System Specific Power Reduction Techniques for Wearable Navigation Technology

by

Ishwarya Ananthabhotla

S.B., Massachusetts Institute of Technology (2015)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Author .....
Department of Electrical Engineering and Computer Science
May 20, 2016

Certified by .....
Anantha P. Chandrakasan
Joseph F. and Nancy P. Keithley Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Christopher J. Terman
Chairman, Masters of Engineering Thesis Committee

System Specific Power Reduction Techniques for Wearable Navigation Technology

by

Ishwarya Ananthabhotla

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 2016, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

As a result of advances in computer vision, mapping, and controls, wearable technology for visually-impaired individuals has become a growing space of research within Assistive Technology. A team at the MIT Energy Efficient Circuits Group has made an important stride forward by presenting a wearable navigation prototype in a fully integrated hardware form factor, but one of biggest barriers to usability of the device is its excessive power consumption. As such, the goal of this work is, broadly, to– (1) Understand the largest sources of power consumption in the initial navigation prototype system, and expose relevant features for control; (2) Develop a set of algorithms that can capitalize on the motion of a user, the motion of the environment around a user, and the proximity of obstacles within the environment to the user, in order to dynamically tune the exposed parameters to scale power as necessary; and (3) Lay the foundation for the next generation wearable navigation prototype by translating critical software operations and the power scaling algorithms into a hardware architecture capable of working with a smaller and less power intensive depth camera.

The first portion of this work focuses on the wearable navigation prototype built around Texas Instrument's OPT9220/9221 Time of Flight chipset. Illumination voltage, frame rate, and integration duty cycle are identified as key control features, and a step rate estimation algorithm, scene statistics algorithm, and frame skipping controller to tune these features are built and tested. The latter half the work focuses on the newer OPT8320 evaluation platform, for which a Bluespec System Verilog implementation of these power algorithms and the point cloud generation operation is presented and tested.

Overall, the work demonstrates the critical concept that simple, system specific, fully integrated algorithms can effectively be used to reduce analog power system-wide.

Thesis Supervisor: Anantha P. Chandrakasan
Title: Joseph F. and Nancy P. Keithley Professor of Electrical Engineering

Acknowledgments

First, last, foremost, and always, my gratitude to the Almighty for guiding me through this journey. Without You, I am nothing; and I humbly offer every small success, every fruitful failure, and every lesson learned to You.

I must begin with my endless gratitude to Professor Chandrakasan. It has been an honor and a wonderful experience to have had you as a mentor not only over the course of this year, but from the moment I set foot into this department as an extremely shy and fearful sophomore. You encouraged me to voice my opinion during USAGE meetings, gave me the opportunity to thrive as a SuperUROP student by nominating me to speak at various events, welcomed me into your group when I expressed an interest in working on the navigation project, and challenged me to leap into the field of digital design without reservation. As I look back upon my five years within the walls of MIT, I realize that there is no faculty member that I admire more for their ability to be a world-renowned leader in their field, but more importantly, for their patience, support, and genuine desire for the success of each and every student. Thank you, Professor.

Next, the latter half of this thesis certainly would not be complete without the mention of some incredible individuals. Andre, a million thanks for being my partner in crime for this year's 6.375 final project. Your diligence, patience in working with my research goals, sharp Bluespec skills and even sharper diagram-drawing skills have made the process really enjoyable. A very special thanks to our mentors, Jamey Hicks and John Anckorn, for taking the time to answer any question at any time of day– Andre and I were extremely fortunate to have you both to guide us along. A huge thanks also to Professor Arvind and class TA Ming for their expertise and for keeping us on a strict timeline!

I would be remiss in my acknowledgements if I forgot to thank all of the talented students in Ananthagroup for their help in bringing a newcomer up to speed. While I have probably asked each one of you a number of silly questions, I'd like to specifically thank Nathan, Donsuk, Priyanka, Mehul, Arun, and Frank for the time and

encouragement they have given me.

I have always thought of my life to be a complicated sheet of music. Notes and rhythms forming beautiful but chaotic patterns, dynamics that rise and fall with the times, chromatic scales and accidentals that make their presence known the most when they are expected the least. There is one thing, however, that remains unwavering– the five staff lines that stand as steadfast guides for the music above.

Mummy, you are the source of love and peace in our household, my source of strength and balance. I hope everyday that I can walk with the morals you have given me and talk as sweetly as you have shown me. Daddy, you are my inspiration in so many ways. Despite my *occasional* incompetance in practical matters, I hope that someday you will look back upon the lives of the girls that you have raised and disciplined, and that your heart will swell with pride. Bhavani, you are my better half. I regret that I haven't always been able to be there for you as you trekked through your own collegiate adventure, but you are my biggest source of happiness– the laughter we share is my biggest gift. Mamma, your stories are the lens through which I view the world and the music you have taught me to love is my most trusted companion. I always look to you for words of reassurance and faith, while counting down the minutes until we can be united again. And Thatha. When I am tired and have no more patience to work, I remember the words you utter whenever I leave home– *"Kshemamga Velli Labhamga Raa"* (go safely, and come back successfully). It is the drive to live up to your words, to bring back little bits of success each time I see you again, that keeps me going.

