the wearer's hand is in the field of view (e.g., detecting if the wearer is manipulating objects). These two cascades are evaluated in Section 4.7.

Similarly, Figure 4.5b illustrates another cascade that programs Glimpse to trigger whenever the wearer is in a conversation with another human. The steps are identical to the first cascade except here we replace "motion detection" step with "audio detection" which uses a low-power microphone and a microcontroller to detect if there is a conversation happening. A noteworthy point about this cascade is that the "speech detection" feature calculation library function consists



(a) Human + hand detection



(b) Conversation partner detection

Figure 4.5: **Sample cascades.** 4.5a shows a cascade that detetcts if the wearer is either interacting with a human or with an object in his hand. 4.5b shows a cascade that triggers whenever the wearer is in conversation with another person.

of a simple linear classifier coupled with a hardware implementation of spectral features suitable for voice classification tasks [40, 75], yielding useful classification at the 1-mW power level.

We believe the Glimpse architecture attains a useful balance of simplicity, flexibility and safety. By providing optimized libraries that perform most of the heavy lifting, Glimpse makes programming simple. By allowing applications to nevertheless combine a variety of feature libraries in different orders, and by allowing early exits from processing these libraries, Glimpse programs stay expressive and fast. Finally, since applications can only specify linear orderings in which features may be calculated, it is relatively simple to bound the runtime cost of execution of cascades. In particular, because rejection cascades contain no application-level loops, their safety is relatively easy to ensure.

4.4 Coarse Image Processing

Early-discard systems such as Glimpse have different accuracy requirements from primary imaging pipelines. In particular, primary imagers must discriminate between several classes of interest, and achieve low false-positive *and* false-negative rates for each class. Early-discard systems, on the other hand, typically discriminate between a "background" or "uninteresting" class and interesting ones, often a task requiring less discriminatory power. More importantly, because the primary pipeline will re-process their results, they are only required to have low false negative rates (i.e., they should rarely reject interesting frames) and are allowed modest false-positive rates: as long as they let through only a small number of frames, it is acceptable for some large fraction of those frames to be uninteresting.

Glimpse exploits these less-stringent accuracy requirements by providing, via feature calculation libraries, *coarse* variants of several vision algorithms that are optimized for low false-negative but modest false-positive classification. We have found that *it is feasible to approximate many existing vision algorithms by systematically relaxing traditional computer vision design choices* such as distance metrics, accuracy, data representation bitwidth, output precision, trading off global for local search and "giving up" conservatively on hard cases. In the other words, we show that our approximated algorithms are not only more power efficient than their fine-grained variants, but

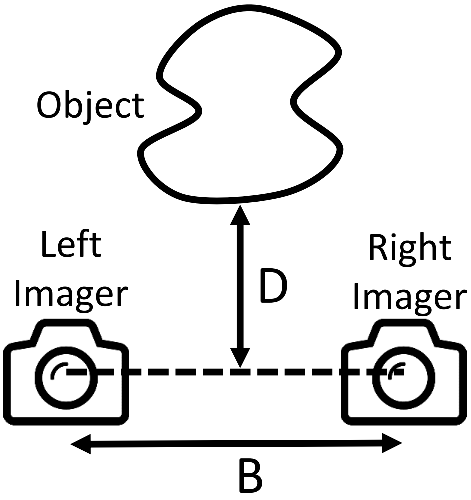


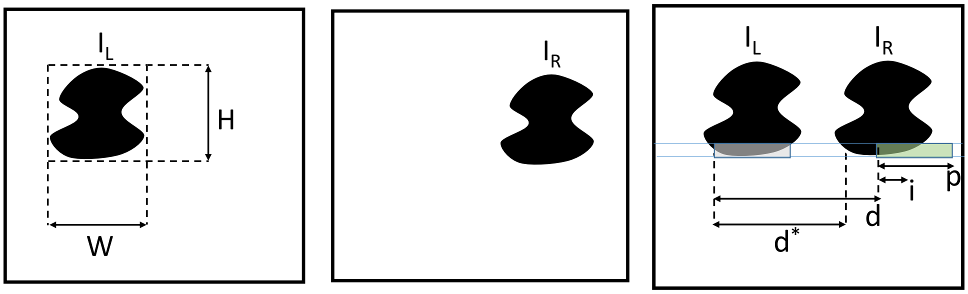
Figure 4.6: **Stereo vision setup.** An object at distance D in front of a stereo pair of cameras.

they also provide satisfactory accuracy when coarser levels of granularity are sufficient. Below, we present two detailed case studies where we use such coarsening techniques, stereo-based depth estimation, and tracking in thermal video. We validate our design choices via measurement studies in Section 4.6.

4.4.1 Coarse Stereo on FPGA

As illustrated in Figure 4.7, traditional stereo takes as input a pair of synchronized images ("left" and "right" images, Figures 4.7a and 4.7b) captured by two aligned imagers ("Left Imager" and "Right Imager" in Figure 4.6) at a fixed baseline distance B from each other. When the left and right images are superimposed (Figure 4.7c), the image I_L of a given rigid object in the left image is displaced from its image I_R by an x-distance of d^* pixels. The actual depth D of the object is inversely proportional to d^*, so that finding the displacement d^* between corresponding patches in an image (the "correspondence problem") is the central computation in stereo vision [50].

For each original patch at pixel (x,y) in the left image, say the gray length-p patch in Figure 4.7c, the check is performed by iterating over every length-p *candidate* patch in the corre-



(a) left image (b) right image (c) superimposed

Figure 4.7: **Components of stereo matching.**

sponding row of the right image (say the relative offset, or *disparity* of the new patch is d pixels), calculating a patch difference metric Δ between the candidate patch (e.g., the green one in Figure 4.7c) and the original one, and finding disparity d^* that minimizes Δ:

$$d^*(x,y) = \arg\min_{d} \Delta(I_L(x : x+p,y), I_R(x+d : x+d+p,y)) \tag{4.1}$$

We optimize this calculation in several ways to reduce power and real-estate consumption on our FPGA. Our goal is a *coarse* depth algorithm that (a) only reports a few discrete depth levels (e.g., 1m, 3m and ∞), and (b) only detects objects greater than a minimum size up to a minimum distance (e.g., size 10cm up to 3m).

Relaxed Distance Metric

Glimpse relaxes the distance measure Δ. The standard measure sum-of-squares (SOS). We instead use the sum of absolute difference (SAD) [66]:

$$\begin{aligned} &\Delta(I_L(x : x+p,y), I_R(x+d : x+d+p,y)) = \\ &\sum_{i=1}^{p} |I_L(x+i,y) - I_R((x+i+d),y)| \end{aligned} \tag{4.2}$$

SAD can be implemented in FPGA fabric using a subtraction followed by a comparison, and is significantly less complex than SOS, which requires multiplications.

Buffer-Free Microarchitecture

Most imagers output the pixels of each frame row-by-row. Traditional stereo algorithms read the entire frame out to memory (typically DRAM) and then iterate on this buffered data when solving Equation 4.1. Such a buffered implementation requires almost continuous access to on-board DRAM, adding a significant overhead to the stereo subsystem power draw. To eliminate the overhead of buffering in memory, we implement the correspondence algorithm in a streaming fashion.

The key observation is that finding the correspondence between two rows of spatially separated pixels translates to finding the correspondence between delayed variants of the pixels as they stream out from the imagers. Figure 4.8 shows the hardware architecture of the depth calculator design. This is composed of three HW building blocks: delay, computing absolute difference between pixels, and accumulator, which adds in and stores the running sum.

Delays are used to shift the pixel stream coming from the left image sensor by the various disparity levels d we wish to try. The delayed pixels from the left image are differenced with the pixels from the right image. Each accumulator then aggregates the results and outputs the measure of similarity for the two image windows associated with a particular delay length. The optimal disparity for a particular window is the delay value which minimizes the similarity measure.

An important implication of the above design is that every iteration of the minimization loop over d in Equation 4.1 has corresponding circuitry on the FPGA (i.e., the loop is unrolled). Further, within the accumulator, the area of SAD circuitry itself is proportional to the bitwidth of the pixel values being SAD'ed. In addition to avoid buffers, this design has the advantage that these operations can be parallelized, but poses the challenge of requiring additional real-estate.

Low-Bitwidth Pixel Representation

Most off-the-shelf cameras use 7 or 8 bits to represent monochrome pixels. We hypothesize that *if only coarse depth estimates are required*, it is possible to truncate the least significant bits (LSB) of the pixels $I(x,y)$ while introducing little error. Note that a few bits of truncation makes the

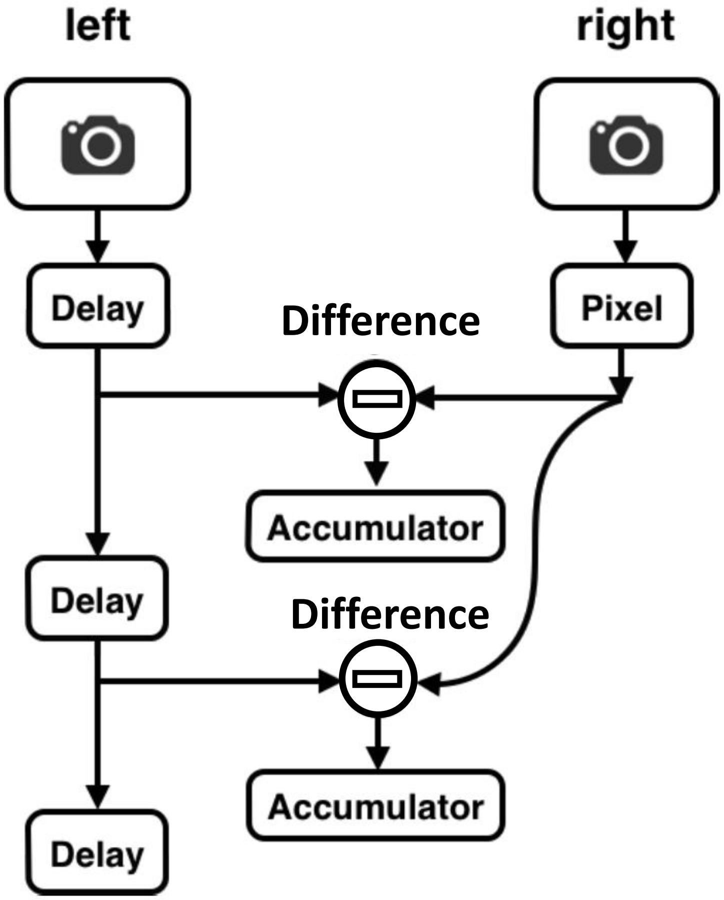


Figure 4.8: **Microarchitectural support for buffer-free disparity checking.**

resulting picture indistinguishable to the human eyes from the original. On the other hand, in an FPGA implementation, reducing the bitwidth of numbers manipulated in the innermost loop yields roughly proportional reductions in gate count and power at little reduction in accuracy of coarse depth estimation (Sections 4.6.1 and 4.6.2).

Sparse Disparity Checking

Equation 4.1 moves the candidate patch to every possible disparity d with respect to the original patch. In our subsampled VGA image, we must consider $n_d = 60$ different disparity values. The power consumption (and in our FPGA implementation, the circuit size) is directly proportional to

n_d. Reducing n_d could therefore substantially improve matching cost.

The intuition behind considering every possible disparity d is to find the d^* that absolutely minimizes the patch difference Δ. However, if we are willing to settle for a difference d^+ that is "close enough" to d^*, we could possibly get away with testing fewer disparities. An initial try may be to shift the candidate patch by $k > 1$ (instead of $k = 1$) pixels when searching for d^+. This brings up the question of how big k can be. If it is too big, we run the risk of not detecting objects smaller than k in image space. If too small, we lose efficiency.

Our approach is to require only that objects of width $\geq W_{min}$ will be detected. For instance, we may specify that the object must be at least the size of a face, e.g., $W_{min} = 25$cm. It is well known [50] (and simple to verify via high-school geometry) that the width in pixels W_D of an object's image is linearly related to the disparity d^* between its left and right images: $W_D = ad^* + b$ for constants a and b. Intuitively, the closer the object to the stereo pair, the larger the disparity d^*, and the larger the size of the image, so that the disparity and image size vary together. Now if we only require that the candidate patch *overlaps* by a factor of ω to its corresponding image from the other imager, as d increases, the patch expands linearly so that the set of pixels at which the candidate patch can be placed expands: the gap k between positions of the candidate patch (i.e., values of d) can increase with d instead of remaining fixed.

In Glimpse, we fit the values of k greedily using data "labeled" by the brute-force $k = 1$ algorithm. Instead of checking 60 disparities for every original patch, we settle on checking 18 disparities d: $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 13, 15, 17, 20, 23, 26, 30, 34, 39, 44, 50, 57\}$ (thus k increases from 1 to 7), a 70% reduction in matching overhead and circuitry. We require overlap $\omega = 0.95$, and require faces to be detectable: $W_{min} = 25$cm. Section 4.6.1 shows that sparse checking is accurate when applied to coarse depth estimation.

Depth Quantization

Coarse stereo introduces noticeable error in depth estimates. This is problematic if applications rely on the coarse depth estimation module for fine-grained depth information. For instance, assume coarse depth estimation may introduce an error of 0.3m when the object is about 5m away. If an application were only interested in objects in a band 5.0-5.1m away, using coarse depth to select frames for this purpose could result in mistakenly ignored objects.

To remain conservative, i.e., to maintain good recall, Glimpse's coarse depth API only reports depth in a limited number of bins (e.g., less than 1m, 1-3m, 3-10m and beyond 10m). Given disparity d^+ we derive the corresponding object depth D^+ and quantize it into these bins. The additional quantization circuitry is minimal. The big impact, as we show in our evaluation (Section 4.6.1) is that although the optimizations described in the previous subsection introduce noticeable error in *raw* depth estimation, they are usually well within the slack provided by the *quantized* bins.

4.4.2 Coarse FIR-Based Tracking

We now turn to functionality based on the thermal (or Far Infra-Red (FIR)) imager, a sensor not broadly available on camera platforms. FIR imagers have lower resolution (16×16-80×80 pixels are common) than RGB, but provide temperature at every pixel. In principle, therefore, they can often directly detect the presence of exceptionally cool or warm bodies (e.g., humans, vehicles, machines, monitors, lighting fixtures) in the environment. One simple but useful feature calculator based on the thermal sensor that Glimpse therefore provides, is a (time-averaged) temperature-based segmentation module, which detects boxes in the field of view with temperature above or below a specified threshold.

In this section, however, we present a related feature calculator, a *tracker*, implemented on the Glimpse microcontroller, given the relatively small volume of data from the FIR imager, that detects and tracks across frames objects in a particular temperature range. Tracking is especially important in early-discard systems such as Glimpse because for many applications, once an object of interest is detected in the field of view, it ceases being of interest. For instance, an application may wish to be notified of a new face coming into the field of view (it may capture a high-resolution frame of the face using the primary imager for further analysis), but once the face is identified it may wish to ignore it.

Glimpse allows high-level applications to notify it to ignore the entity in a particular window. Glimpse does so by tracking the window and not enabling the primary imager on re-detecting it, even if it is otherwise interesting. Prior work [38] has noted a similar benefit in tracking on mobile devices to avoid uploading to the cloud.

Traditional Tracking in RGB

At a high level, tracking in RGB video works as follows [93, 49, 100]. Suppose a window w_t of frame f_t is to be tracked. Given the next frame f_{t+1}, the goal is to calculate window w_{t+1} in f_{t+1}, such that w_{t+1} matches w_t in appearance and dynamics. To perform this match, the tracker may recursively maintain velocity v_t of w_t, hypothesize that the new location of the window is $w'_{t+1} = w_t + v_t$, and search the vicinity of w'_{t+1} for the window w_{t+1} with the best appearance match to w_t. Then the velocity gets updated and the whole process repeated with the next frame, f_{t+2}.

A key design choice is the manner in which the search for the window with the best appearance match is performed. Typically (e.g., [93]), the tracker identifies a set of *keypoints* (locations) in w_t. Each keypoint is associated with a *descriptor* that encodes its surroundings in an illumination (and often scale) invariant manner. The tracker then identifies all keypoints and associated descriptors in f_{t+1}, and matches the old descriptors to a subset of the new ones to determine an appearance match to w_t. Keypoints and related descriptors are relatively computationally intensive to find because they require scanning an entire (relatively) high-resolution image, and need to be illumination invariant since illumination can change significantly across frames in RGB video.

A second choice that adds to the computational overhead of conventional tracking is the manner in which multiple possible matches are resolved, especially when tracking multiple objects. Resolving this ambiguity usually requires some amount of global reasoning [100] possibly including geometry-based consensus [93]. Below, we describe designs for temperature-based keypoints; descriptors that are simple, but fast and adequately accurate by conservatively and quickly handle ambiguities in matching.

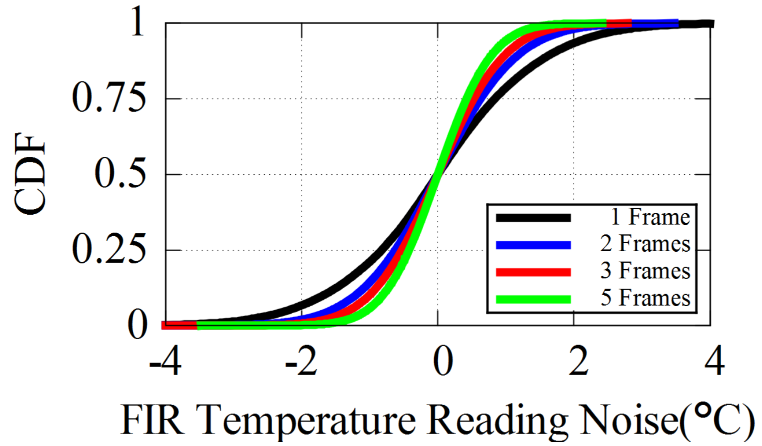


Figure 4.9: **White noise on the FIR camera's pixel reading.** CDF of noise at each pixel of FIR camera for different frame averaging lengths.

Temperature-Thresholded Keypoints

An advantage of FIR imaging is that it is fairly robust to the environment, unlike RGB imaging. The measured temperature of a point does not change due to change in direction of viewing, illumination sources in the environment, shadows, etc. Given that the tracker is intended for use to track objects that have temperature T significantly different from the background, Glimpse therefore uses the "raw" temperature to find candidate locations for keypoints: all pixels in some temperature range $T \pm \delta$ (we pick $\delta = 3°C$ currently) are considered potentially matching keypoints. Given that the FIR imager has only 1024 pixels (significantly less pixels than a regular RGB camera), keypoint generation is extremely fast.

One complication is that FIR imagers are prone to (zero-mean Gaussian) white noise [41]. We therefore average across $n = 3$ frames to reduce the temperature variance. Figure 4.9 shows CDF of pixel reading noise from a fixed scene when averaged over frame sub-sequences of varying lengths. It is clear that no averaging (black curve) will result in about $\pm 4°C$ of noise, but a moving average over three consecutive frames will reduce the generated noise to about $\pm 2°C$. Although averaging over five frames reduces the noise even further, the resulting blur is unacceptable for our purposes.

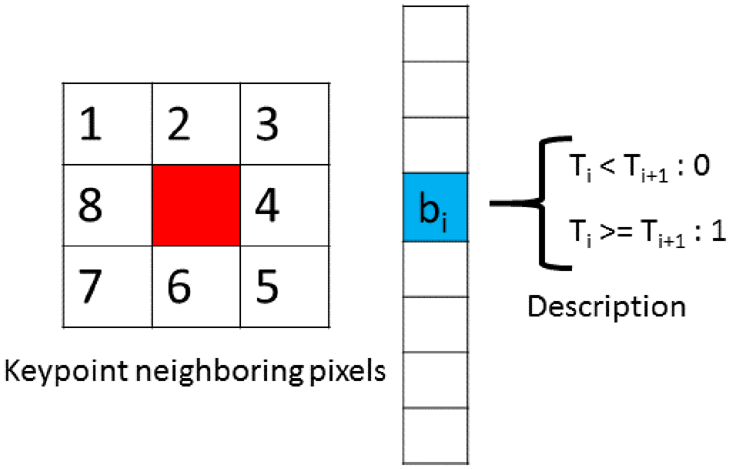


Figure 4.10: **8-bit binary spiral descriptor.**

Temperature-Spiral Descriptors

Given keypoints, we seek to associate descriptors with them in order to determine which, if any, match with descriptors from the previous frame. To ensure illumination, rotation and scale invariance, traditional descriptors are fairly computationally intensive, involving multiple rounds of gradient, histogram and multi-scale smoothing calculations [39, 74].

For our descriptors, we simply choose an encoding of the temperature values around the keypoint arranged in a spiral. We present four different encodings/descriptors here and evaluate their performance later in this chapter. The descriptors, detailed below, are **1.** 8-bit binary **2.** 24-bit binary **3.** 8-point absolute temperature and **4.** 24-point absolute temperature.

Figure 4.10 shows the 8-bit binary descriptor. The keypoint (red pixel) and its neighboring pixels are shown in the Figure 4.10. This descriptor is an 8-bit number such that each of its bits (b_i) represent the logical result of comparing temperature at pixel i and $i+1$. Similarly, 24-bit binary descriptor follows the same procedure with the difference that we consider two layers of circular neighboring pixels around the keypoint instead of just one. The 8 and 24-points absolute temperature descriptors are similar to the binary versions with the difference that the i^{th} element of the descriptor vector is the temperature of the i^{th} neighboring pixel of the corresponding keypoint.

Finding the best match is standard: we use the sum of absolute differences to compare absolute temperature descriptors and using bitwise-xor for binary descriptors.

Losing Track Conservatively

A limitation of FIR imaging is that its resolution is low, and different instances of objects often have indistinguishable thermal signatures. For instance, it is infeasible (for even humans) to tell one person from another using currently available low-power thermal imagers. The problem of resolving ambiguities between descriptors from multiple objects is therefore especially challenging in FIR tracking.

Our solution is simply to be conservative about matching. Applications will be notified if there is a potential break in the track and it is up to the application to stitch together tracks, potentially using higher level information such as RGB pixels.

To implement conservative matching, at each time step, Glimpse first expands each keypoint in the new frame f_{t+1} into a "blob" (i.e., connected component of similar temperature values) surrounding it by iteratively thresholding on pixel values, a simplified version of Otsu's segmentation method [94]. If any pair of blobs in the new frame are "too close to each other", we terminate the track.We determine if two blobs are close to each other by dilating them by a single pixel [2] and checking if the blobs merged. Given that iterative thresholding, connected components and dilation are fast operations, the implementation is fast.

4.5 Hardware Implementation

Glimpse is fully implemented as a modular add-on board (4.1a) called the "Glimpse Gating Board" (GGB). The board is designed to run autonomously for a day on a 200mAh battery. Any primary imaging system that has an "image enable" line can be connected to Glimpse via a control line and can communicate with it via a 2 Mbps SPI bus.

---

[2] Note that at 32×32 resolution, a single pixel represents several degrees of spatial separation.

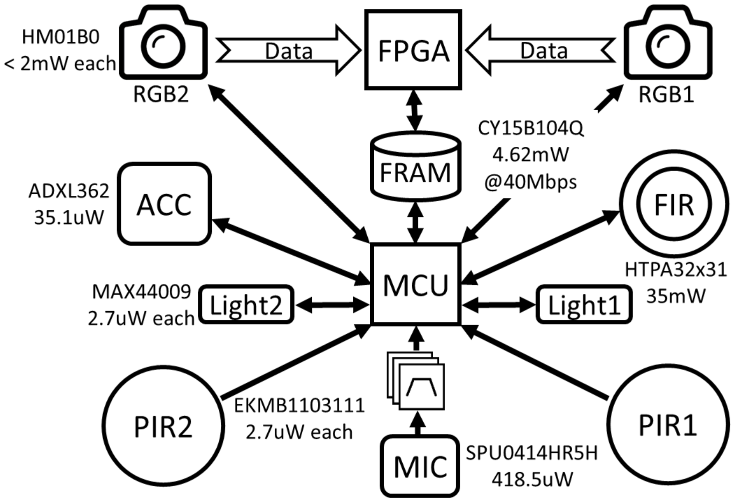


Figure 4.11: **Glimpse gating board implementation block diagram.**

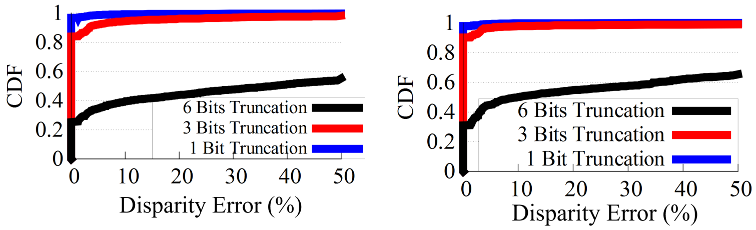
Figure 4.11 shows the block diagram for GGB. Given our small power budget, all components are selected to balance low power while providing adequate performance for coarse, cascaded algorithms. Both the (Ti MSP430) microcontroller and the (AGLN250 Igloo Nano) FPGA are some of the lowest-power parts available in their category. Even a careful implementation of coarse stereo matching (along with video readout circuitry for the two imagers), for instance, consumes half the FPGA. In many cases, power consumption of components will reduce significantly in the near future, tracking advances in commercially available parts. For instance, that the highest power consumer is the (relatively low-resolution) thermal FIR imager, a part ripe for improvement.

4.6 Evaluating Coarse Vision

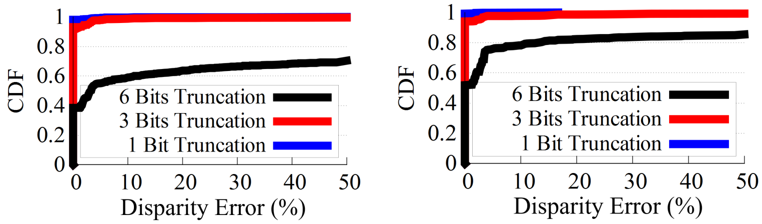
We wish to answer three questions about the coarse vision (depth estimation and stereo tracking) algorithms introduced in Section 4.4:

- What is their accuracy?
- What is their resource consumption?
- How do these vary with our design choices?

We use data collected under controlled conditions in these experiments. We restrict detailed exper-



(a) 20x20 (b) 40x40



(c) 60x60 (d) 100x100

Figure 4.12: **Disparity error versus truncation length.**

imental evaluation to the low-bitwidth representation, sparse disparity checking and depth quantization optimizations of Section 4.4. With respect to the other optimizations, briefly: 1. Using SAD instead of SOS for distance measure (Section 4.4.1) yields a depth measurement circuit that uses 42% fewer gates[3], and 2. Buffer-free implementation of stereo (Section 4.4.1) saves 13% in average power draw over the buffered version.

4.6.1 Coarse Stereo: Accuracy

What is the impact of reducing the number of bits used to represent pixels (Section 4.4.1) on precision of depth measurement? To answer this question, we placed the Glimpse camera in front of a textured wall and moved the camera between 0.5m to 3.9m from the wall, by 0.3m increments. For a set of patches of varying sizes $s \times s$ (s selected from 20, 40, 60 and 100pixels) in the field of view, we calculated the pixel disparity d^* while varying the number b of bits used to represent

---

[3] Power reductions should be similar. The SOS-based circuit is too large for implementation on our FPGA, so we cannot report measured power gains.

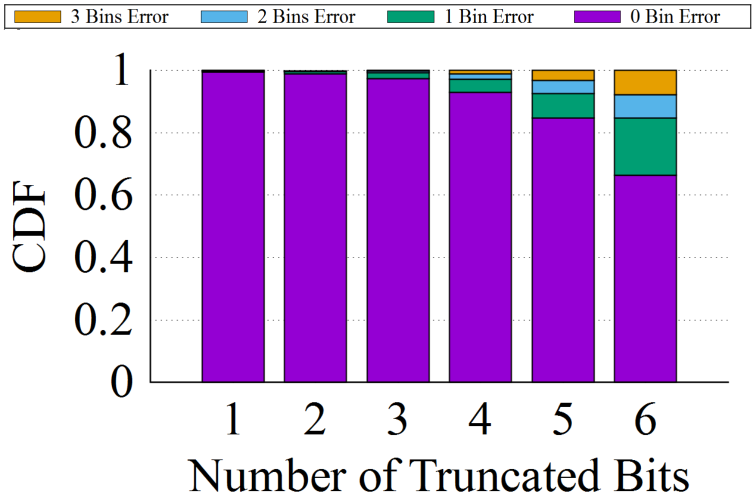


Figure 4.13: **Bin-level depth error vs. truncation.**

the captured stereo pairs. We used b = 7, 6, 4 and 1, corresponding to 0, 1, 3 and 6-bit truncation. Figure 4.12 reports the CDF of error in disparity d^* relative to the disparity measured from the un-truncated stereo-pair version.

Two points are worth noting. First, relative error with a 3-bit truncation (i.e., b=4) is quite acceptable, whereas going down to 6-bit truncation introduces a significant amount of error in disparity estimation. We therefore use 3-bit truncation in Glimpse. Second, although the error depends to some extent on the size of the patch being matched (larger patches contain more visual information and thus less matching error), even small patches can be matched with quite small error with 3 bits truncated. Even with s = 20-sized patches, relative disparity is zero 82% of the time (red line in 4.12(a)) and 94% of the time, it is less than 9%. With s = 60, disparity is zero over 93% of the time.

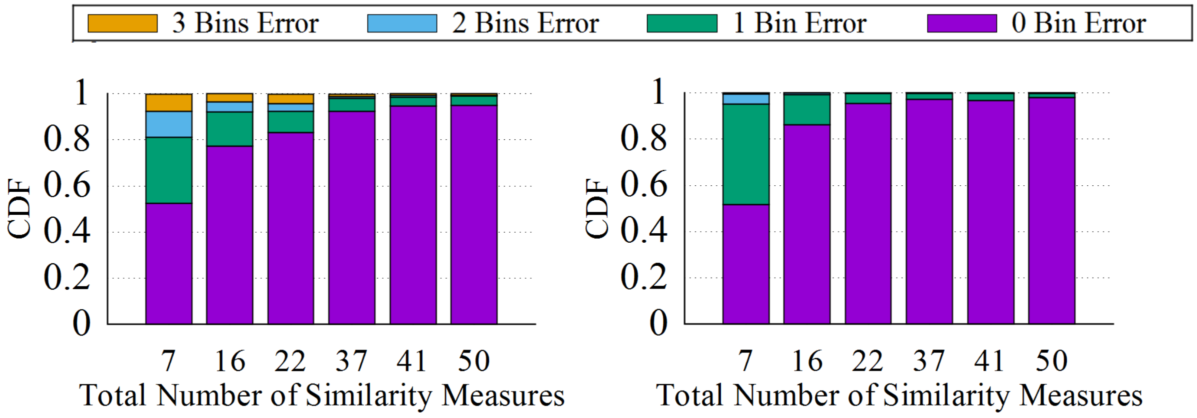
We now study the impact of quantized depth levels on the accuracy of depth estimation Section 4.4.1. Suppose that instead of estimating depth as precisely as possible, we are only interested in knowing which of four bins the depth belongs to: bin_0:0-1m, bin_1: 1-2m, bin_2: 2-3m or bin_3: 3-∞m. We focus now on the challenging 20×20 patches and quantize the results of the above experiment into these bins. In other words, we convert the pixel disparity for each measurement into the actual depth then instead of basing our analysis on the actual depth, we measure the bin index in which it falls into. Figure 4.13 shows the results when 1 to 6 bits are truncated. For each truncation level, the bar shows the distribution of the *bin error*. The bin error is simply the absolute value of the difference between bin index calculated using the truncated and un-truncated data. Focusing on the 3-bit truncation case, we see that *at the bin level*, 96% of the measurements have zero bin error, and almost all the rest are off by just one bin. We also note in passing that even for 6-bit truncation (i.e., just black and white pixels), quantized accuracy is encouraging: $\sim$60% of measurements are binned correctly.

In the above algorithms, we calculated the optimal disparity d^* by exhaustively checking all 60 possible disparities. We now evaluate *sparse disparity checking* (from Section 4.4.1), assuming that a) we are only interested in depth of objects wider than 25cm, and b) we are only interested in bin-level precision. We recomputed bin-level results as in the previous experiment, but now require the algorithm to use just $n \in \{7, 17, 22, 37, 41, 50\}$ different disparity checks, using fixed bitwidth $b = 4$ (i.e., truncation length 3). We compare the binning error relative to using all 60 checks. Figure 4.14 shows results for patches of sizes $s = 20$ and 100. The upshot is that $n \approx 37$ checks are enough for excellent bin-level accuracy over all patch sizes. In Glimpse, because our patches tend toward being larger, we use just $n = 22$ disparity checks.

4.6.2 Coarse Stereo: Resource Consumption

How does resource consumption vary with our approximate implementation of depth detection? Our primary target FPGA is the Igloo Nano AGLN250. However the AGLN250 is not big enough to handle stereo depth detection with full disparity checks and un-truncated pixels. For comparison purposes, we therefore consider the more powerful (and power-hungry) AGL400 and M1AGL1000 FPGAs as well.

In our first experiment, we fixed bitwidth at 7, increased the total number of allowed disparity checks until the AGLN250 ran out of resources, switched to AGL400 and continued increasing the number of disparity checks until it ran out of resources too and finally switched to M1AGL1000. At each point we let the FPGA calculate the depth map over the entire VGA frame which is compu-



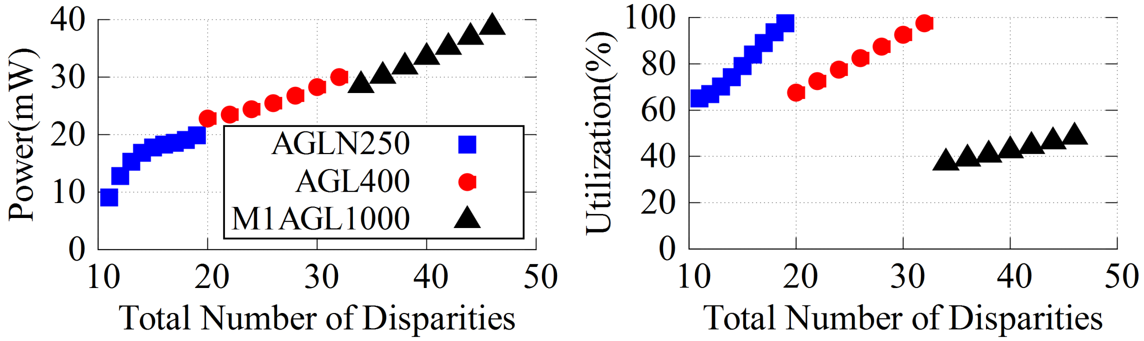
(b) 20x20

(c) 100x100

Figure 4.14: **Bin-level error vs. no. of disparity checks.**

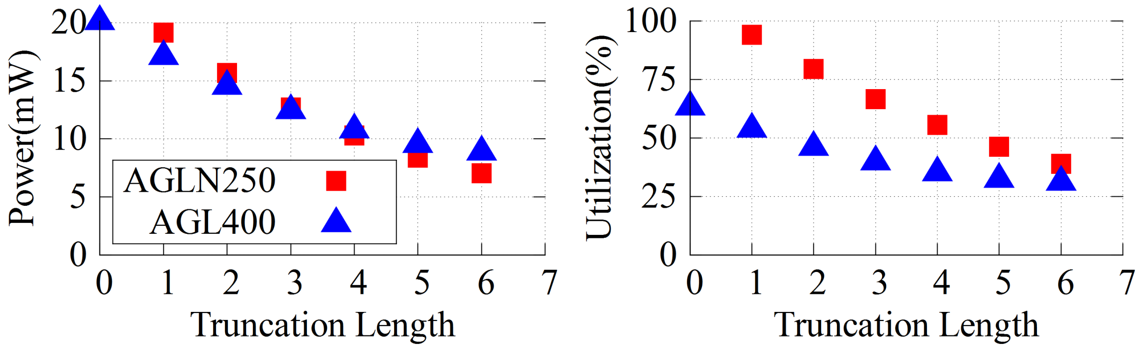
tationally the most demanding version of our depth detection algorithm. We then simulate power consumption and resource utilization of the FPGA using Microsemi Libero SoC software. Figure 4.15a shows power consumption and Figure 4.15b shows resource utilization (i.e. fraction of all available logic elements on the FPGA used) of the three FPGAs under different pixel disparities. As shown in the figures, the power consumption increases generally as the number of disparity checks increases. Also the utilization of a particular FPGA increases as the number of disparity checks increases.