Thank you.

Contents

1 Introduction **17**

1.1 Assistive Technology for the Visually Impaired 17

1.2 Current Approach and Challenges 18

2 Implementation for the Wearable Navigation Platform, Version 1.0 **21**

2.1 Background 21

2.1.1 Introduction to the Wearable Navigation Platform, Version 1.0 21

2.1.2 System Power Model 24

2.1.3 Power Saving Approach 26

2.2 Algorithm Theory Description 29

2.2.1 IMU Pose Estimation 29

2.2.2 Scene Statistics 34

2.2.3 Frame Skipping and Feedback Control 39

2.3 Implementation Details 41

2.3.1 Power Utility for a General ARM Processor 41

2.3.2 Power Measurement Board Design 42

2.4 Results 44

2.4.1 Power Measurement Data 44

2.4.2 Power Comparisons and Observations 45

3 Implementation for Calculus Platform (TI OPT8320) **49**

3.1 Background 49

3.1.1 Motivation for Next Generation Navigation Device 49

3.1.2 Relevant Platform and System Specifications 50
3.1.3 Power Consumption and Power Model of the System 50
3.2 Translating Power Optimization Algorithms into Hardware 51
3.2.1 Point Cloud Generation: Theory and Architecture 54
3.2.2 IMU Pose Estimation: Theory and Architecture 56
3.2.3 Scene Differencing: Theory and Architecture 57
3.2.4 Scene Statistics: Theory and Architecture 59
3.2.5 Camera Controller 61
3.2.6 Software Simulation 62
3.2.7 Potential Control Challenges 62
3.3 Results 63
3.3.1 Timing Analysis, Testing, and Visualization 63

4 Conclusion **69**

4.1 Impact 69
4.2 Summary of Contributions 69
4.3 Future Work 70

List of Figures

2-1 A system overview of the first wearable navigation platform displaying each of the major components [5]. 22

2-2 Bottom: A detailed depiction of the navigation system data flow. Top: A visualization of each of the stages in the system dataflow. [5] . . . 23

2-3 This diagram illustrates the contribution of the illumination panel to the system power curve. For the configuration of the chipset used on the navigation platform, each frame consists of 16 integration periods, and the ratio between the total integration time (sum of all of the pulse widths) and the total frame duration is known as the integration duty cycle, or $D_{Integration}$. 24

2-4 Power consumption of the system as a function of frame rate. 26

2-5 Power consumption of the system as a function of illumination base voltage. 27

2-6 This image illustrates the high level approach taken towards the dynamic power scaling task. The scene differencing and scene statistics concepts make use of point cloud or phase data, and the step rate concept makes use of IMU data. The step rate estimation and scene statistics concepts together inform the illumination parameter, while the scene differencing concept informs the frame rate. 28

2-7 Sample IMU data of an individual jogging (Top), walking at a normal pace (Middle), walking and then stopping to sit (Bottom). 31

2-8 Raw data and FFT results for each window of sample IMU data run through the Hanning Windowed FFT approach. Identified peaks are marked with X's. 32
2-9 Temporal sample IMU data run through the CWT peak finding algorithm. Identified peaks are marked with X's. 32
2-10 Raw Window sample followed by autocorrelated result. Amplitude of first non-centered maxima provides frequency estimate, which can be found by a CWT peak finder or naively. 33
2-11 Confidence histogram for static object at a 1 meter distance. Notice that the confidence increases only slightly with higher illumination voltages. 35
2-12 Confidence histogram for static object at a 1.5 meter distance. Notice that the confidence increases noticeably with higher illumination voltages. 35
2-13 Confidence histogram for static object at a 2 meter distance. Notice that the confidence increases dramatically with higher illumination voltages. 36
2-14 Average number of dropped pixed for a frame at a distance of 1M, for different illumination voltages and integration percentages 37
2-15 Average number of dropped pixed for a frame at a distance of 1.5M, for different illumination voltages and integration percentages 37
2-16 Average number of dropped pixed for a frame at a distance of 2M, for different illumination voltages and integration percentages 38
2-17 Minimum illumination and integration strictly required plotted at each distance. 38
2-18 Illustration of the frame skipping algorithm as implemented in the Vision Processor ASIC [5]. 40
2-19 Schematic of the Power Monitor used for experimentation with the power algorithms applied to the navigation board. 42
2-20 Layout of the Power Monitor used for experimentation with the power algorithms applied to the navigation board. 43