Figure 4.16 examines the power draw and utilization impact of different bit truncation. These experiments are performed with $n = 22$ disparity checks. Resource use drops almost linearly with optimization level (truncation length in this case). Comparing with Figure 4.13 and Figure 4.14, we see that 3-bit truncation and 22 disparity checks is likely the accuracy/power sweet spot, yielding 95%+ accuracy on size-60 patches at a stereo-computation draw of $\sim 12mW$ while using less than 75% of the AGLN250 logic elements.



(a) Power Consumption (b) Gate Utilization

Figure 4.15: **Resource use versus disparity checks.**



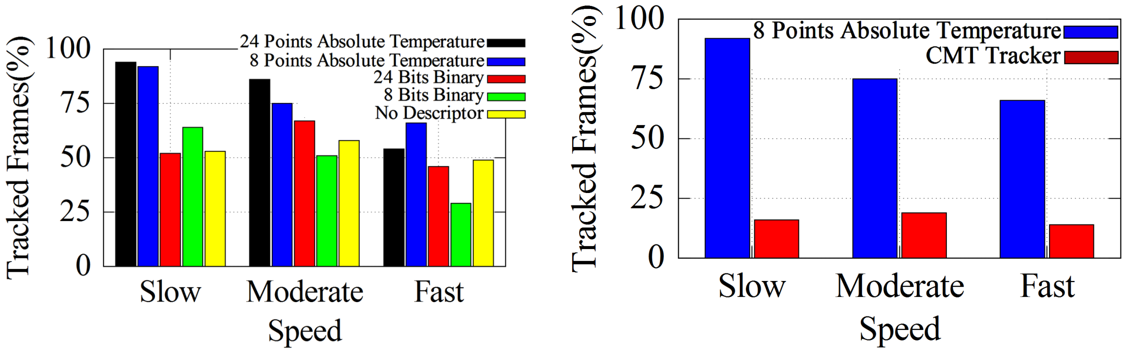
(a) Power Consumption (b) Resource Utilization

Figure 4.16: **Resource use versus pixel truncation.**

4.6.3 FIR Tracking

We now evaluate the coarse FIR tracking algorithm. We try to answer three questions: 1. **Accuracy:** How well does the algorithm keep track? 2. **Resource use:** How much energy does the algorithm consume? 3. **Design choices:** How well do the various spiral descriptors work?

To answer these questions, we performed the following experiment. We pointed Glimpse into an empty office room and asked two human subjects to enter and interact in the field of view in a scripted manner for up to 25 seconds per interaction. We collected 7 versions of the interaction



(a) Comparison of descriptors

(b) FIR vs RGB tracker

Figure 4.17: **FIR tracking evaluation.** 4.17a shows for how many frames our FIR tracker can contiuousely track an object given different descriptors, and 4.17b compares the track length of our FIR tracking algorithm vs a well-known object tracking algorithm when applied to a theremal video.

where subjects were moving at "slow" (leisurely strolling with stops), "medium" (walking) and "fast" (brisk walking, almost running) speeds. We manually annotated the tracks i.e., added distinct boxes surrounding each subject in the video and compared these "ground-truth" tracks to those returned by (variants of) our algorithm.

Figure 4.17a shows how the average fraction of properly tracked frames over total available frames varies with the speed of movement and the various descriptors (in the "no descriptor" case, just the temperature of the central pixel is used as descriptor). The noteworthy points are: First, absolute temperature descriptors are much better than the binary ones; the absolute value of the temperature is clearly a lot more informative than a single bit representation *temperature change* at each pixel. Second, the difference in tracking quality between the 8-point and 24-point descriptors is relatively small, encouraging the use of 8-point descriptors due to their lower computational complexity. Third, given the absolute descriptors, the tracker does an acceptable job of tracking: it maintains track for 60-90% of tracks on average, depending on whether subjects are moving fast or slow. Fourth (Figure 4.17b), we applied a tuned state-of-the-art RGB tracker [93] to the FIR videos to evaluate the importance of FIR-specific descriptors. 4.17b shows that the FIR-specific tracker clearly does better; we found that standard RGB descriptors do not fare well with low- resolution thermal footage.

4.7 End to End Evaluation

Finally, we wish to understand how well suited the Glimpse system is to its primary role of supporting day-long wearable vision. One of the authors wore a Glimpse-based camera for 15 hours over three days of daily activity (roughly 3, 7 and 5 hours each) while running a cascade (Figure 4.5a) on Glimpse. Glimpse produced a high-resolution 30-frame video snippet every time a person came within 3 meters from the wearer (typically a conversation partner) or the wearer's hand entered the Glimpse's field of view. The rejection cascade used is similar to that in 4.5a, except that the thermal segmentation module was replaced by an FIR tracker as in Section 4.4.2. The analysis of another cascade that triggers the primary imager only when the wearer's hand appeared in the field of view is presented in Section 4.7.3.

All video produced by the primary imager was offloaded via wireless to a mobile device. In this experiment we did not attempt off-board analysis of the video. Common activities performed by the wearer included but were not limited to, *driving*, *going for lunch with friends*, *one-to-one meeting*, *group meeting*, *playing ping-pong*, *corridor walking*, *outdoor walking*, and *working in front of a monitor*.

We paired up a Glimpse Gating Board (GGB) with a Raspberry Pi 3 board (which we call *Gated Pi* below) as the primary imager, via a trigger wire. The Pi has an 8MP imager. For the mobile device, the wearer carried a WiFi connected laptop in a backpack; a production system would likely use a mobile phone instead. To collect ground truth, we added a second Raspberry Pi (the *Continuous Pi*) that recorded *all* video in the field of view. Both Pis offloaded data to the laptop via WiFi, one sporadically, the other continuously. We logged *real-time* power draw of the GGB, the Gated Pi and its components to the laptop via a National Instrument data acquisition system in the backpack.

Scenario	PIR	FIR	Depth	Total	Precision
Driving	0.96	0.30	0.43	0.12	31%
Corridor	0.94	0.00	0.33	0.00	100%
Monitor	0.83	0.07	0.34	0.02	8%
Outdoor	0.99	0.23	0.20	0.04	1.7%
Lunch	0.99	0.50	0.34	0.17	42%
Meeting	0.97	0.13	0.23	0.03	12%
1 to 1	0.99	0.09	0.31	0.03	36%

Table 4.1: **Glimpse pass-through rates and precision.**

4.7.1 Gating Accuracy

The primary job of Glimpse is to reject many uninteresting frames while letting through almost all interesting ones. In daily use, what fraction ("pass-through rate") of frames did it allow to pass through? What fraction of these truly should have passed through ("precision")? What fraction of all interesting events actually went through ("recall")? We answer these questions in this section.

Our detailed power measurements allowed us to monitor in real time when each component of each cascade triggered, and whether or not the Gated Pi was finally triggered. We were therefore able to measure the pass-through rate of each stage of the rejection cascades, and of the two cascades as a whole. Table 4.1 reports the results, broken down per scenario. For each cascade stage, we report the fraction of *its* input passed through.

Overall, total pass-through rates range from 0 to 17%, with an average (not shown) of roughly 6%. The ultra-low-power PIR can reject 1-17% of input: it is especially good at detecting inactivity when sitting roughly still in front of the monitor. Both FIR and Depth are highly effective gates. When walking in the corridor on these three days, the wearer encountered just two persons, for which the FIR stage let through 3 frames, which rounds off to a 0.00% pass-through. The Depth system allowed one of these frames, for a 1/3 pass-through rate.

Event Duration, d (s)	$d \geq 3$	$3 > d \geq 2$	$2 > d \geq 1$
Recall Rate (%)	100	96	86.5

Table 4.2: **Glimpse recall rate vs event duration.**

Scenario	Total	Primary	GGS	Depth	FIR
Driving	116	67	49	6	38
Corridor	43	0.6	42	0.2	37
Monitor	80	39	41	1.4	33
Outdoor	80	30	50	5.1	40
Lunch	153	99	54	10	39
Meeting	85	39	46	2.8	38
1 to 1	73	26	47	2.2	39

Table 4.3: **Glimpse power consumption (mW).**

We inspected all passed through frames manually to determine what fraction had wearer hands or nearby people i.e., the precision. As the “Precision” column shows, Glimpse’s precision is modest. Again, since the one frame let through in the corridor was indeed a person, precision is 100%, even though the overall pass-through rate rounds to 0.00.

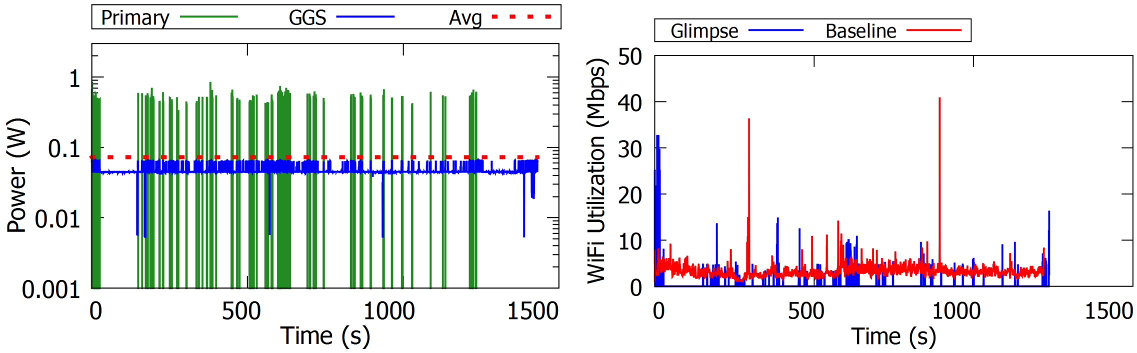
Measuring recall is harder, since we need to inspect roughly 1.6M frames from Continuous Pi to determine if we missed any events. We looked through the video manually and counted instances where a hand or person appeared in the field of view for duration d seconds, such that $d \geq 3$, $3 > d \geq 2$ and $2 > d \geq 1$. We found the shorter the event, the more episodes there were when the object moved so fast through the field of view that GGS, at 10FPS, was unable to detect it. For events of modest (e.g., > 2s) length, however, Glimpse has excellent recall (Table 4.2).

4.7.2 Resource Usage

What is Glimpse's average power and wireless bandwidth consumption? The low overall pass-through rate (6%) indicates that power consumed by the primary imaging pipeline, and data sent by it, should be correspondingly low. Further, by design, the Glimpse board itself should have low power draw. How do these play out in practice?

Table 4.3 shows the average power (mW) breakdown for the major components of GGS, that of Gated Pi and their sum. Overall average for the total power consumption (not shown in the table) was roughly **112mW**. 4.18a shows the real-time power traces for Glimpse system in blue, Gated Pi in green, and their overall power consumption in red. This curve illustrates that power consumption of the Gated Pi becomes bursty in the existence of Glimpse system in a sample scenario of one-to-one. For comparison, the average power draw of Continuous Pi, which off-boards all data is **1016mW**. A few points are worth noting. First, Glimpse *total* power consumption is low both in absolute and relative terms. In relative terms, it is **7x** to **25x** lower than pure offloading, depending on the scenario. In absolute terms, its *worst-case* 153mW draw will allow it to run for over 5 hours on a tiny 200mAh battery; the average of 112mW would allow almost 7 hours. Second, the gating board itself (average draw 55mW) can easily be powered by a 200mAh battery through a day. Third, the FIR sensor and tracker is a power bottleneck, with the sensor itself averaging well over 20mW.

Finally Table 4.4 shows savings in bandwidth utilization. Figure 4.18b shows intermittent WiFi transmissions of Gated Pi when triggered by Glimpse system in a sample scenario of one-to-one meeting. Even when Gated Pi transmits a one-second burst of video every time it is triggered, it still sends 4× to 16× less data. Our measurements show that if it only sent one frame each time it was triggered, data sent would reduce by another 3.2×. Overall, therefore, Glimpse clearly has the potential to reduce data transmitted by roughly 12-50× relative to pure offloading. Of course, all these reductions are predicated on the application only being interested in a sparse set of frames.



(a) Real-time Power Consumption (b) Real-time WiFi Utilization

Figure 4.18: **Real-time Power and WiFi Utilization.** Power consumption and WiFi transmissions of a primary imager become bursty when it is being gated by Glimpse system during an example one-to-one meeting scenario.

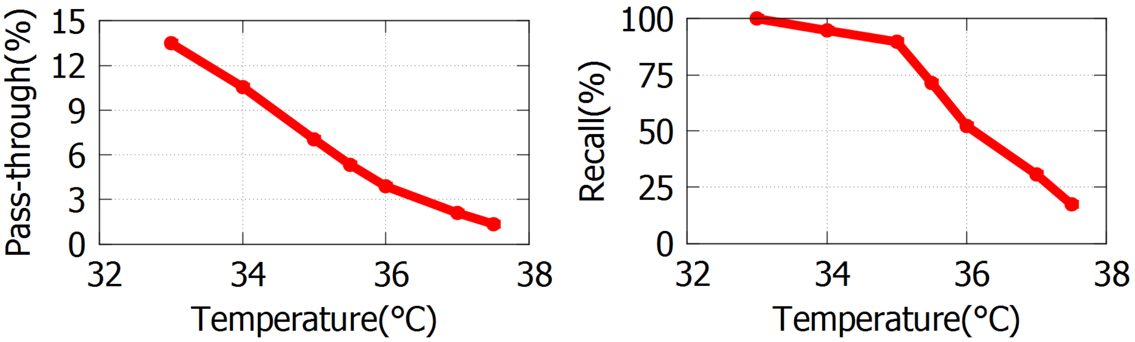
Scenario	Gated BW(Mbps)	Baseline BW(Mbps)
Driving	1.01	5.90
Corridor	0	5.15
Monitor	0.58	8.72
Outdoor	2.46	10.56
Lunch	2.61	8.25
Meeting	0.14	1.58
1 to 1	0.24	3.14

Table 4.4: **Glimpse WiFi bandwidth consumption.**

4.7.3 Glimpse Programmability

Here we run a cascade on the Glimpse similar to Figure 4.5a except the depth threshold is set to 1 meter. The goal here is to trigger the primary imager whenever the wearer's hand enters the field of view, indicating that the wearer is probably interacting with an object. Here we ask the

Glimpse wearer to interact with objects in hand while standing in an office setting with background thermal activities. In each experiment we varied the temperature threshold from $33°C$ to $37.5°C$ and measured the recall and the pass-through rates of the detected frames.



(a) Pass-through vs Temperature

(b) Recall vs Temperature

Figure 4.19: **Recall vs pass-through rate trade-off.** Recall and trigger dependence on the temperature threshold for hand detection cascade.

Figure 4.19 shows the recall and pass-through rates versus the threshold temperature on two separate graphs. The key observation is that as the temperature threshold increases the pass-through rate decreases which helps lowering the power consumption but it decreases the recall rate as well. There is a clear trade-off between the total power consumption (pass-through rates) and the resulting recall rate, which can be controlled straightforwardly via the corresponding parameter in the rejection cascade.

4.8 Related Work

Glimpse is built based on previous research on continuous mobile vision, ultra-low power depth sensing, and FarIR tracking. However, our system differs from previous work from multiple perspectives.

Continuous mobile vision: Prior work investigates various layers across the vision processing pipeline to identify the power bottleneck of each layer. Traditional off-the-shelf image sensors' power consumption is not proportional to their frame rate and resolution, thus it is not feasible to trade quality with power consumption using them. [56] [68] [30] tune hardware parameters of an image sensor to achieve proportional energy consumption to the number of pixels sensed. [70] leverages shared intermediate results within a vision processing software framework to reduce redundant computational overhead. [46] and [31] look at the benefit of using low-power accelerometers to gate the operation of the image sensor. Perhaps the biggest difference between our work and these is our advocacy for a *dedicated* hardware/software subsystem that, perhaps paradoxically, includes imagers, that applications can use in a *programmable* manner to control the computer vision costs they would incur by using the primary imaging pipeline.

Sensing depth at low power: Sensing depth at low power is hard in general. Several depth sensors [12] can operate at low power and high resolution. However, these sensors just report the depth of a single pixel. Other sensors [119, 83, 20, 134, 135, 123] can compute the depth of a wide field of view with high resolution. However, their power consumption is high. Our primary insight is that *for purposes of early discard*, it is often adequate to have depth estimated in subsets of the field of view and at coarse resolution. By identifying corresponding relaxations in the algorithm design, we deliver useful depth estimation at a few mW.

FarIR tracking at low-power: Prior works use FarIR sensors to detect the presence of objects of interests for Heating, Ventilation, Air Conditioning, (HVAC) and lighting control of building [42]. [52] also looks at the opportunity of using FarIR sensors for objects tracking. Different from these work, our FarIR subsystem is a customized design that can run on a microcontroller at low-power consumption.

Battery-free cameras: Work parallel to ours has shown battery-free cameras that harvest energy from ambient Radio Frequency (RF) signal sources and capture and transmit images on a forced duty-cycled basis [88] [116] [91]. These devices reduce the maintenance costs of camera systems by eliminating the battery, but due to their constrained energy budget, they suffer from limited computational capabilities required for on-board complex machine vision operations [86]. Although the vision algorithm presented here cannot be applied to those such energy constrained devices, the idea of application-specific cascaded operation for gating a high-power sensor using low-power sensors and algorithms (such as the use of motion detectors) seems a feasible approach to enable battery-free devices with more continuous operation.