2-21 Photograph of the Power Monitor used for experimentation with the power algorithms applied to the navigation board. 43

2-22 Holodeck Experiment Layout. The correlated walls formed a maze which allowed for a diversity in step rate and environmental proximity between the user and the camera. 44

2-23 Sample power consumption plotted by power monitor at 7.5V illumination, 30fps. 45

2-24 Power consumption with algorithms applied and a user walking at a slower pace. The blue line shows the default power consumption, the green line shows the application of the skip frame algorithm only, the red line shows the application of the scene analysis algorithm only, and the light blue line shows both algorithms combined. 46

2-25 Power consumption with algorithms applied and a user walking at a faster pace. The blue line shows the default power consumption, the green line shows the application of the skip frame algorithm only, the red line shows the application of the scene analysis algorithm only, and the light blue line shows both algorithms combined. 46

3-1 OPT8320 main board power consumption as a function of frame rate, at illumination panel current of 30mA. The blue plot shows a fixed integration duty cycle of 20 percent, where the orange plot displays a fixed ratio equal to the integration duty cycle over frame rate, set here at 0.01. 51

3-2 OPT8320 main board power consumption as a function of illumination, at frame rate of 30 fps and 20 percent integration. 52

3-3 System overview of the hardware implementation for the dynamic power scaling algorithms. Courtesy of Andre Aboulian. 53

3-4 This figure shows the pinhole camera model that was used in the point-cloud transformation for this study. We find point P, the point in the real world, from point Q, the point on the image plane, and the depth ray that is read by the imager. 54
3-5 A microarchitecture diagram of the pointcloud generation module. Courtesy of Andre Aboulian. 57
3-6 High level diagram of the superfolded circular FFT module used to compute the step rate in the IMU module. 58
3-7 A microarchitecture diagram of the step rate module. Courtesy of Andre Aboulian. 58
3-8 Microarchitecture of the scene differencing module. Courtesy of Andre Aboulian. 59
3-9 Microarchitecture of the scene statistics module. Courtesy of Andre Aboulian. 61
3-10 Test Case A: The frame rate compensation is plotted with the density of skip flags being produced by the scene differencing module. The camera is alternated between being held still and then shaken rapidly at 5 second intervals for a total of approximately 25 seconds. 65
3-11 Test Case B: The frame rate compensation is plotted with the density of skip flags being produced by the scene differencing module. The camera is again alternated between being held still and then shaken rapidly at 5 second intervals for a total of approximately 25 seconds, but the gain of the control loop is now perturbed by the step rate, resulting in sharper increases and decreases. 66
3-12 Test Case C: The camera is slowly drawn further and further away from a wall at 6 different stages of distance in 5 second intervals. As can be seen, the illumination scales appropriately by selecting one of the three stages of illumination – 0, 1, or 2. The noise at the transition points likely results from the unpredictable motion of the camera as it was drawn away, or the presence of an artifact such as a hand, etc. . . 67

3-13 Test Case D: Visualization of the pointcloud generated by the pointcloud module. The top image is the pointcloud rendered by TI's VoxelViewer software from the phase data. The bottom image is the pointcloud produced by the module in this system from the same phase data. 68

List of Tables

3.1 Table that shows the frequency at which each module outputs a result, constrained by the rate of delivery of new frames from each module's respective sensor shown in the column on the right. 63

3.2 Table that lists the four test cases for which strategic input data was generated. The resulting data is shown in later figures. 64

Chapter 1

Introduction

1.1 Assistive Technology for the Visually Impaired

With modern advances in computer vision, planning algorithms, and feedback systems, wearable navigation devices have begun to occupy a crucial niche in the space of Assistive Technologies. In particular, it has been well-established by the visually-impaired community that such systems can be pivotal in providing users independence, confidence, and safety in both familiar and unfamiliar environments. Within the last decade, dozens of approaches have been taken by the vision and engineering communities to tackle the development of such local navigation devices, focusing on several key components that inherently demand trade-offs– identifying the most relevant features for environment detection, obtaining resolution and accuracy in this detection, providing a feedback mechanism that is an adequate modal exchange, and minimizing the size and weight of such a device for the sake of comfort, amongst other features. Landmark surveys in this space have outlined a simple, four part success metric for the design of wearable navigation technology– the system should be hands-free, ears-free, wearable, and simple in the context of learning, use, and maintenance [2].

In an attempt to satisfy these metrics, more software focused systems have incorporated high-density stereovision imaging[8], sonar range-finding coupled with IMU orientation estimation, and infrared transceiver based localization[3]. Other systems

have employed image-to-audio mapping, 3D laser finders or range sensors, realtime processing on low-cost digital or web camera data, global positioning data for localization, and a variety of other prototypes that consist of some combination of the techniques mentioned above [2].

While much of this technology has demonstrated experimental success, none of the devices have reached a stage of development that suggest readiness for use by a visually-impaired individual, arguably because they do not meet all of the components of the framework or do not meet them sufficiently. For example, several of the highest scoring systems presented in [2] include computers that need to be carried for realtime processing, body appendages with bulky hardware, require being held or moved manually for object detection, or have chosen auditory communication as the preferred method of feedback.

1.2 Current Approach and Challenges

In the Energy Efficient Circuits Group of the Microsystems Technology Laboratory at MIT, a team of researchers has proposed an alternate solution. The team's wearable navigation project is a design that attempts to address the points of the framework with an entirely hardware-based approach that is robust and compact [4]. The system architecture includes a ToF camera and associated peripherals, a novel ASIC dedicated to realtime processing of depth frame data from the ToF processor and computing a persistent point-cloud[5], a Zynq SOC with an FPGA and ARM Core, and other generic peripherals including wifi/bluetooth modules, pulsed illumination panels, etc. The first prototype has resulted in a encased PCB that is no more than an estimated 6 cm x 10 cm and 5 cm in depth, and is mounted to one's chest. The system relies on bluetooth transmission-enabled haptic feedback that is sent to a small belt, in line with a well-established feedback protocol in this research area [7][2].

While the system demonstrates a novel approach and promising initial results, there is certainly one large barrier to "wearability"– power consumption. Given that the frequent recharging/ replacement of batteries or the use of heavy, voluminous

batteries are both obvious impediments to the wearability of a system, there is an important need for intelligent power management of such a device that is informed by the user's location, patterns of movement, and surrounding environment.

As such, this work focuses primarily on improvements on this system from the perspective of power expenditure. The first portion of this work entails understanding the major sources of power expenditure that have been built into the system, exposing the relevant parameters that can be tuned to scale power, and developing a set of algorithms that have been tailored specifically to this application. The latter half of this work seeks to consider an improved second generation prototype that characterizes the performance of a different ToF camera platform, and translates both a computationally hefty operation and the power algorithms formalized in the first part of this work to a hardware FPGA implementation. Ultimately, the goal of this thesis is to demonstrate that a "power intelligent" implementation for vision and mobility based systems is capable of meeting the standards expected for a wearable device from a power consumption standpoint.

Chapter 2

Implementation for the Wearable Navigation Platform, Version 1.0

2.1 Background

2.1.1 Introduction to the Wearable Navigation Platform, Version 1.0

The first wearable navigation platform, designed by former students in the Energy Efficient Circuits group, is a device that addresses some of the constraints and questions raised earlier in an integrated hardware form-factor. As mentioned, the device measures roughly 6cm x 10cm in area and 5cm in depth, is designed to be worn as pendant around the neck. The high-level functionality of the device can be described as follows: as a user moves about his or her environment with the pendant on, an on-board camera segments the user's field of view into discrete segments and identifies the distance to the nearest obstacle in each of these directions. This information is then conveyed back to the human user as haptic or tactile feedback.

A system overview can be seen in the diagram 2-1 below.

As shown, the device consists of a Time-of-Flight (ToF) camera, a Zynq SOC consisting of an FPGA and ARM processor, and Inertial Measurement Unit (IMU), a Bluetooth module, and a custom ASIC developed for performing a piece of the

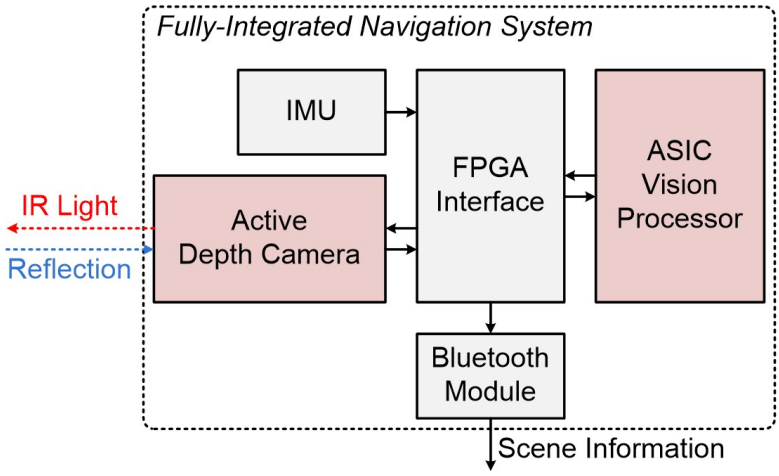

Figure 2-1: A system overview of the first wearable navigation platform displaying each of the major components [5].

energy efficient computation relevant to this application. In order to present the features and algorithms considered for the power scaling of a system comprised of these components, and before detailing the dataflow of the system, it is useful to elaborate briefly on the theory of operation of the ToF Camera and the associated processing required to handle this data in later stages of the application flow.