4.9 Conclusion

In this work we designed, implemented and evaluated Glimpse, a programmable embedded system designed to detect at low power events worthy of further processing. Key innovations include a low-power imager array and computational fabric dedicated to image-discard calculations, a safe but expressive programming model for applications and coarse variants of conventional stereo depth and tracking algorithms suited for the discard pipeline. Combining Glimpse with a high-power imaging pipeline system allows the high-power pipeline to process less than 10% of the frames it would have processed otherwise, while drawing about 100mW of overall power. Correspondingly, the amount of video data transmitted for further processing is also lower by an order of magnitude. Based on these results, we believe that Glimpse moves continuous vision on lightweight wearables to the realm of the practical.

Chapter 5

TOWARDS BATTERY-FREE HD VIDEO STREAMING

5.1 Introduction

There has been recent interest in wearable cameras like Snap Spectacles [24] for applications ranging from lifecasting, video blogging and live streaming concerts, political events and even surgeries [33]. Unlike smartphones, these wearable cameras have a spectacle form-factor and hence need to be both ultra-light weight and have no heating issues during continuous operation. This has resulted in a trade-off between the usability of the device and its streaming abilities since higher resolution video streaming requires a bigger (and heavier) battery as well as power-consuming communication and processing units. For example, the Snap Spectacle is light-weight and usable but cannot stream live video [24] and can record only up to one hundred 10-second videos (effectively below 20 minutes) [24, 19] on a single charge.

In this chapter, we ask the following question: Can we design a low-power camera that can perform HD video streaming to a nearby mobile device such as a smartphone? A positive answer would enable a wearable camera that is light weight, streams high quality video, and is safe and comfortable to wear. Specifically, reducing the power consumption would reduce the size of the required battery, which in turn addresses the key challenges of weight, battery life and overheating. Finally, since users typically carry mobile devices like smartphones that are comparatively not as weight and power constrained, they can relay the video to the cloud infrastructure.

To understand why this is a challenge, let us look at the different components of a video-streaming device: image sensor, video compression and communication. Image sensors have optical lens, an array of photo-diodes connected to amplifiers and finally an ADC to translate the analog pixels into digital values. A video codec then performs compression in the digital domain to produce compressed video, which is then transmitted on the wireless medium. Existing ap-

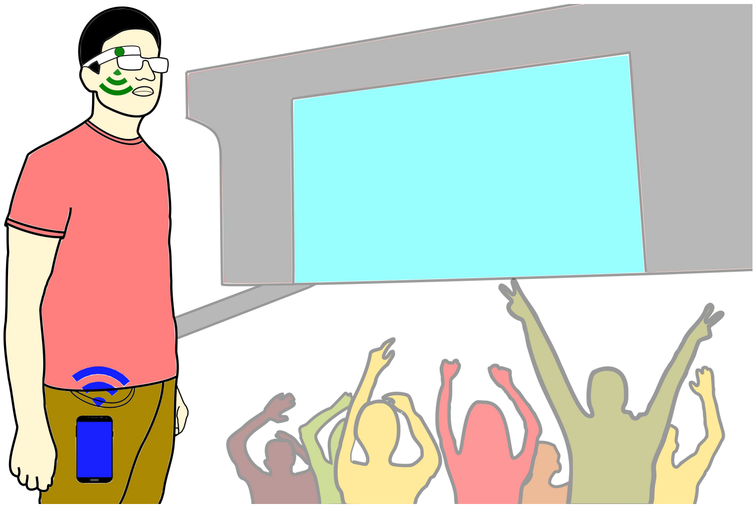


Figure 5.1: **Target application.** Our ultra-low power HD streaming architecture is targeted for wearable cameras. We achieve this by performing analog video backscatter from the wearable camera to a nearby mobile device (e.g., smartphone).

proaches optimize the camera and communication modules individually to minimize their power consumption. However, designing a video streaming device requires power-consuming hardware components and video codec algorithms that interface the camera and the communication modules. Specifically, optical lens and photo-diode arrays can be designed to consume as little as 1.2 μW [48]. Similarly, recent work on backscatter can significantly lower the power consumption of communication to a few microwatts [60, 55], using custom ICs. However, interfacing the camera hardware with backscatter requires ADCs and video codecs that significantly add to the power consumption.

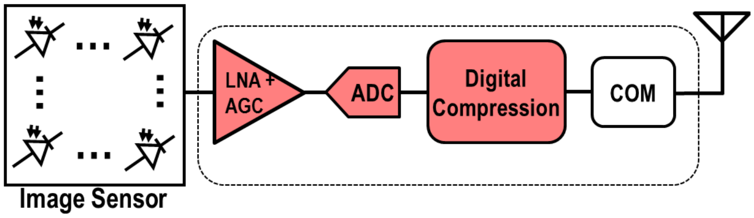
Table 5.1 shows the sampling rate and data rate requirements for the ADC and video codec respectively. HD video streaming requires an ADC operating at a high sampling rate of more than at least 10 MHz. While the analog community has reduced the ADC power consumption at much lower sampling rates [109, 37], state-of-the-art ADCs in the research community consume at least a few milliwatts at these high sampling rates [82]. Additionally, the high data rate requires the oscillator and video codec to run at high clock frequencies that proportionally increases the power

Figure 5.2: **Sample HD video frame streamed with our analog video backscatter design.** Our prototype was placed at 4 feet from the reader.

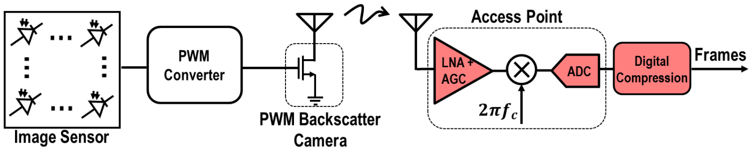
consumption. Specifically video codecs at these data rates consumes 100s of milliwatts to a few watts of power [2].

We present a novel architecture that enables video streaming on low-power devices. Instead of independently optimizing the imaging and the communication modules, we jointly design these components to significantly reduce the system power consumption. Our architecture, shown in Fig. 5.3b, takes its inspiration from the Great Seal Bug [8] that uses changes in a flexible metallic membrane to reflect sound waves. Building on this idea, we create the first "analog" video backscatter system. At a high level, we feed analog pixels from the photo-diodes directly to the backscatter antenna; thus, eliminate power-hungry ADCs, amplifiers, AGCs and codecs.

Our intuition for the analog video backscatter approach is as follows. Our goal is to shift as much as possible of the power-hungry analog-to-digital conversion operation to the reader. Because analog signals are more susceptible to noise than digital signals, we split the ADC conversion process into two phases, one that is performed at the video camera, and one that is accomplished at the reader. At the video camera, we convert the analog pixel voltage into a pulse that is discrete in amplitude but continuous in time. This signal is sent via backscatter from the camera to the reader. Avoiding the amplitude representation in the wireless link provides for better noise immunity. The



(a) Conventional Camera Design



(b) Our Camera Design Approach

Figure 5.3: The amplifier, AGC, ADC and compression module consume orders of magnitude higher power than what is available on a low-power device. In our design, these power hungry modules have been delegated to the reader eliminating their power consumption overhead from the wireless camera.

reader measures the continuous length pulse it receives to produce a binary digital value. Philosophically, the continuous-time, discrete amplitude pulse representation used in the backscatter link is similar to the one used in the extremely power efficient biological nervous system, which encodes information in spikes that do not vary in amplitude, but are continuous in time [108].

Specifically, our design synthesizes three key techniques. First, we show how to interface the pixels directly to the backscatter hardware without using ADCs. To do this we transform the analog pixel values into different pulse widths using a passive ramp circuit and map these pulses into pixels at the reader. Second, we achieve intra-frame compression by leveraging the redundancy inherent to typical images. Intuitively, the bandwidth of the signal is proportional to the rate of change across adjacent pixels and since videos tend to be redundant, the bandwidth of the analog signal is inversely proportional to the redundancy in the frame. Thus, by transmitting pixels consecutively, we can implicitly perform compression and significantly reduce the wireless bandwidth. Finally to achieve inter-frame compression, we design a distributed algorithm that reduces the data transmitted by the camera while delegating most of inter-frame compression functionality to the reader. At a high level, the camera performs averaging over blocks of nearby pixels *in the analog domain* and transmits these averaged values using our backscatter hardware. The reader compares these averages with those from the previous frame and only requests the blocks that have seen a significant change in the average pixel value thus reducing the transmission between subsequent video frames.

We implement a proof-of-concept prototype of our analog backscatter design on an ultra-low power FPGA platform and a custom implementation of the backscatter module. Because there exists no HD camera that provides access to its raw pixel voltages, we connect the output of a DAC converter to our backscatter hardware to emulate the analog camera and stream raw analog video voltages to our backscatter prototype. A sample frame from an HD video streamed with our backscatter camera is presented in Fig. 5.2. More specifically, our findings are:

- We can stream 720p HD video at 10 frames per second up to 16 feet from the reader. The Effective Number of Bits (ENOB) received for each pixel at distances below six feet is more than 7 bits. This is for all practical purposes identical to the quality of the source HD video.
- Our inter and intra frame compression algorithms can reduce the total bandwidth requirements by up to two orders of magnitude comparing to the raw video. For example, for 720p HD video at 10 fps, our design uses a wireless bandwidth of only 0.98 MHz and 2.8 MHz in an average-case and worst-case scenario video respectively.

We design and simulate an ASIC implementation for our system, taking into account the power consumption for the pixel array. Our results show that the power consumption for video streaming at 720p HD is 321 μW and 252 μW for 60 and 30 fps respectively. The power consumption at 1080p full-HD is 806 μW and 561 μW at 60 and 30 fps. We run experiments with RF power harvesting from the reader which shows that we can support 1080p full-HD video streaming at 30 fps up to distances of 8 feet from the reader. *This demonstrates that we can eliminate batteries and achieve HD video streaming on battery-free devices using our analog video backscatter approach.*

Table 5.1: Raw Digital Video Sampling and Bitrate Requirement

	Frame Rate: 60 fps		**Frame Rate: 30 fps**		**Frame Rate: 10 fps**	
Video Quality	**Sampling Rate (MHz)**	**Data Rate (Mbps)**	**Sampling Rate (MHz)**	**Data Rate (Mbps)**	**Sampling Rate (MHz)**	**Data Rate (Mbps)**
1080p (1920x1080)	124.4	995.3	62.2	497.7	20.7	165.9
720p (1280x720)	55.3	442.4	18.4	221.2	9.2	73.7
480p (640x480)	18.4	147.4	9.2	73.7	3.1	24.58
360p (480x360)	10.4	82.9	5.2	41.5	1.7	13.8

5.2 A Case for Our Architecture

Fig. 5.3a shows the architecture of a traditional wireless camera. The output of the photodiodes is first amplified by a low noise amplifier (LNA) with automatic gain control (AGC). The AGC adjusts the amplifier gain to ensure that the output falls within the dynamic range of the analog to digital converter (ADC). Next, the ADC converts the analog voltage into discrete digital values. The video codec then compresses these digital values, which is then transmitted on the wireless communication link.

Unfortunately, this architecture cannot be translated to an ultra low-power device. Although camera sensors consisting of an array of photodiodes have been shown to operate on as low as 1.2 μW of power [48] at 128 × 128 resolution, amplifiers, AGC, ADC and the compression block require orders of magnitude higher power. Furthermore, the power consumption exacerbates as we scale the resolution and/or the frame rate of the camera. A low resolution 144p video recorded at 10 fps requires an ADC sampling at 368 KSPS that generates uncompressed data at 2.95 Mbps. With the latest advancements in ADCs [80] and backscatter communication system [60], uncompressed data can be digitally recorded using low-power ADCs and transmitted using digital backscatter while consuming below 100 μW of power which is within the power budget of energy harvesting platforms. However, as we scale the resolution to HD quality and higher such as 1080p and 1440p, the sampling rate of ADC increases to 10-100 MHz and uncompressed data is generated at 100 Mbps to Gbps. ADCs operating at such high sampling rates consumes at least a few mW [82]. Further, a compression block which operates in real time on 100 Mbps to 1 Gbps uncompressed data streams consume up to one Watt of power [2]. This power budget is orders of magnitude more than that is available on harvesting platforms. This demonstrates that while existing architectures could possibly operate on harvested power for low-resolution video by leveraging recent advancements in low-power ADCs and digital backscatter, as we scale the resolution and/or the frame rate of the wireless camera, existing architectures use of ADCs and compression block power budget exacerbates power consumption beyond the realm of battery-free devices.

5.3 System Design

In the rest of this section, we first describe our analog video transmission scheme followed by the intra-frame compression and finally the interactive distributed inter-frame compression technique.

5.3.1 Analog Video Backscatter

Problem. At a high level, we eliminate the ADC on the camera to significantly reduce the power consumption. This means that we are limited to working with analog voltage output of the photodiodes. A *naive* approach would be to leverage existing analog backscatter technique used in wireless microphones [114] and implement an analog backscatter camera. In an analog backscatter system, the output of a sensor directly controls the gate of a field-effect transistor (FET) connected to an antenna. As the output voltage of the sensor varies, it changes the impedance of the FET which amplitude modulates the RF signal backscattered by the antenna. The reader decodes the sensor information by demodulating the backscattered RF signal which is amplitude modulated. However, this approach cannot be translated to a camera sensor. The output of the photodiode has very limited dynamic range (less than 100 mV under indoor lighting conditions). These small voltage changes map to a very small subset of radar cross-sections at the antenna [34]. As a result,

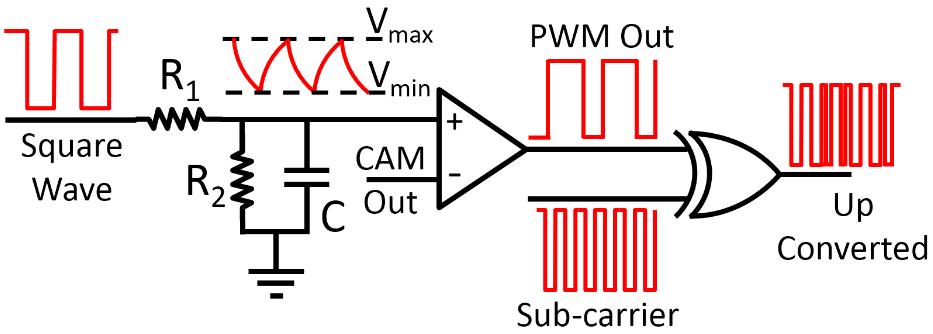


Figure 5.4: **Architecture of the PWM converter**.

the antenna backscatters a very weak signal and since wireless channel and receivers add noise, this approach results in low SNR signal at the reader. This limits the system to poor signal quality and limited operating range. One can potentially get around this constraint by introducing a power-hungry amplifier and AGC but this would negate the power savings of analog backscatter.

Table 5.2: Intra-Frame compression for average/worst-case scenarios across 100 videos.

	Frame Rate: 60 fps		**Frame Rate: 30 fps**		**Frame Rate: 10 fps**	
Video Resolution	**Raw Data Rate(Mbps)**	**Analog BW (MHz)**	**Raw Data Rate(Mbps)**	**Analog BW (MHz)**	**Raw Data Rate(Mbps)**	**Analog BW (MHz)**
1080p	995.3	10.8/30.5	497.6	5.6/18.7	165.8	1.9/9.3
720p	442.3	6.3/15.6	221.1	8/3.1	73.7	0.98/2.8
480p	147.4	3/6.7	73.7	1.6/3.3	24.5	0.53/1.4
360p	82.9	1.9/3.9	41.4	0.94/2.3	13.8	0.32/0.78

Our solution. Instead of using linear amplifiers to amplify the analog video stream, we use comparators to pulse width modulate [58] camera sensor's analog output and convert into digital domain as shown in Fig. 5.4. At a high level, pulse width modulation (PWM) converts analog input voltage to different pulse widths. Specifically, the output of PWM modulation is a square wave, where the duty cycle of the output square wave is proportional to the analog voltage of the input signal. In the frequency domain, in additional to the desired baseband signal, PWM modulation also generates additional harmonics which contain redundant copies of the input analog video signal [58**?**] and can result in electromagnetic interference (EMI) from the antenna at higher frequencies. However, since the typical power re-radiated by the antenna in a backscatter system is significantly lower than the power transmitted by an active radio, the resulting EMI from the harmonics of the baseband PWM signal in a backscatter system are also negligible (less than ¡-20 dBm in all scenarios). In section §**??**, we discuss how one can further suppress the miniscule EMI generated from backscattering of baseband PWM signals. However, the implementation of EMI suppression is beyond the scope of this chapter. As we show next, one can implement a PWM converter with passive RC components and a comparator, thereby consuming very low power. Fig. 5.4 shows the architecture of our PWM converter. The input is a square wave operating at frequency f, which is determined by the frame rate and resolution of the camera. The square wave is first low pass filtered by an RC network to generate a triangular waveform as shown in the figure. This triangular waveform is then compared to the pixel value using a comparator. The comparator output a zero when the triangular signal is less than the pixel value and one otherwise. Thus, the width of the pulse being generated changes with the pixel value: lower pixel values have larger pulse duration while higher pixel values have a low pulse duration. We choose the minimum and maximum voltages for the triangular signal to ensure that the camera's pixel output is always within this limit which also maximizes the gain (analog voltage to digital pulse width) of the PWM modulation operation.

The final component in our hardware design is sub-carrier modulation that addresses the problem of self-interference. Specifically, in addition to receiving the backscatter signal, the receiver also receives a strong interference from the transmitter. Since the sensor data is centered at the carrier frequency, the receiver cannot decode the backscattered data in the presence of a strong in-band interferer. Existing backscatter communication systems such as RFID and Passive Wi-Fi get around this problem by using sub-carrier modulation. These systems use a sub-carrier to shift the signal from the carrier, to a frequency offset Δf from the carrier frequency. The receiver can then decode the backscattered signal by filtering the out of band interference. Another consideration in picking the right sub-carrier frequency is to avoid aliasing, basically the sub-carrier frequency should not be smaller than the effective bandwidth of the analog signal.

Our PWM based design integrates subcarrier modulation. Specifically we implement this modulation with a simple XOR gate. The sub-carrier can be approximated by a square wave operating at Δf frequency. We input the sub-carrier and the PWM output to an XOR gate to up convert the PWM signal to a frequency offset Δf. Sub-carrier modulation addresses the problem of self-interference at the reader and as a result the PWM backscattering wireless camera can now operate with high SNR and achieve large operating ranges. We show in §5.5.1 that our PWM backscatter wireless camera can operate up to 16 feet for 2.76 MHz bandwidth 10 fps monochrome video signal in HD resolution. We also show in §5.6 that our camera system can work up to 150 feet for 50 KHz video signal in 112×112 resolution. This is over 4x improvement over the 3 kHz bandwidth analog backscatter wireless microphone [118].

5.3.2 Intra-Frame Compression

There is significant redundancy in the pixel values of each frame of uncompressed video. This is because natural video frames usually include objects larger than a single pixel, which means that the colors of nearby pixels are highly correlated. At the boundaries of objects (edges), larger pixel variations can occur, but in the interior of an object, the amount of pixel variation is much less than the theoretical maximum. The net result of the pixel correlations is that the information needed to represent natural images is far less than the worst case maximum. Traditional digital systems use a video codec to reduce this redundancy. Specifically, the raw output of the camera is first digitized by the ADC at the Nyquist rate determined by the resolution and frame rate. The raw digital data stream is then fed to the video codec that implements compression algorithms.

In the absence of the ADC, our wireless camera transmits analog video directly. We however note that the bandwidth of any analog signal is a function of the new information contained in the signal. Inspired by analog TV broadcast that transmit pixel information in a raster scan (left to right), we introduce and implement a zig-zag pixel scanning in which the pixels are scanned from left to right in odd rows and from right to left in even rows. The intuition here is that neighboring pixels have less variation and hence the resulting signal would occupy less bandwidth.

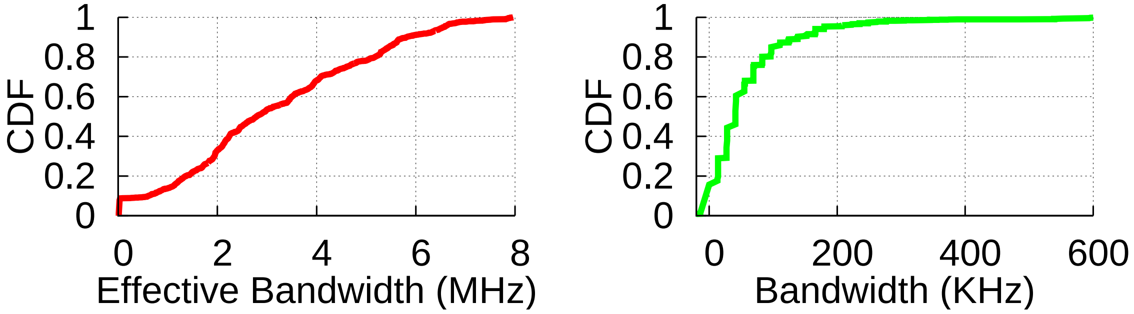
We evaluate how well such a zigzag transmission performs in terms of bandwidth reduction. We download one hundred 60 fps Full-HD (1080p) videos from [26] to use as our baseline. These videos range from slow to fast moving videos with movie action stunts, running animals, racing, etc. To create video baselines at different resolutions and frame rates we resize and subsample these Full-HD video streams. For each of the resulting video resolutions and frame rates, we create an analog video by zig-zagging the pixels as described above. We then apply low-pass filters with different bandwidth on this analog signal and report the minimum bandwidth at which the PSNR of the resulting signal is greater than 30 dB. The bandwidth requirement reported in Table 5.2 shows the average/worst-case scenario i.e. the bandwidth which ensures that average/worst-case types of videos can be recovered with minimal quality degradation. We outline the uncompressed digital data rate for reference. We note that compared to the digital uncompressed data sampled at Nyquist rate, the analog signal occupies 17.7–32.6x at worst case and 43–92x at average case less bandwidth demonstrating the intra-frame compression capability. The CDF of effective bandwidth for our 30 fps 720p videos dataset is shown in Fig. 5.5a and demonstrates that on an average-case 30 fps 720p video we get up to $\sim$71x improvement comparing to raw digital video transmission.

Finally we note that, comparing to raster, zig-zag scan faces less discontinuity in pixel values, because instead of jumping from the last pixel in a row to the first pixel in the next row, it continues at the same column in the next row, thus takes more advantage of the inherent redundancy in the video. This translates to further lowering bandwidth utilization in the wireless medium. As an example, on average, zig-zag pixel scanning occupies $\sim$120KHz and $\sim$60KHz less bandwidth than raster scan in a 60 fps and 30 fps 720p video streaming respectively. The CDF of bandwidth improvement for zig-zag scanning over raster in our one hundred 30 fps 720p videos dataset is shown in Fig. 5.5b. The plot shows that use of zig-zag pixel scanning provides bandwidth efficiency benefit over raster.

5.3.3 Distributed Inter-Frame Compression

In addition to the redundancy within a single frame, the raw video output also has significant redundancy between consecutive frames. Existing digital architectures use a video codec to compress the raw video prior to wireless transmission to remove redundancy between frames and reduce power and bandwidth. Our approach to address this challenge is to again leverage the reader. Similar to backscatter communication, where we move the power-hungry components such as ADCs to the reader, we also move the compression functionality to the reader. Specifically, our design distributes the compression algorithm between the reader and the camera. We delegate the power-hungry computation to the reader and leverage that our camera system is capable of transmitting *super-pixels*. A super-pixel value is the average of a set of adjacent pixel values. A camera frame consists of an $N \times N$ array of pixels which can be divided into smaller sections of $n \times n$ pixels. A super-pixel corresponding to each section is the average of all the pixels in the $n \times n$ section of the frame. The camera sensor is a photodiode that outputs a current proportional to the intensity of light. The camera uses a buffer stage at the output to convert the current into an output voltage. To compute the super-pixel, the camera first combines the current from the set of pixels and then converts the combined current into a voltage output, all in the analog domain. We note that averaging of close by pixels is supported by commercial cameras including the CentEye Stonyman camera [6].

Now, the camera instead of transmitting the entire $N \times N$ pixel frame, transmits a lower resolution frame consisting of $n \times n$ sized super-pixels (the average value of the pixels in the $n \times n$ block) called the low resolution or L frame, reducing the data transmitted by the camera by a factor of $\frac{N^2}{n^2}$. The reader performs computation on L frames and implements a change driven compression technique. At a high level, the reader in real time compares the incoming L frame with the previous L frame. If a super-pixel value differs by more than a predetermined threshold between frames, it determines that the super-pixel has sufficiently changed and requests the camera to transmit all the pixels corresponding to the super-pixel. If the difference does not cross the threshold, then the reader uses the pixel values corresponding to the previous reconstructed frame to synthesize



(a) Effective bandwidth

(b) Zig-zag vs. raster improvement

Figure 5.5: **CDF of effective bandwidth for zig-zag pixel scanning and how much improvement it provides over raster.**

the new frame and does not request new pixel values. We call the frame which contains the pixel values corresponding to the sufficiently changed super-pixels, the super-pixel or *S* frame. In addition to transmitting the *S* and *L* frames, the camera periodically transmits uncompressed frames (*I*) to correct for potential artifacts and errors that may have been accumulated in the compression process.

In streaming camera applications, the communication overhead of reader requesting specific pixel values is minimal and is implemented using a downlink similar to prior 100-500 kbps designs [60, 105] where the receiver at the camera uses a simple envelope detector to decode the amplitude modulated signal from the reader. We note that prior design can achieve Mbps downlink transmissions [96] as well as full-duplex backscatter [73] which can be used to remove the downlink as a bottleneck.

The sequence of frames and pixels transmitted by our camera is shown in Fig. 5.6. The figure shows that between two *I* frames, a camera transmits *M* low resolution *L* frames and *M* super-pixel *S* frames which contain pixel values corresponding to super-pixels whose values have differed more than the threshold between consecutive frames. The number of *L* and *S* frames (*M*) transmitted between consecutive *I* frames is a tradeoff between the overhead associated with transmission of

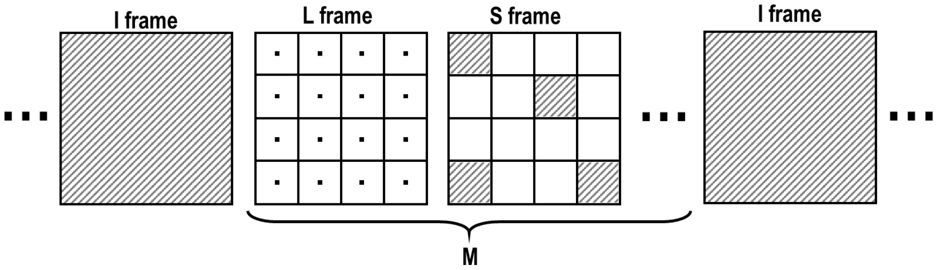


Figure 5.6: **Distributed Compression** Sequence of frames and pixels transmitted by the camera to the reader.

full resolution frames and artifacts and errors that are introduced by the compression algorithm. In our implementation of 10 fps HD video streaming, we transmit an *I* frame after a transmission of every 80 (M = 80) *L* and *S* frames.

5.4 Implementation

We first describe our implementation of the wireless camera built using off-the-shelf components on a custom design Printed Circuit Board (PCB). We use the COTS prototype to evaluate the performance of the wireless camera in various deployment. We then present the application specific integrated circuit (ASIC) design of the wireless camera that we used to quantify the power consumption for a range of video resolutions and frame rates.

COTS implementation. Our wireless camera design eliminates the power-hungry ADCs and video codecs and only consists of the image sensor, PWM converter, a digital block for camera control and sub-carrier modulation, backscatter switch and an antenna. We built two hardware prototypes; a high definition (HD) and another low-resolution version of the wireless camera.

We built the low resolution wireless camera using the $112x112$ grayscale random pixel access camera from CentEye [6] which provides readout access to the individual analog pixels. We implement the digital control block on a low power Igloo Nano FPGA by Microsemi [10]. The analog output of the image sensor is fed to the PWM converter built using passive *RC* components and an NCX2200 comparator by Maxim [15]. We set $R_1 = 83K\Omega$, $R_2 = 213K\Omega$ and $C = 78pF$ in our

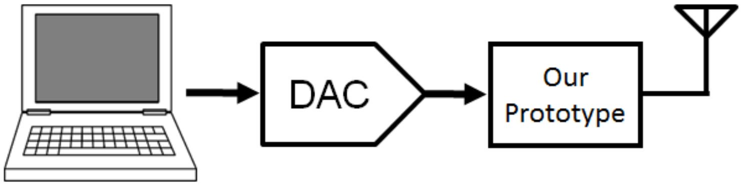


Figure 5.7: **HD Wireless Camera Prototype.** A laptop emulating analog output camera, feeds raw pixel data into our backscatter prototype. Then the PWM encoded pixels will be backscattered to the reader.

PWM converter design which supports video frame rate of up to 13 fps. The digital output of the PWM converter is the input to the FPGA which performs sub-carrier modulation at 1.024 MHz using an XOR gate and outputs the sub-carrier modulated PWM signal to the ADG919 switch by Analog Devices which switches a 2 dBi dipole antenna between open and short impedance states. The FPGA injects frame and line synchronization patterns into the frames data before backscattering. We use Barker codes [4] of length 11 and 13 for our frame and line synchronization patterns respectively. Barker codes have high-autocorrelation property which will help the reader more efficiently detect them in the presence of noise.

We use Verilog to implement the digital state machine for camera control and sub-carrier modulation which can be easily translated into ASIC using industry standard EDA tools. We can further reduce the power consumption of our system by using the distributed compression technique. As shown in §5.6, when camera is deployed in a normal lab space, we achieve an additional compression ratio of around 30× which proportionately reduces wireless transmissions.

To develop an HD resolution wireless camera, we need access to the raw analog pixel outputs of an HD camera and currently there exists no camera in the market that provides that access. To get around this constraint, we download HD resolution sample videos lasting 1 minute each from YouTube. We output the recorded digital images using a USB interface to an analog converter (DAC) followed by an amplifier to simulate voltage levels corresponding to an HD quality image sensor operating at 10 fps. Given the USB speeds, we could achieve the maximum frame rate of

Table 5.3: Power Consumption

	Frame Rate: 60 fps	**Frame Rate: 30 fps**	**Frame Rate: 10 fps**
Video Quality	**Power (μW)**	**Power (μW)**	**Power (μW)**
1080p (1920x1080)	806.50	560.63	167.77
720p (1280x720)	320.94	252.10	78.31
480p (640x480)	126.88	106.78	36.71
360p (480x360)	75.63	65.68	25.11

10 fps. We feed the voltage output to our PWM converter. The high-level block diagram of this implementation is shown in Fig. 5.7. For the high-resolution version of the wireless camera, we set $R_1 = 10K\Omega$, $R_2 = 100K\Omega$ and $C = 10pF$ and used LMV7219 comparator by Texas Instruments [11] in our PWM converter. The digital block and the rest of the components of the system were exactly same as the low resolution wireless camera described above except that sub-carrier frequency is set to ~10 MHz here to avoid aliasing.

ASIC Design. Our design eliminates the power hungry LNA, AGC and ADC at the wireless camera and instead, delegates them to the reader to reduce the power consumption of the wireless camera by orders of magnitude. However, since commercially available Stonyman cameras like CentEye and components such as FPGA are designed for flexibility and ease of prototyping and are not optimized for power, our COTS implementation cannot achieve the full power savings from our design. To that end, next, we analyze the power consumption of an application specific integrated circuit (ASIC) implementation of our design for a range of video resolutions and frame rate. An ASIC would integrate the image sensor, PWM converter, digital core, oscillator and backscatter modulator in a small silicon chip. Our design is implemented in TSMC 65 nm LP CMOS process.

We use Design Compiler by Synopsis [21] to synthesize transistor level from the behavioral model of our digital core written in Verilog. We custom design the PWM converter, oscillator and the backscatter modulator described in §5.3 in Cadence software and use industry standard simulation tools to estimate power. To support higher resolution and higher frame rate video, we simply increase the operating frequency of the oscillator, PWM converter and the digital core. As an example, 360p at 60 fps requires a 10.4 MHz input clock which consumes a total of 42.4 μW in the digital core, PWM converter and backscatter switch whereas a 1080p video at 60 fps requires $\sim$124.4 MHz input clock which consumes 408 μW in the digital core, PWM converter and the backscatter switch. To eliminate aliasing in all cases we pick a sub-carrier frequency equal to the input clock of each scenario; note that sub-carrier frequency cannot be less than effective bandwidth of the signal reported in Table 5.2.

We use existing designs to estimate the power consumption of the image sensor for different video resolutions. State of the art image sensors consume 3.2pW/($frame \times pixel$) [120] for the pixels-only image sensor which results in 33.2 μW for 360p resolution which increases to 398 μW for 1080p resolution video at 60fps. Table 5.3 shows the power consumption of the ASIC version of our wireless camera for different video resolution and frame rates. Note that this is the power consumption before the distributed inter-frame compression distributed across the reader and the camera which could further reduce wireless bandwidth and power consumption.

5.4.1 Reader

We implement the reader on the USRP X-300 USRP software defined radio platform by Ettus Research [23]. The reader uses a bi-static radar configuration with two 6 dBi circularly polarized antennas [1]. The transmit antenna is connected to a UBX-160 daughterboard which transmits a single tone signal. The output power of the USRP is set to 30 dBm using the RF5110 RF power amplifier [16]. The receive antenna is connected to another UBX-160 daughter board configured as a receiver. The receive daughter board down converts the PWM modulated backscattered RF signal to baseband and samples it at 10 Msps. The digital samples are transmitted to the PC via the Ethernet interface.

The block diagram of the signal processing steps required to recover the transmitted video is shown in Fig. 5.8. For example at the low-resolution video, the received data is centered at an offset frequency of the 1.024 MHz, so, we first filter the received data using a 500 order bandpass

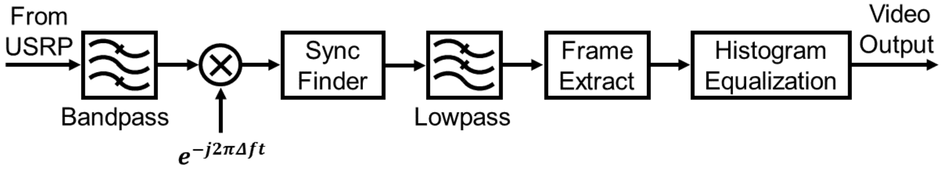


Figure 5.8: Signal processing steps at the reader to recover video from PWM backscatter signal.

filter centered at 1.024 MHz. Then, we down convert the signal to baseband using a quadrature down conversion mixer. Next, we correlate the received data with 13 and 11 bit Barker codes to find the frame sync and line sync. After locating the frame and line sync pulses, we extract the time periods corresponding to a row of PWM modulated pixel values and low pass filter the signal with a 500 order filter to remove out of band noise. We divide the row in evenly spaced time intervals corresponding to the number of pixels in a single row of the image sensor. We recover the pixel value by calculating the average voltage of the signal which corresponds to the duty cycle of the PWM modulated signal. We sequentially arrange the recovered pixel values into rows and columns to create video frames. Finally, we run histogram equalization algorithm on the video which adjusts the intensity of the frames and enhances the contrast of the output video [44].