The ToF camera used in this system is Texas Instrument's OPT9220/OPT9221 chipset, with a frame dimension of 320x240 (QVGA) and a maximum range of approximately 2.5 meters. The ToF camera operates by actively modulating infrared illumination at a known frequency and capturing the light that is returned to a sensor array after it has been reflected from various obstacles in the environment. The sensor array calculates the phase difference between the reflected light and the pulsed light, and uses this quantity in conjunction with the known range of the camera (which is a function of the modulation frequency) to extrapolate the distance to the object. This process is undertaken for each pixel in the sensor array, so each frame returned by the ToF camera is a 320 x 240 depth map of the scene in its field of view.

Investigating this process further, we find that the process of collecting the reflected light and generating frames occurs in a sequence of four steps: Reset, Integration, Readout, and Dead Time. 'Reset' clears the sensor array of accumulated light; 'Integration' pulses the illumination panel and accumulates the IR light over a

designated period of time (see 2.1.2 for more details); 'Readout' hands off the signal to an ADC for processing and piping through the rest of the workflow; 'Dead Time' is the number of additional empty clock cycles spent before the cycle of the four steps begins again. In this system, each frame delivered by the camera consists of four divisions known as subframes, each of which consist of an additional four divisions known as quads, each of which finally contain the four steps mentioned above. These variables are relevant in the control of a few power-centric parameters, to be detailed in the following section.

Returning to the prototype system itself, a detailed understanding of the data flow of the device is presented in 2-2.

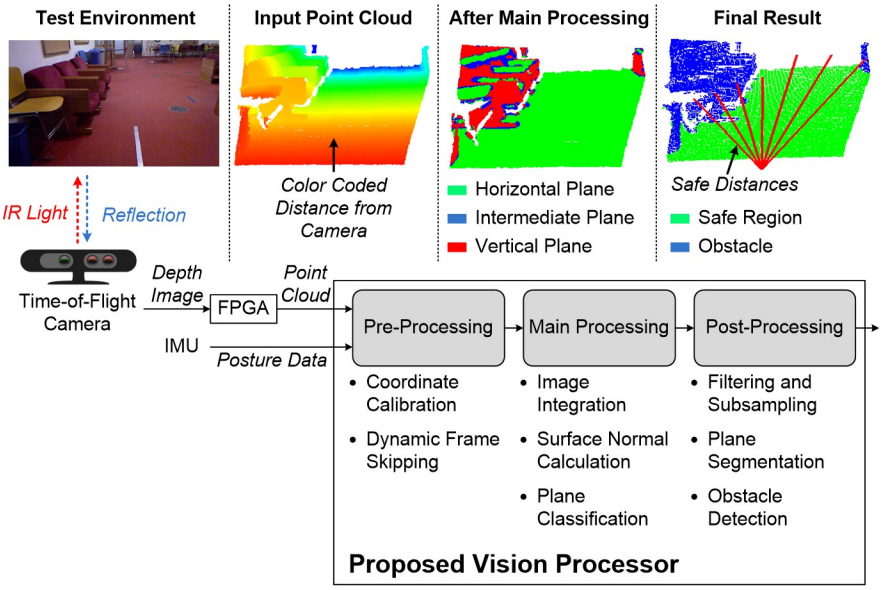

Figure 2-2: Bottom: A detailed depiction of the navigation system data flow. Top: A visualization of each of the stages in the system dataflow. [5]

The ToF camera controlled by the FPGA passes along a depth frame to the SOC, where a 3D pointcloud is computed in software on the ARM processor. The pointcloud is then handed to the ASIC vision processor, which assigns surface normal vectors to the points in the pointcloud, characterizes the scene into planes using a region-growing algorithm, and finally computes the distance between the camera and the planes which oppose it in 7 discrete directions, known as the "safe distances". These safe-distances are finally passed along to the bluetooth module for transmission to

the haptic feedback instruments. Additionally, the IMU data is used as input into the system to capture information about a user's pattern of movement, though this functionality was implemented later, by the author of this work.

2.1.2 System Power Model

We begin by providing a simplified power model of the navigation platform. The model is an aggregate representation of the average power consumed by each of the major components in the system– the illumination LED panel, the ASIC, the FPGA/ ARM processor, the IMU, and the Bluetooth transmission chip. First, it is critical to understand the behavior of the illumination panel in terms of power, which gives rise to the model presented later in the section. Figure 2-3 shows a diagram illustrating how the system power oscillates as dictated by the pulsing of the illumination panel during integration periods.