5.5 Evaluation

We evaluate various aspects of our wireless camera system. We start by characterizing the received video quality from the camera as a function of its distance to the reader. Next, we evaluate the performance of our wireless camera while it is being worn by the user under different head positions and orientations. We also evaluate the power available by harvesting from RF signals transmitted by the reader and demonstrate the feasibility of a battery-free wireless camera. Finally, we evaluate the distributed interactive compression algorithm under two different scenarios.

5.5.1 Operational Range

We deploy our high definition wireless camera device in a regular lab space. We use the USRP based reader implementation (described in §5.4.1) and set it to transmit 23 dBm into a 6 dBi patch antenna. This is well below the 30 dBm maximum transmit power permitted by FCC in the 900 MHz ISM band. We vary the distance between the reader and the wireless camera prototype from 4 to 16 feet and configure the camera to repeatedly stream a 15-second-long video using PWM backscatter communication. The 720p resolution video is streamed at 10 fps and the pixel values are encoded as 8-bit values in monochrome format. We record the wirelessly received video at the reader and measure the Signal to Noise Ratio (SNR) and from that we calculate Effective Number of Bits (ENOB) [61] at the receiver.

We plot the ENOB of the video digitized by the reader as a function of distance between the reader and the wireless camera in Fig. 5.9. The plot shows that up to 6 feet from the reader, we achieve an ENOB greater than 7 which translates to negligible degradation in the quality of the video streamed using PWM backscatter. As the distance between the reader and the camera increases the SNR degrades which translates to decrease in ENOB. Beyond the distance of 16 feet we stop reliably receiving video frames. A separation of 16 feet between the reader and the wearable camera is more than sufficient for wearable cameras which are typically located a few feet away from readers such as smartphones. For reference, Fig. 5.10 shows frames from a 720p HD color video backscattered with our prototype at different distances from the reader which resulted in different ENOB values.

We note the following: our analog video backscatter approach is ideal for the wearable camera scenario since the video is streamed to a nearby mobile device such as a smartphone. In this scenario the SNR is high and hence the quality degradation due to an analog approach is not severe. However, we get the significant power reduction benefits using our analog backscatter approach.

5.5.2 Effect of Head Pose and Motion

Next, we asked a participant to wear the antenna of our wireless camera on their head and perform different head movements and poses while standing at around five feet from the reader with fixed location on a table. The poses included standing still, moving head to left and right, rotating head to the side, moving head up and down and talking. Hence, our evaluation includes scenarios where there is relative mobility between the reader and the camera, and in fact it also includes cases

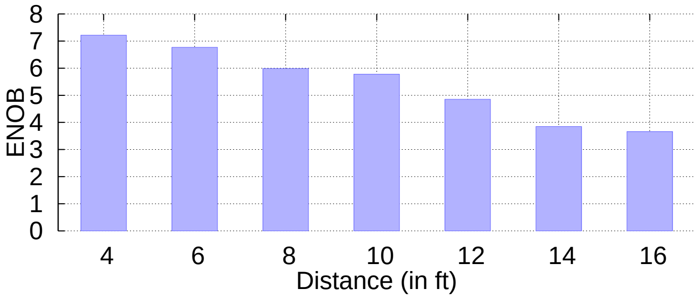


Figure 5.9: **ENOB of the Received Video.** ENOB of the received video versus our wireless camera prototype distance from the reader.

(a) Original Image

(b) 7 bits

(c) 4 bits

(d) 3 bits

Figure 5.10: **Effective Number of Bits (ENOB) versus video quality.** The frame corresponding to ENOB of 3 shows a video quality degradation.

where there is no line of sight communication between the reader and the camera. This evaluates our wireless camera for in-situ applications and how movements and antenna contact with body can impact its video quality. We recorded the streaming video with the reader and measured the SNR and ENOB as we did in §5.5.1. Fig. 5.11 plots the ENOB of the received video for five different

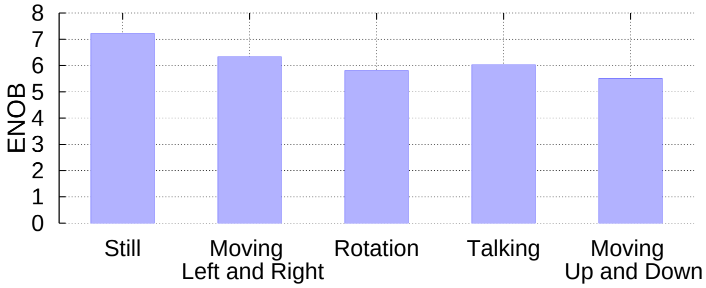


Figure 5.11: **ENOB of Received Video Under Different Head Motions.**

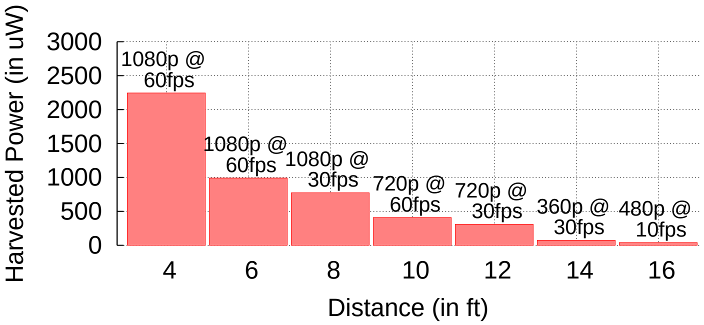


Figure 5.12: **Power Harvesting.** We plot the average power harvested by the battery-free hardware over different distances from the reader.

poses and movements. This plot shows that we can achieve sufficient ENOB while performing most of these gestures which provides a high quality video comparing to the original source video.

5.5.3 RF Power Harvesting

Next, we evaluate the feasibility of developing a battery-free HD streaming camera that operates by harvesting RF signals transmitted by the reader. We build an RF harvester for the 900 MHz ISM band based on state of the art RF harvester design [97]. The harvester consists of a 2 dBi dipole antenna followed by the rectifier which converts incoming RF signals into a low voltage DC output. The low voltage DC is amplified by a DC-DC converter to generate the voltage levels required for the operation of the image sensor and digital control block. We measure the power available at the output of the DC-DC converter.

We configure the reader to transmit a single tone at 30 dBm into a 6 dBi patch antenna and move the RF harvester away from the reader. Fig. 5.12 plots the available power at the harvester as a function of distance. Based on the available power, we also plot the maximum resolution and frame rate of the video that could be transmitted by an RF-powered ASIC version of our wireless camera. The plot shows that at close distances of 4 and 6 feet there is sufficient power available from RF energy harvesting to operate the wireless camera at 60 fps 1080p resolution. As the distance increases, the available power reduces which translates to a lower resolution video being continuously streamed from the wireless camera. At 16 feet the wireless camera can continuously stream video at 10 fps 480p resolution; beyond this distance, the harvester does not provide sufficient power to continuously power the wireless camera. We note that this is the performance of the camera without using the distributed inter-frame compression algorithm described in §5.3.3. The distributed inter-frame compression algorithm, which is distributed across the wireless camera and the reader, reduces transmissions from the camera. This translates to lower power consumption and consequently increased operating distances.

5.5.4 Distributed Inter-Frame Compression Evaluation

We consider two scenarios to evaluate our distributed inter-frame compression algorithm discussed in Section 5.3.3. We analyze HD video streamed from a fixed highway monitoring camera and an action camera mounted on a user riding a motorcycle [13]. We evaluate the trade-off between compression ratio and PSNR under both static and dynamic video feeds with our design. We measure the Peak Signal to Noise Ratio (PSNR) for different compression ratios by varying the threshold at which we consider that the super-pixel (20 by 20 pixels) has significantly changed. A higher threshold would result in higher compression ratios but at the cost of degraded PSNR. Fig. 5.13 shows PSNR of the compressed video as a function of the compression ratio for our distributed inter-frame compression technique. The plot shows that for static cameras, we can achieve compression ratio of at least 35 while maintaining a PSNR of above 30 dB. On the other hand,

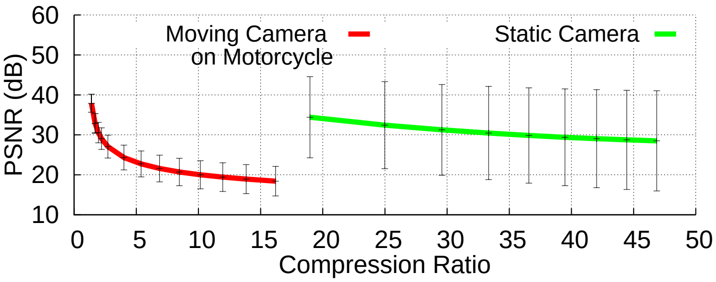
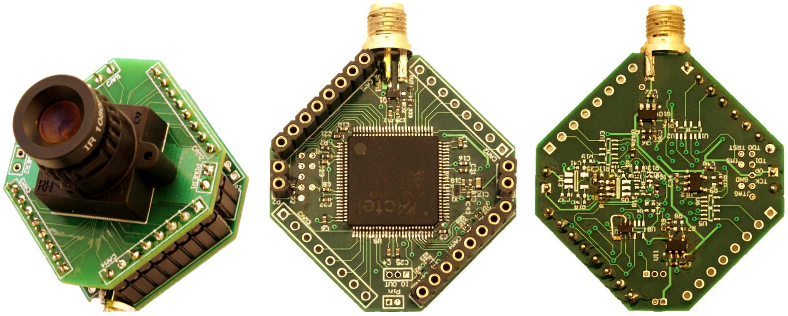


Figure 5.13: **Evaluation of distributed inter-frame compression algorithm for HD video.** The plots show PSNR at different compression ratio for two HD videos.

for dynamic videos recorded using a mobile camera mounted on a motorcycle, we were able to achieve a compression ratio of 2 for PSNR of greater than 30 dB. This is expected since in mobile scenarios, majority of pixel values significantly change between frames resulting in lower compression using our approach. We note that we could address this by implementing a more complex compression algorithm at the reader which tracks moving objects in the frame and requests specific pixel locations to achieve better compression similar to video codecs. Implementing such complex compression algorithms however is not in the scope of this chapter.

5.6 Low-Resolution Security Camera

So far, we have demonstrated how a high-resolution video is a useful paradigm for a wearable camera system. However, there are various other applications, such as security system and smart homes, which do not require a high-resolution video and a lower resolution video would suffice for applications such as face detection. Specifically, wireless cameras are increasingly popular in security and smart home applications. In contrast to the wearable camera application, these cameras require low-resolution but much longer operating distances. To show that our design can extend to such applications, we evaluate the operation range at 112×112 resolution at 13 fps. Our IC design at this resolution consumes less than 20 μW, without accounting for any savings that result due to distributed inter-frame compression.



(a) (b) (c)

Figure 5.14: **Prototype of our low-resolution video streaming home security camera.** Image of the analog camera, FPGA digital core and pulse width modulated (PWM) backscatter, all implemented using COTS components. The overall board measures 3.5 cm by 3.5 cm by 3.5 cm.

To evaluate the range, we use a 13 fps 112×112 resolution gray-scale random pixel access camera from CentEye [6] as our image sensor. The camera consists of a photodiode image sensor array, a trans impedance buffer stage (to convert photodiode current to voltage) followed by a low noise amplifier. The camera is extremely flexible and the user can modify various settings such as gain of the amplifier stage and if desired, completely bypass the amplifier. We use this unique feature of the camera to prototype our wireless camera that directly transmits analog values from the image sensor (sans amplification) using PWM backscatter. Photographs of our wireless camera prototype are shown in Fig. 5.14. The camera allows random access, i.e., any pixel on the camera can be accessed at random by setting the corresponding value in the row and column address registers. The camera can also be configured to output a single pixel, two adjacent pixels or a super-pixel with sizes ranging from 2×2 to 7×7. We use the random access and super-pixel functionality of the camera to implement our distributed inter-frame compression algorithm. The power consumption of our low-resolution off-the-shelf analog video streaming prototype is 2.36 mW. We emphasize that this off-the-shelf camera is only used to demonstrate the operational range; to achieve the tens of microwatts power budget we need to use our ASIC design.

Deployment results. We deploy our wireless camera system in the parking lot of an apartment complex. We use the USRP based reader implementation and set it to transmit 30 dBm into a 6 dBi patch antenna. We vary the distance between the wireless camera prototype and the reader and at each separation, we stream 20 seconds long video from the camera to the reader. Simultaneously, we also record the output of the camera using a high input impedance National Instrument USB-6361 DAQ as the ground truth. We use the popular PSNR metric commonly used in video applications to compare the video wirelessly streamed to the reader using PWM backscatter to the ground truth video recorded at the camera using an NI DAQ. PSNR computes the difference between the ground truth and wirelessly received video.

We measure the PSNR of the received video to evaluate the performance of the wireless camera under normal lighting conditions (below 300 lux) at a frame rate of 13 fps. To evaluate how much our sole wireless communication method impacts the quality of received video, we considered the output of the PWM converter as the ground truth for PSNR measurement. Also, to isolate the impact of AGC which is happened at the reader, unaltered received video by the reader prior to applying any AGC was compared to the ground truth for PSNR measurement. Fig. 5.15 plots the PSNR of the received video at the reader as a function of the separation between the reader and the wireless camera. The plots show the following. The wireless camera can stream video with an average PSNR greater than 24 dB to a reader up to a distance of 150 feet. Beyond 150 feet, the reader cannot reliably decode the sync pulses which limit the operating range of our wireless camera system. This shows that our analog backscatter approach can achieve significantly longer range for low-resolution video compared to the HD version of the camera due to the trade-off between bandwidth/data rate and operating distances.

Applying distributed inter-frame compression Algorithm on low resolution videos. We deploy this security camera in a normal lab space and then implemented our distributed inter-frame compression technique on the videos received. We evaluate the performance of algorithm for three different super pixels sizes of 3×3, 5×5 and 7×7 pixels and plots the results in Fig. 5.16. We achieve $29.4\times$ data reduction using our distributed inter-frame compression technique while maintaining a PSNR of greater than 30 dB.

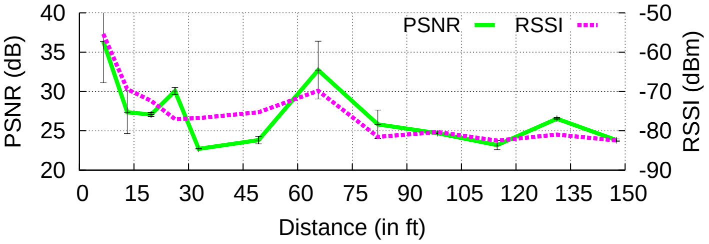


Figure 5.15: **Operational range of low-resolution security camera.**

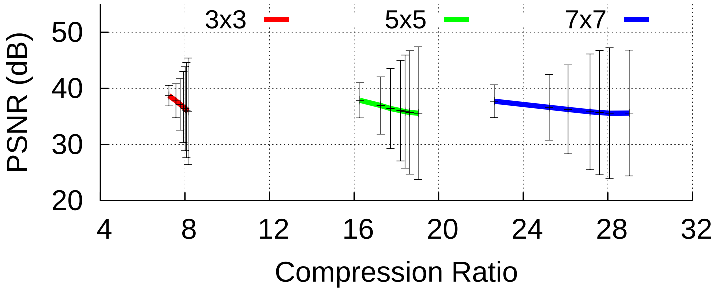


Figure 5.16: **Distributed interactive compression with our low-resolution security camera.** *PSNR* of the compressed video as a function of compression parameters in a typical lab setting.

Face detection accuracy. Next, we demonstrate that the quality of the video streamed from our low resolution COTS implementation is sufficient for detecting human faces. Such a system can be used to detect human occupancy, grant access (such as Ring [17]) or set off an alarm in case of an intruder. To evaluate the system, we place the wireless camera at five different distances ranging from 16 to 100 feet from the reader. We ask ten users to walk around and perform gestures within 5 feet of the camera. We stream a 2 minute video at each location at 13 fps. We use the MATLAB implementation of Viola-Jones algorithm to analyze the approximately four thousand video frames. Fig. 5.17 shows the accuracy of face detection as a function of the PSNR of the wirelessly recorded video. The plots show that as the quality (PSNR) of the video improves, the accuracy of face detection also increases. We can accurately detect up to 95% of human faces when PSNR is greater than 30 dB.

5.7 Related Work

Prior work falls in two different categories.

Backscatter Communication. One of the early examples of analog backscatter was a gift by the Russians to the US embassy in Moscow that included a passive listening device. This spy device consisted of a sound-modulated resonant cavity. The voice moved the diaphragm to modulate the resonance frequency of the cavity which could be detected by analyzing the RF signals reflected by the cavity. [8]. A more recent example of analog backscatter is a microphone enabled battery-free tag that amplitude modulates its antenna impedance with the output of the microphone [118, 114]. In contrast, we design the first analog video backscatter system. Further, prior microphone designs have a low data-rate compared to video streaming. Our camera at 10 frames per second transmits about $9.2M$ pixels per second whereas for a microphone, a few kilo-samples of audio transmission is sufficient to fully recover the voice. In addition, our 13 fps $163K$ pixels per second camera operates at more than four times the range of the microphone in [118] due to our pulse-width modulation implementation.

Ekhonet [133] optimizes the computational blocks between the sensor and the backscatter module to reduce the power consumption of backscatter-based wireless sensors. Our design builds on this work but differs from it in multiple ways: 1) prior work still uses ADCs and amplifiers on the cameras to transform the pixels into the digital domain and hence cannot achieve streaming video on the limited harvesting power budget. In contrast we provide the first architecture for battery-free video streaming by designing an analog video backscatter solution.

Recent work on Wi-Fi and TV based backscatter systems [59, 60, 55, 72, 96] can achieve mega bits per second of communication speed using backscatter technique. Integrating these designs with our video backscatter approach is a worthwhile engineering effort.