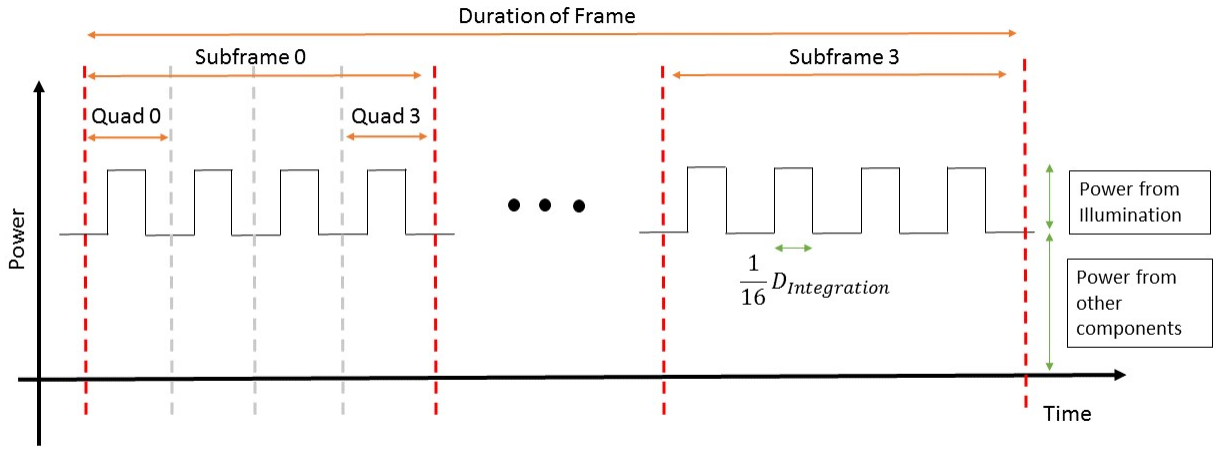

Figure 2-3: This diagram illustrates the contribution of the illumination panel to the system power curve. For the configuration of the chipset used on the navigation platform, each frame consists of 16 integration periods, and the ratio between the total integration time (sum of all of the pulse widths) and the total frame duration is known as the integration duty cycle, or $D_{Integration}$.

As described in the introduction, each frame is divided into subframes and further divided into quads, each of which has an integration period for some duration of the quad. The configuration used for the chipset on the navigation platform has 4

subframes and 4 quads for each frame, and hence 16 integration periods when the illumination panel is actively modulated at a preset frequency to collect depth data. The ratio of the total integration time (the sum of all 16 pulse widths) to the total frame duration is known as the integration duty cycle, which will be referred to as $D_Integration$.

Given this, the equation representing the model for this system is given below:

$$P_{total} = (D_{Integration} \cdot \Delta V \cdot I_{LED}) + P_{ToFProcessor} + P_{ASIC} + P_{IMU} + P_{Bluetooth} + P_{ARM} + P_{FPGA} \tag{2.1}$$

where ΔV is the voltage required to power the illumination panel when the LEDs are turned 'ON' fully, I_{LED} is the current draw of the LED panel, $D_{Integration}$ is the per-frame integration duty cycle, and the P_x terms represent the power draw of the components designated by the term subscript. If instead of a constant integration duty cycle, it is scaled to maintain a constant ratio with the changing frame rate, we have:

$$P_{total} = (N \cdot FrameRate \cdot \Delta V \cdot I_{LED}) + P_{ToFProcessor} + P_{ASIC} + P_{IMU} + P_{Bluetooth} + P_{ARM} + P_{FPGA} \tag{2.2}$$

where N is the ratio between $D_{Integration}$ and $FrameRate$. This approach is commonly used to ensure that there is a sufficient amount of integration time for high frame rates. It should be noted that the P_x terms are not necessarily constant factors, as the other components in the system apart from the illumination panel might also be a function of the frame rate. For example, the ASIC, Bluetooth transmission chip, and the ToF processor can be duty cycled with the frame rate, and power demanded by the ARM processor will also likely vary as the rate of input information changes. However, in this simple model, we consider the illumination panel to be the most impactful and controllable source of power consumption. Based on this model, the voltage required to power the illumination panel, the integration duty cycle, and the frame rate of the ToF camera are chosen as the control parameters for task of

dynamically scaling power. These features were exposed for external control through a custom software interface written for the navigation platform software.

To demonstrate how the system power scales with the parameters of interest, quantitative average power measurements are provided. The average power for the system as a function of frame rate can be seen in 2-4 below, with both fixed integration duration values, and integration time values scaled with the frame rate. Note that the range of the power consumption for the scaled integration plot is much greater than the fixed integration plot, which results from the equations above.