Low-Power Cameras. [92] introduces a self-powered camera that can switch its photo diodes between energy harvesting and photo capture mode. Despite being self-powered, these cameras do

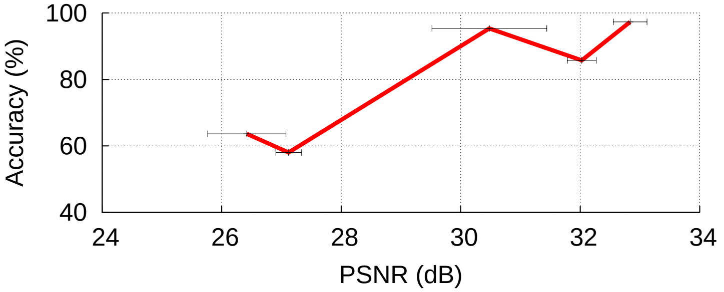


Figure 5.17: **Face detection with our low-resolution security camera.** We show the accuracy of face detection on the video streamed from our wireless camera.

not have wireless data transmission capabilities. [117, 88, 91, 85, 86, 87] show that using off-the-shelf low-resolution camera modules, one can build battery-free wireless cameras that will capture still images using the energy they have harvested from RF waves including Wi-Fi and 900 MHz transmissions. Despite their ability to transmit its data wirelessly, they are heavily duty cycled and cannot stream video. In particular, these designs can send a new frame at most every ten seconds when it is very close to the RF power source (within about a foot) and once every few tens of minutes at longer distances [117].

[76] presents a 90×90 pixels image sensor that its pixels are sensitive to changes in the environment. If enough illumination variation is received by a pixel, the address of that pixel will be stored in a FIFO, thus compresses the image to the pixels that have significantly changed. Despite enabling image compression right at the image sensor, this system does not stream live video. In addition, at this low resolution, it burns about 3mW of power when running at 30 fps. [67] introduces a 128×128 pixels event-driven image sensor with emphasis on low-latency for detecting very fast moving scenes so the power consumption is orders of magnitude higher than our system.

[68] addresses the problem that conventional image sensors' power consumption does not scale as their resolution and frame rate. In particular, the authors propose to change the camera input clock as well as aggressively switch the camera to standby mode, based on the desired image quality. However streaming video requires addressing the power consumption of multiple components

including camera, communication as well as compression. Our work jointly integrates all these components to achieve the first battery-free video streaming design.

Finally, [89] showed that a regular camera on wearable devices burn more than 1200 mW, which limits the operation time of the camera to less than two hours on a wearable device. They instead design a low-power gating wearable vision system that only looks for certain events occurrence in the field of view, and turns on the primary imaging pipeline when those events happened. The power and bandwidth savings however are limited to the application and does not deal with communication. In contrast, we present the first battery-free video streaming application by jointly optimizing backscatter communication and camera design and eliminating power-consuming interfaces such as ADCs and amplifiers.

5.8 Limitations and Conclusion

This chapter takes a significant first step in designing video streaming for battery-free devices. In this section, we discuss limitations and a few avenues for future research.

Security. Our current implementation does not account for security. However, to secure the wireless link between the camera and reader, we can leverage that our digital core processes the PWM signal. Each wireless camera can be assigned a unique pseudo random security key. Based on the random security key, the digital core of the camera can modulate the width of the PWM encoded pixel value using XOR gate. The reader, which knows the security key, can map the received data to the desired pixel values by performing the analogous operation.

ASIC versus off-the-shelf. While existing work on backscatter cameras focus on using off-the-shelf components, they treat cameras and backscatter independently and just interface the two. Thus, they cannot achieve video streaming and the low-power demonstrated in this chapter. The key contribution of this work is to make a case for a joint camera and backscatter architecture and designing the first analog video backscatter solution. Such a joint architecture however cannot be achieved with off-the-shelf components. The cost of building ICs in general in a research environment is prohibitively high. We believe that we spec out the IC design in §5.4 to enough detail using industry standard EDA tools that it can be taken out of the lab to the industry.

Mobile device as reader. To support our design for HD video streaming from wearable cameras, the smartphone has to support a backscatter reader. We can use RFID readers that can be plugged into the headjack of the smartphone [9] to achieve this. In the future, we would believe that backscatter readers would be integrated into smartphones to support video and other applications [55].

Enabling concurrent streaming. Our analog backscatter camera design can use frequency division multiplexing techniques to share the wireless medium across multiple devices. The reader can co-ordinate communication by assigning different frequency channels to each camera. Multiple cameras can simultaneously backscatter in different channels by using different (assigned) frequency offsets in our sideband modulation. Evaluating this however is not in the scope of this chapter.

Suppressing EMI. We outline two approaches to suppress the minuscule EMI generated by harmonics of baseband PWM signal. A PWM conversion with full dynamic range generates harmonics which are about 4 dB lower than the desired baseband signal. However, if we increase the dynamic range by a factor of 2, the PWM harmonics reduce to 10 dB lower than the baseband. So, we can tradeoff harmonic suppression for slightly reduced performance. This is especially useful since EMI due to harmonics is a potential issue when the camera is close to the reader where it can re-radiate higher power EMI. However, at close distances we can safely tradeoff a bit of performance for EMI rejection. At farther distances, since the backscattered power is low, EMI is not an issue and we don't need to tradeoff any performance. Finally, in addition to performance tradeoff, we can introduce additional low power pass filter after the sub-carrier modulation operation in the digital domain to suppress harmonics and transmit filtered signal using backscatter.

Chapter 6

THESIS CONCLUSION AND FUTURE WORK

In this chapter of my thesis I present the future directions of this research and finish by concluding this dissertation.

6.1 Future Work

The work presented in this thesis can be advanced in the future along the below proposed directions:

- **Exploring delta-sigma modulators instead of PWM converters for ultra-low-power video streaming.** Delta-sigma modulators are used at the input of delta-sigma ADCs which have been known as an efficient and high resolution ADC architecture for many years. Delta-sigma modulators convert an analog input signal into a stream of zeros and ones, in such a way that as the analog input signal becomes larger, the density of ones in the output becomes greater. One of the interesting benefits of delta-sigma modulators is shifting the noise power more toward the higher frequencies while reducing the noise power in the lower frequencies (baseband), called **noise shaping**. **Noise shaping** can potentially increase the SNR of the underlying baseband signal. There is a significant analogy between delta-sigma modulators and PWM converters. So it is worthwhile to explore how the video streaming performance might change if a delta-sigma modulator is used instead of a PWM converter.

- **Generating PWM signal using a counter.** Another approach instead of using the PWM modulator explained in chapter §5, is to use an ADC to digitize the analog pixels' raw voltages and then use a counter to create an output pulse that its width is proportional to the pixel value. Although this approach requires an ADC, due to inherent correlation present in adjacent pixels of a frame we can operate the ADC at a much lower conversion rate than the rate at which conventional cameras operate their ADCs. It is an interesting research project to understand what are the trade-offs in terms of video quality and power consumption when employing this alternative approach.

- **Change driven image sensor design.** The intra-frame compression explained in chapter §5 leverages the fact that adjacent pixels do not change significantly. Building on top of this, in the future we can implement image sensors that only output the pixels which have significantly changed. This approach reduces the effective pixel rate generated by the camera.

- **ASIC fabrication of the design presented in chapter §5.** In chapter §5 we designed and simulated an HD video streaming wireless camera to estimate the power consumption of our wireless camera architecture but the actual ASIC was not fabricated. We believe taping-out the ASIC version of our HD camera architecture can demonstrate the full capacity of our system and enable several new research directions that rely on energy-constrained wireless video streaming cameras such as tiny flies and micro-robots. It can also enable very power constrained IoT platforms with machine vision capabilities by imposing negligible power overhead on them.

- **Design and build true battery-free wireless HD video streaming cameras.** Although in this dissertation the feasibility of developing battery-free HD video streaming cameras demonstrated, we only built a 112×112 pixels resolution battery-free version of our new camera architecture. So designing a reliable battery-free HD video streaming is an interesting potential future work that can open up new research avenues for machine vision and IoT domains. These battery-free cameras can harvest from multiple sources of energy and combine their power together to improve their performance, for example they can harvest from both RF signals and solar.

- **Develop custom designed and low-cost readers that not only talk to a massive network of wireless battery-free video streaming cameras but also transmit FCC-compliant RF signal to power them.** Developing these readers powers a network of low-power wireless video streaming cameras through its transmitted RF signal; the readers can query different cameras and provide a two-way communication between camera and the user application; these readers are low-cost and can enable real massive deployment of low-power wireless HD video streaming cameras which is an important infrastructure for future research directions.

- **Develop a hybrid analog/digital encryption tailored for these custom designed low-power cameras.** As of now, there exists no pure analog encryption. Given how important is the privacy and security of wireless video stream, one interesting future work is to devote substantial effort to develop the first hybrid analog/digital video stream encryption. The new encryption algorithms will build on top of already existing sophisticated digital encryption algorithms and extend them to our targeted scenarios that involve PWM-encoded video streams, a hybrid of analog and digital streams. Developing the encryption algorithm can turn these low-power cameras into secured and trusted sources of information streamed from our environment.

6.2 Thesis Conclusion

In this thesis we showed multiple ways to reduce the power consumption of wireless cameras that are conventionally known as power hungry devices. The primary focus of this dissertation was to exploit how we can, for the first time, enable battery-free HD video streaming. Three main approaches were presented and discussed. Our prior research publications validated and characterized our approaches. First, we developed *WISPCam* and showed that we can take advantage of **duty-cycling** to bring the average power consumption of a wireless camera within the battery-free power budget range. Then Glimpse camera presented which exploited how we can leverage **gating** (basically a smart duty-cycling approach) to significantly reduce the power consumption of a wireless wearable camera but it was not low enough to enable battery-free video streaming. And at the end we showed how we can **redesign the whole wireless camera architecture** to develop the world's lowest power wireless HD video streaming camera. We showed our wireless camera system burns as low as $\sim 250\mu W$ when streaming grayscale 30 fps 720p HD video.

BIBLIOGRAPHY

[1] 6-dBi gain patch antenna. `https://www.arcantenna.com/index.php/product_documents/get/document/id/492/`. Accessed: March 2017.

[2] Ambarella a9 ultra hd 4k camera soc product brief. `http://www.ambarella.com/uploads/docs/A9-product-brief.pdf`.

[3] Avx bestcap datasheet. `http://www.avx.com/docs/Catalogs/bestcap.pdf`.

[4] Barker code. `http://mathworld.wolfram.com/BarkerCode.html`. Accessed: September 2017.

[5] Bq25570 datasheet. `http://www.ti.com/product/BQ25570/datasheet/`.

[6] CentEye Stonyman image sensor datasheet. `http://www.centeye.com/products/current-centeye-vision-chips/`. Accessed: March 2017.

[7] Farsens battery-free sensor solutions. `http://www.farsens.com/en/battery-free-sensor-solutions`. Accessed December 2014.

[8] The great seal bug. `http://counterespionage.com/the-great-seal-bug-part-1.html`. Accessed: March 2017.

[9] Head jack pluggable rfid reader for smartphones. `http://www.rfidtagworld.com/products/Ear-Jack-reader-UHF_1056.html`.

[10] IGLOO Nano FPGA datasheet. `https://www.microsemi.com/document-portal/doc_view/130695-ds0110-igloo-nano-low-power-flash-fpgas-datasheet`. Accessed: March 2017.

[11] Lmv7219 comparator. `http://www.ti.com/lit/ds/symlink/lmv7219.pdfl`. Accessed: September 2017.

[12] Long distance measuring sensor datasheet. `http://www.sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a02yk_e.pdf`. Accessed: May 2017.

[13] Moving camera on a motorcycle video clip. `https://www.youtube.com/watch?v=sHj3xSG-R_E&t=376s`.

[14] MSP430 flash memory characteristics. `http://www.ti.com/lit/an/slaa334a/slaa334a.pdf`. Accessed December 2014.

[15] NCX2200I low power comparator datasheet. `http://www.nxp.com/documents/data_sheet/NCX2200.pdf`. Accessed: March 2017.

[16] RF5110 amplifier. `http://www.rfmd.com/store/downloads/dl/file/id/30508/5110g_product_data_sheet.pdf`. Accessed: March 2017.

[17] Ring video doorbell. `https://ring.com/`. Accessed: March 2017.

[18] sllurp – Python client for LLRP-based RFID readers. `https://github.com/ransford/sllurp`. Accessed December 2014.

[19] Spectacles battery life. `http://www.barrons.com/articles/snapchat-spectacles-review-the-good-the-bad-the-revolutionary-1487846715`.

[20] Stereographic depth mapping on an fpga. `https://courses.cit.cornell.edu/ece576/FinalProjects/f2010/pfk5_jk459/pfk5_jk459/index.html`.

[21] Synopsis design complier. `http://www.synopsys.com/Tools/Implementation/RTLSynthesis/DesignCompiler/Pages/default.aspx`.

[22] Tire pressure and brake temperature systems – smartstem. `http://www.craneae.com/Products/Sensing/SmartStem.aspx`. Accessed December 2014.

[23] USRP X-300. `https://www.ettus.com/product/details/X300-KIT`. Accessed: March 2017.

[24] Wearable spectacles. `http://www.techradar.com/reviews/snap-spectacles`.

[25] Wisp 5 firmware repository. `http://www.github.com/wisp/`. Accessed December 2014.

[26] Youtube website. `http://www.youtube.com`.

[27] Human factors issues in aircraft maintenance and inspection. `https://tinyurl.com/ycqt5m7z`, October 1989. Accessed December 2014.

[28] The battery in my ring device seems to drain too fast. `http://www.faa.gov/`, February 2016. Accessed Feb 2018.

[29] Sharad Agarwal, Matthai Philipose, and Paramvir Bahl. Vision: The case for cellular small cells for cloudlets. In *International Workshop on Mobile Cloud Computing and Services*, 2014.

[30] Suat U Ay. A 1.32 pw/frame pixel 1.2 v cmos energy-harvesting and imaging (ehi) aps imager. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pages 116–118. IEEE, 2011.

[31] Paramvir Bahl, Matthai Philipose, and Lin Zhong. Vision: cloud-powered sight for all: showing the cloud what you see. In *Proceedings of the third ACM workshop on Mobile cloud computing and services*, pages 53–60. ACM, 2012.

[32] Sven Bambach, John M Franchak, David J Crandall, and Chen Yu. Detecting hands in childrens egocentric views to understand embodied attention during social interaction. *Proceedings of the 36th Annual Conference of the Cognitive Science Society (pp. 134-139). Quebec City, Canda: Cognitive Science Society*, pages 134–139, 2014.

[33] H Randall Beard, Alejandro J Marquez-Lara, and Kamran S Hamid. Using wearable video technology to build a point-of-view surgical education library. *JAMA surgery*, 151(8):771–772, 2016.

[34] Steven R Best and Bradley C Kaanta. A tutorial on the receiving and scattering properties of antennas. *IEEE Antennas and Propagation Magazine*, 51(5), 2009.

[35] S. Blessenohl, C. Morrison, A. Criminisi, and J. Shotton. Improving indoor mobility of the visually impaired with depth-based spatial sound. In *ICCV-ACVR workshop*, December 2015.

[36] Michael Buettner, Richa Prasad, Matthai Philipose, and David Wetherall. Recognizing daily activities with RFID-based sensors. In *UbiComp*, September 2009.

[37] Long Chen, Xiyuan Tang, Arindam Sanyal, Yeonam Yoon, Jie Cong, and Nan Sun. A 0.7-v 0.6-uw $100-ks/s$ low-power sar adc with statistical estimation-based noise reduction. *IEEE Journal of Solid-State Circuits*, 52(5):1388–1398, 2017.

[38] Tiffany Yu-Han Chen, Lenin Ravindranath, Shuo Deng, Paramvir Bahl, and Hari Balakrishnan. Glimpse: Continuous, real-time object recognition on mobile devices. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, SenSys 2015, Seoul, South Korea, November 1-4, 2015*, pages 155–168, 2015.

[39] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893, 2005.

[40] Y. Deng, S. Chakrabartty, and G. Cauwenberghs. Analog auditory perception model for robust speech recognition. In *IEEE International Joint Conference on Neural Networks*, 2004.

[41] E. L. Dereniak and G. D. Boreman. *Infared Detectors and Systems*. Wiley, second edition, 1996.

[42] Varick L Erickson, Alex Beltran, Daniel A Winkler, Niloufar P Esfahani, John R Lusby, and Alberto E Cerpa. Toss: Thermal occupancy sensing system. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pages 1–2. ACM, 2013.

[43] Gabriel Evans, Jack Miller, Mariangely Iglesias Pena, Anastacia MacAllister, and Eliot Winer. Evaluating the microsoft hololens through an augmented reality assembly application. In *Degraded Environments: Sensing, Processing, and Display 2017*, volume 10197, page 101970V. International Society for Optics and Photonics, 2017.

[44] Rafael C Gonzalez and Richard E Woods. *Image processing*, volume 2. 2007.

[45] Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai, and Mahadev Satyanarayanan. Towards wearable cognitive assistance. In *MobiSys*, 2014.

[46] Seungyeop Han, Rajalakshmi Nandakumar, Matthai Philipose, Arvind Krishnamurthy, and David Wetherall. Glimpsedata: Towards continuous vision-based personal analytics. In *Proceedings of the 2014 workshop on physical analytics*, pages 31–36. ACM, 2014.

[47] Seungyeop Han, Haichen Shen, Matthai Philipose, Sharad Agarwal, Alec Wolman, and Arvind Krishnamurthy. MCDNN:An Approximation-Based Execution Framework for Deep Stream Processing Under Resource Constraints. In *Proceedings of the 14th International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM, 2016.

[48] Scott Hanson, ZhiYoong Foo, David Blaauw, and Dennis Sylvester. A 0.5 v sub-microwatt cmos image sensor with pulse-width modulation read-out. *IEEE Journal of Solid-State Circuits*, 45(4):759–767, 2010.

[49] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M. M. Cheng, S. L. Hicks, and P. H. S. Torr. Struck: Structured output tracking with kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2096–2109, Oct 2016.

[50] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[51] Jennifer Healey and Rosalind W. Picard. Startlecam: A cybernetic wearable camera. In *Second International Symposium on Wearable Computers (ISWC 1998)*, pages 42–49, 1998.