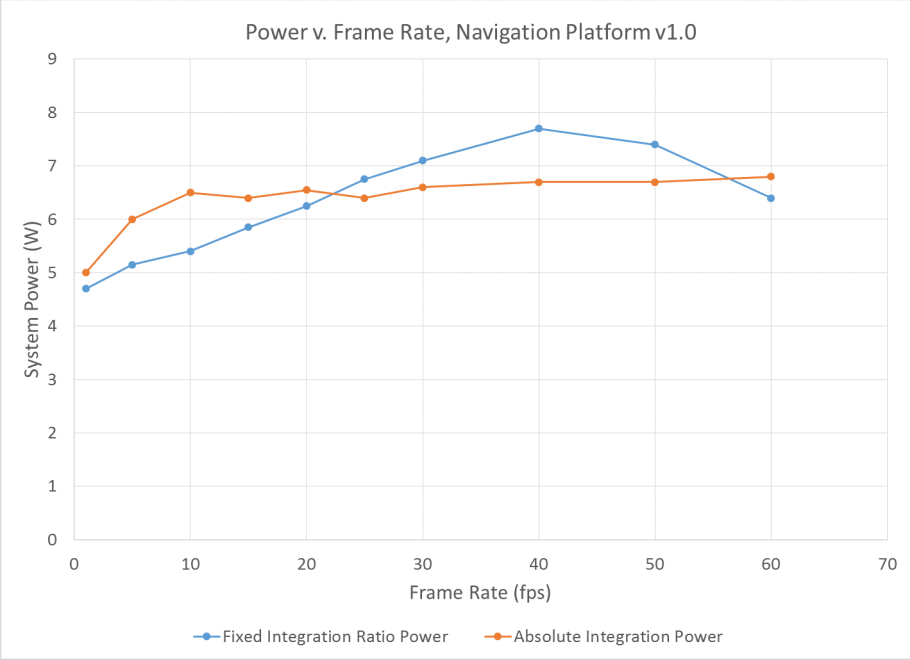

Figure 2-4: Power consumption of the system as a function of frame rate.

Additionally, the system power as a function of illumination is also given. The average power across illumination voltage increments is given by the graph shown in 2-5.

2.1.3 Power Saving Approach

The initial portion of this thesis is aimed at capitalizing on the three controllable features that have been extracted from the power model, and tuning them dynamically as the wearable navigation system is in use. An illustration of the approach taken

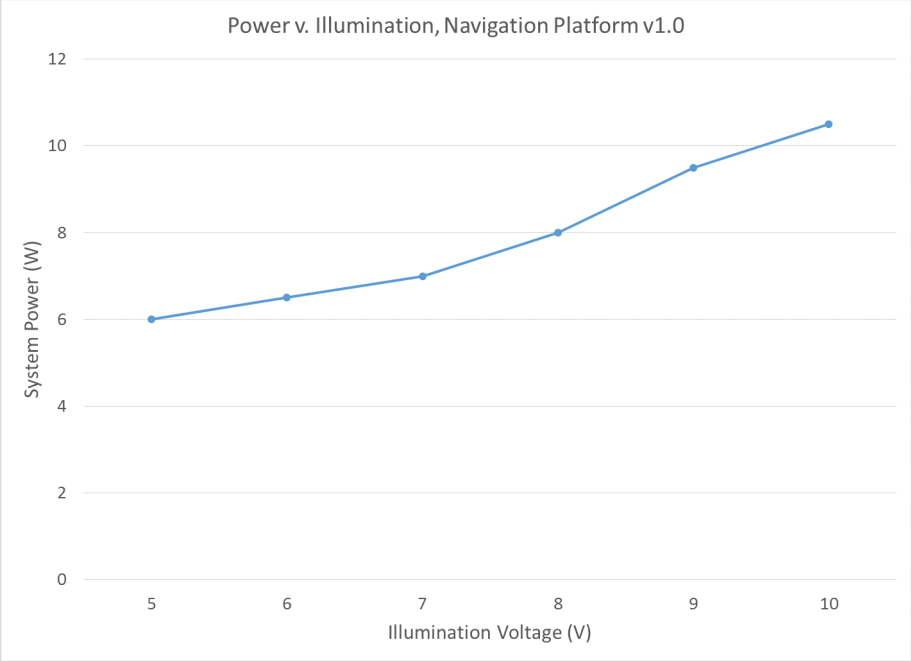

Figure 2-5: Power consumption of the system as a function of illumination base voltage.

towards this task is shown in 2-6.

In order to automatically scale the system power as needed, the system must be able to characterize how quickly a user wearing it is moving, how quickly the environment around the user is varying (if at all), and how the proximity of objects in the environment to the user is changing. As shown in the diagram, the first point is addressed by processing data from the onboard IMU to characterize the user's step rate. The second point is addressed by a scene differencing control algorithm, which capitalizes on the "skip frame" infrastructure already present as a part of the ASIC [5]. The third point is addressed by a scene statistics algorithm, which uses a histogramming approach to quantify environmental proximity. In this first implementation, the step rate algorithm and the scene statistics algorithm are combined to inform the illumination and integration parameters, while the scene differencing algorithm is used to inform the frame rate parameter. The algorithms are presented in more detail in the following sections.