[52] Peter Hevesi, Sebastian Wille, Gerald Pirkl, Norbert Wehn, and Paul Lukowicz. Monitoring household activities and user location with a cheap, unobtrusive thermal sensor array. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 141–145. ACM, 2014.

[53] Steve Hodges, Lyndsay Williams, Emma Berry, Shahram Izadi, James Srinivasan, Alex Butler, Gavin Smyth, Narinder Kapur, and Kenneth R. Wood. Sensecam: A retrospective memory aid. In *UbiComp 2006*, pages 177–193, 2006.

[54] Jyrki Hoisko. Context triggered visual episodic memory prosthesis. In *Fourth International Symposium on Wearable Computers (ISWC200*.

[55] Vikram Iyer, Vamsi Talla, Bryce Kellogg, Shyamnath Gollakota, and Joshua Smith. Inter-technology backscatter: Towards internet connectivity for implanted devices. In *Proceedings of the 2016 Conference on ACM SIGCOMM 2016 Conference*, SIGCOMM '16, pages 356–369, New York, NY, USA, 2016. ACM.

[56] K Kagawau, Sanshiro Shishido, Masahiro Nunoshita, and Jun Ohta. A 3.6 pw/frameâ· pixel 1.35 v pwm cmos imager with dynamic pixel readout and no static bias current. In *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, pages 54–595. IEEE, 2008.

[57] Takeo Kanade and Martial Hebert. First-person vision. *Proceedings of the IEEE*, 100(8):2442–2453, 2012.

[58] Kerem Kapucu and Catherine Dehollain. A passive uhf rfid system with a low-power capacitive sensor interface. In *RFID Technology and Applications Conference (RFID-TA), 2014 IEEE*, pages 301–305. IEEE, 2014.

[59] Bryce Kellogg, Aaron Parks, Shyamnath Gollakota, Joshua R. Smith, and David Wetherall. Wi-fi backscatter: Internet connectivity for rf-powered devices. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, 2014.

[60] Bryce Kellogg, Vamsi Talla, Shyamnath Gollakota, and Joshua Smith. Passive wi-fi: Bringing low power to wi-fi transmissions. In *Usenix NSDI*, 2016.

[61] Walt Kester. Understand sinad, enob, snr, thd, thd+ n, and sfdr so you don't get lost in the noise floor. *MT-003 Tutorial, www. analog. com/static/importedfiles/tutorials/MT-003. pdf*, 2009.

[62] S. Kim and P. H. Chou. Size and topology optimization for supercapacitor-based sub-watt energy harvesters. *Power Electronics, IEEE Transactions on*, 28(4):2068–2080, April 2013.

[63] Krishna Reddy Konda, Andrea Rosani, Nicola Conci, and Francesco GB De Natale. Smart camera reconfiguration in assisted home environments for elderly care. In *European Conference on Computer Vision*, pages 45–58. Springer, 2014.

[64] J.G. Koomey, S. Berard, M. Sanchez, and H. Wong. Implications of historical trends in the electrical efficiency of computing. *Annals of the History of Computing, IEEE*, 33(3):46–54, March 2011.

[65] Sreekar Krishna, Greg Little, John Black, and Sethuraman Panchanathan. A wearable face recognition system for individuals with visual impairments. In *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '05, pages 106–113, New York, NY, USA, 2005. ACM.

[66] Nalpantidis Lazaros, Georgios Christou Sirakoulis, and Antonios Gasteratos. Review of stereo vision algorithms: From software to hardware. *International Journal of Optomechatronics*, 2(4):435–462, 2008.

[67] Juan Antonio Leñero-Bardallo, Teresa Serrano-Gotarredona, and Bernabé Linares-Barranco. A 3.6us latency asynchronous frame-free event-driven dynamic-vision-sensor. *IEEE Journal of Solid-State Circuits*, 46(6):1443–1455, 2011.

[68] Robert LiKamWa, Bodhi Priyantha, Matthai Philipose, Lin Zhong, and Paramvir Bahl. Energy characterization and optimization of image sensing toward continuous mobile vision. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 69–82. ACM, 2013.

[69] Robert LiKamWa, Zhen Wang, Aaron Carroll, Felix Xiaozhu Lin, and Lin Zhong. Draining our glass: An energy and heat characterization of google glass. *arXiv preprint arXiv:1404.1320*, 2014.

[70] Robert LiKamWa and Lin Zhong. Starfish: Efficient concurrency support for computer vision applications. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 213–226. ACM, 2015.

[71] Haowei Liu, Matthai Philipose, Martin Pettersson, and Ming-Ting Sun. Recognizing object manipulation activities using depth and visual cues. *Journal of Visual Communication and Image Representation*, 25(4):719–726, 2014.

[72] Vincent Liu, Vamsi Talla, and Shyamnath Gollakota. Enabling instantaneous feedback with full-duplex backscatter. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom ’14.

[73] Vincent Liu, Vamsi Talla, and Shyamnath Gollakota. Enabling instantaneous feedback with full-duplex backscatter. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 67–78. ACM, 2014.

[74] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.

[75] Hong Lu, AJ Bernheim Brush, Bodhi Priyantha, Amy K Karlson, and Jie Liu. Speakersense: Energy efficient unobtrusive speaker identification on mobile phones. In *International Conference on Pervasive Computing*, pages 188–205. Springer, 2011.

[76] U Mallik, M Clapp, Edward Choi, G Cauwenberghs, and R Etienne-Cummings. Temporal change threshold detection imager. In *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International*, pages 362–603. IEEE, 2005.

[77] Bappaditya Mandal, Shue-Ching Chia, Liyuan Li, Vijay Chandrasekhar, Cheston Tan, and Joo-Hwee Lim. A wearable face recognition system on google glass for assisting social interactions. In *Computer Vision - ACCV 2014 Workshops - Singapore, Singapore, November 1-2, 2014, Revised Selected Papers, Part III*, pages 419–433, 2014.

[78] Tomasz Marciniak, Agata Chmielewska, Radoslaw Weychan, Marianna Parzych, and Adam Dabrowski. Influence of low resolution of images on reliability of face detection and recognition. *Multimedia Tools and Applications*, 74(12):4329–4349, 2015.

[79] Gaurav Mathur, Peter Desnoyers, Paul Chukiu, Deepak Ganesan, and Prashant Shenoy. Ultra-low power data storage for sensor networks. *ACM Trans. Sen. Netw.*, 5(4):33:1–33:34, November 2009.

[80] Fridolin Michel and Michiel Steyaert. A 250mv 7.5 μw 61db sndr cmos sc $\delta\sigma$ modulator using a near-threshold-voltage-biased cmos inverter technique. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pages 476–478. IEEE, 2011.

[81] Oliver J Muensterer, Martin Lacher, Christoph Zoeller, Matthew Bronstein, and Joachim Kübler. Google glass in pediatric surgery: an exploratory study. *International journal of surgery*, 12(4):281–289, 2014.

[82] B. Murmann. Adc performance survey 1997-2017. `http://web.stanford.edu/~murmann/adcsurvey.html`.

[83] Chris Murphy, Daniel Lindquist, Ann Marie Rynning, Thomas Cecil, Sarah Leavitt, and Mark L Chang. Low-cost stereo vision on an fpga. In *Field-Programmable Custom Computing Machines, 2007. FCCM 2007. 15th Annual IEEE Symposium on*, pages 333–334. IEEE, 2007.

[84] S. Naderiparizi, A.N. Parks, Z. Kapetanovic, B. Ransford, and J.R. Smith. Wispcam: A battery-free rfid camera. In *RFID (RFID), 2015 IEEE International Conference on*, pages 166–173, April 2015.

[85] Saman Naderiparizi, Zerina Kapetanovic, and Joshua R Smith. Battery-free connected machine vision with wispcam. *GetMobile: Mobile Computing and Communications*, 20(1):10–13, 2016.

[86] Saman Naderiparizi, Zerina Kapetanovic, and Joshua R Smith. Wispcam: An rf-powered smart camera for machine vision applications. In *Proceedings of the 4th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*, pages 19–22. ACM, 2016.

[87] Saman Naderiparizi, Zerina Kapetanovic, and Joshua R Smith. Rf-powered, backscatter-based cameras. In *Antennas and Propagation (EUCAP), 2017 11th European Conference on*, pages 346–349. IEEE, 2017.

[88] Saman Naderiparizi, Aaron N Parks, Zerina Kapetanovic, Benjamin Ransford, and Joshua R Smith. Wispcam: A battery-free rfid camera. In *RFID (RFID), 2015 IEEE International Conference on*, pages 166–173. IEEE, 2015.

[89] Saman Naderiparizi, Pengyu Zhang, Matthai Philipose, Bodhi Priyantha, Jie Liu, and Deepak Ganesan. Glimpse: A programmable early-discard camera architecture for continuous mobile vision. In *Proceeding of the 15th annual international conference on Mobile systems, applications, and services*. ACM, 2017.

[90] Saman Naderiparizi, Yi Zhao, James Youngquist, Alanson P. Sample, and Joshua R. Smith. Self-localizing battery-free cameras. In *Proceedings of the 2015 ACM International Joint*

Conference on Pervasive and Ubiquitous Computing, UbiComp '15, pages 445–449, New York, NY, USA, 2015. ACM.

[91] Saman Naderiparizi, Yi Zhao, James Youngquist, Alanson P Sample, and Joshua R Smith. Self-localizing battery-free cameras. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 445–449. ACM, 2015.

[92] Shree K Nayar, Daniel C Sims, and Mikhail Fridberg. Towards self-powered cameras. In *Computational Photography (ICCP), 2015 IEEE International Conference on*, pages 1–10. IEEE, 2015.

[93] G. Nebehay and R. Pflugfelder. Consensus-based matching and tracking of keypoints for object tracking. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 862–869, March 2014.

[94] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, Jan 1979.

[95] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, 2015.

[96] Aaron N. Parks, Angli Liu, Shyamnath Gollakota, and Joshua R. Smith. Turbocharging ambient backscatter communication. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, 2014.

[97] Aaron N Parks, Alanson P Sample, Yi Zhao, and Joshua R Smith. A wireless sensing platform utilizing ambient rf energy. In *Biomedical Wireless Technologies, Networks, and Sensing Systems (BioWireleSS), 2013 IEEE Topical Conference on*, pages 154–156. IEEE, 2013.

[98] Matthai Philipose, Joshua R Smith, Bing Jiang, Alexander Mamishev, Sumit Roy, and Kishore Sundara-Rajan. Battery-free wireless identification and sensing. *Pervasive Computing, IEEE*, 4(1):37–45, 2005.

[99] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2847–2854. IEEE, 2012.

[100] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1201–1208. IEEE, 2011.

[101] Moo-Ryong Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan. Odessa: Enabling interactive perception applications on mobile devices. In *Mobisys*, 2011.

[102] Swati Rallapalli, Aishwarya Ganesan, Krishna Chintalapudi, Venkat N Padmanabhan, and Lili Qiu. Enabling physical analytics in retail stores using smart glasses. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 115–126. ACM, 2014.

[103] Mastooreh Salajegheh, Shane S. Clark, Benjamin Ransford, Kevin Fu, and Ari Juels. CCCP: secure remote storage for computational RFIDs. In *Proceedings of the 18th USENIX Security Symposium*, pages 215–230, Montreal, Canada, August 2009.

[104] A. Sample and J.R. Smith. Experimental results with two wireless power transfer systems. In *Radio and Wireless Symposium, 2009. RWS '09. IEEE*, pages 16–18, 2009.

[105] Alanson P Sample, Daniel J Yeager, Pauline S Powledge, Alexander V Mamishev, and Joshua R Smith. Design of an rfid-based battery-free programmable sensing platform. *IEEE Transactions on Instrumentation and Measurement*, 57(11):2608–2615, 2008.

[106] A.P. Sample, D.J. Yeager, P.S. Powledge, A.V. Mamishev, and J.R. Smith. Design of an rfid-based battery-free programmable sensing platform. *Instrumentation and Measurement, IEEE Transactions on*, 57(11):2608–2615, Nov 2008.

[107] A.P. Sample, D.J. Yeager, and J.R. Smith. A capacitive touch interface for passive rfid tags. In *RFID, 2009 IEEE International Conference on*, pages 103–109, April 2009.

[108] Rahul Sarpeshkar. Analog versus digital: extrapolating from electronics to neurobiology. *Neural computation*, 10(7):1601–1638, 1998.

[109] Michael D Scott, Bernhard E Boser, and Kristofer SJ Pister. An ultra–low power adc for distributed sensor networks. In *Solid-State Circuits Conference, 2002. ESSCIRC 2002. Proceedings of the 28th European*, pages 255–258. IEEE, 2002.

[110] Joshua R Smith. Range scaling of wirelessly powered sensor systems. In *Wirelessly powered sensor networks and computational RFID*, pages 3–12. Springer, 2013.

[111] Joshua R Smith, Alanson P Sample, Pauline S Powledge, Sumit Roy, and Alexander Mamishev. A wirelessly-powered platform for sensing and computation. In *UbiComp 2006: Ubiquitous Computing*, pages 495–506. Springer Berlin Heidelberg, 2006.

[112] Joshua R Smith, Alanson P Sample, Pauline S Powledge, Sumit Roy, and Alexander Mamishev. A wirelessly-powered platform for sensing and computation. In *UbiComp 2006: Ubiquitous Computing*, pages 495–506. Springer, 2006.

[113] Aslam Forhad Symon, Nazia Hassan, Humayun Rashid, Iftekhar Uddin Ahmed, and SM Taslim Reza. Design and development of a smart baby monitoring system based on raspberry pi and pi camera. In *Advances in Electrical Engineering (ICAEE), 2017 4th International Conference on*, pages 117–122. IEEE, 2017.

[114] Vamsi Talla, Bryce Kellogg, Shyamnath Gollakota, and Joshua R Smith. Battery-free cell phone. ACM UBIComp, 2017.

[115] Vamsi Talla, Bryce Kellogg, Benjamin Ransford, Saman Naderiparizi, Shyamnath Gollakota, and Joshua R. Smith. Powering the next billion devices with wi-fi. *CoRR*, abs/1505.06815, 2015.

[116] Vamsi Talla, Bryce Kellogg, Benjamin Ransford, Saman Naderiparizi, Shyamnath Gollakota, and Joshua R Smith. Powering the next billion devices with wi-fi. In *Proceedings of*

the 11th ACM Conference on Emerging Networking Experiments and Technologies, page 4. ACM, 2015.

[117] Vamsi Talla, Bryce Kellogg, Benjamin Ransford, Saman Naderiparizi, Shyamnath Gollakota, and Joshua R. Smith. Powering the next billion devices with wi-fi. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT '15, pages 4:1–4:13, New York, NY, USA, 2015. ACM.

[118] Vamsi Talla and Joshua R Smith. Hybrid analog-digital backscatter: A new approach for battery-free sensing. In *RFID (RFID), 2013 IEEE International Conference on*, pages 74–81. IEEE, 2013.

[119] Ngo Huy Tan, Nor Hisham Hamid, Patrick Sebastian, and Yap Vooi Voon. Resource minimization in a real-time depth-map processing system on fpga. In *TENCON 2011-2011 IEEE Region 10 Conference*, pages 706–710. IEEE, 2011.

[120] Fang Tang and Amine Bermak. An 84 pw/frame per pixel current-mode cmos image sensor with energy harvesting capability. *IEEE Sensors Journal*, 12(4):720–726, 2012.

[121] Ruxandra Tapu, Bogdan Mocanu, and Titus B. Zaharia. ALICE: A smartphone assistant used to increase the mobility of visual impaired people. *JAISE*, 7(5):659–678, 2015.

[122] S.J. Thomas, T. Deyle, R. Harrison, and M.S. Reynolds. Rich-media tags: Battery-free wireless multichannel digital audio and image transmission with uhf rfid techniques. In *RFID (RFID), 2013 IEEE International Conference on*, pages 231–236, April 2013.

[123] Roger Y Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, 1987.

[124] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[125] Jingdong Wang, Changshui Zhang, and Heung-Yeung Shum. Face image resolution versus face recognition performance based on two global methods. In *Proceedings of Asia Conference on Computer Vision*, volume 47, pages 48–49. Citeseer, 2004.

[126] Qiang Wang, Jing Wu, Chengnian Long, and Bo Li. P-fad: Real-time face detection scheme on embedded smart cameras. In *Distributed Smart Cameras (ICDSC), 2012 Sixth International Conference on*, pages 1–6, Oct 2012.

[127] A.S. Weddell, Geoff V. Merrett, T.J. Kazmierski, and B.M. Al-Hashimi. Accurate supercapacitor modeling for energy harvesting wireless sensor nodes. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 58(12):911–915, December 2011.

[128] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg. A scalable approach to activity recognition based on object use. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[129] Zhixiang Xu, Matt Kusner, Minmin Chen, and Kilian Q. Weinberger. Cost-sensitive tree of classifiers. In *ICML*, 2013.

[130] Yaru Yan, Dong Wu, Huijuan Liu, Liyang Pan, and Jun Xu. A pure logic cmos based low power non-volatile random access memory for rfid application. In *ASIC, 2009. ASICON '09. IEEE 8th International Conference on*, pages 1015–1018, October 2009.

[131] Daniel J Yeager, Jeremy Holleman, Richa Prasad, Joshua R Smith, and Brian P Otis. Neuralwisp: A wirelessly powered neural interface with 1-m range. *Biomedical Circuits and Systems, IEEE Transactions on*, 3(6):379–387, 2009.

[132] Daniel J Yeager, Pauline S Powledge, Richa Prasad, David Wetherall, and Joshua R Smith. Wirelessly-charged UHF tags for sensor data collection. In *RFID, 2008 IEEE International Conference on*, pages 320–327. IEEE, 2008.

[133] Pengyu Zhang, Pan Hu, Vijay Pasikanti, and Deepak Ganesan. Ekhonet: High speed ultra low-power backscatter for next generation sensors. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 557–568. ACM, 2014.

[134] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

[135] Zhengyou Zhang. Camera calibration with one-dimensional objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(7):892–899, 2004.