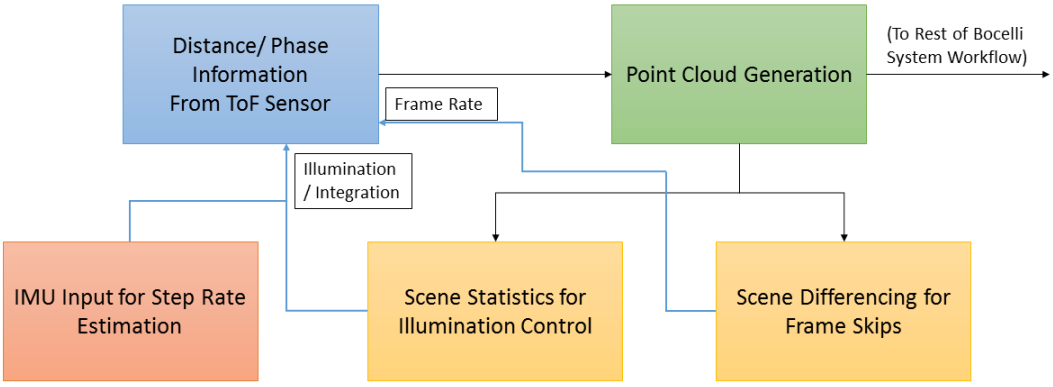

Figure 2-6: This image illustrates the high level approach taken towards the dynamic power scaling task. The scene differencing and scene statistics concepts make use of point cloud or phase data, and the step rate concept makes use of IMU data. The step rate estimation and scene statistics concepts together inform the illumination parameter, while the scene differencing concept informs the frame rate.

2.2 Algorithm Theory Description

The following subsections will detail the algorithms for dynamic power scaling that have been implemented on the initial prototype system. The schemes for power scaling can be classified into three components– inertial measurements for human pose estimation, per-frame statistics for illumination analysis, and frame skipping feedback control.

2.2.1 IMU Pose Estimation

One goal of the power reduction intelligence was to capitalize on a human's natural motion to scale the required power on a wearable device. It was interesting to note that, while a visually-impaired user might choose to keep his or her guidance device on for the majority of the day, a good portion of the work day might be spent with minimal motion– such as being seated at a desk or standing steady during a conversation. Additionally, it was noted that high camera frame rates are valuable in three cases– either when the user is moving very rapidly, or when the scene around the user is changing very rapidly (such as if the user is at the intersection of a very busy street), or if both of the aforementioned cases are true.

While the second case will be addressed in later section, this work proposes to target the first case with the use of an in-built inertial measurement unit (IMU) to characterize the rate of motion of the user. Substantial signal processing research has been done on active pose estimation from IMU data, where pose estimation refers to the task of identifying the stance or state of a human based on sensor data collected over time. The methodologies suggested by the relevant literature falls into the following categories– feature extraction and machine learning or classification on a priori data, absolute position and motion identification by integration, or extrapolation based on zero-velocity foot placements. While the first two approaches are certainly more precise, this application is primarily interested in robust, responsive step rate detection, and only from a chest mounted IMU sensor [1]. A simple variation on the last approach was tested and implemented in this work.

The IMU embedded on the first navigation platform is an MPU9250 IC. It is a 9-DOF device with built in accelerometer, magnetometer, and gyroscope capable of 100 Hz refresh rate. Sample output from the IMU in various user scenarios is shown in 2-7.

Initially, various algorithms were prototyped in software using a small evaluation board containing the same IMU IC, and a Microsoft Kinect device, to mimic the behavior of the ToF Camera as the frame rate was changed dynamically.

The first was a Hanning-Windowed FFT approach. The Hanning Window, to reduce the presence of exaggerated features at frame edges, is represented by the following equation:

Discrete Hann Window:

$$w(n) = \sin^2\left(\frac{\pi n}{N-1}\right) \tag{2.3}$$

Raw accelerometer data on the axis that corresponds to the vertical length of the body was streamed in at a sample rate of 20Hz, and processed in overlapping window sizes of 50 percent. A sample result of this algorithm can be demonstrated by 2-8.

The second algorithm applied is a Continuous Wavelet Transformation (CWT) for temporal peak finding. This involved post-processing of the accelerometer peaks within a frame to compute a representative step rate for the window, by accounting for the number of peaks in a frame. The algorithm can be summarized as follows:

1. Perform a D/CWT on the vector of data that represents that particular window, for the supplied value of widths. This is defined as a convolution of the data with a 'wavelet' of a particular width for each width that is supplied.

2. Identify the ridge lines in the cwt matrix, which are the relative maxima.

3. Recursively filter the ridge lines.

A sample result of the application of this algorithm can be seen in 2-9.

The third algorithm applied is a standard autocorrelation, where the frame data

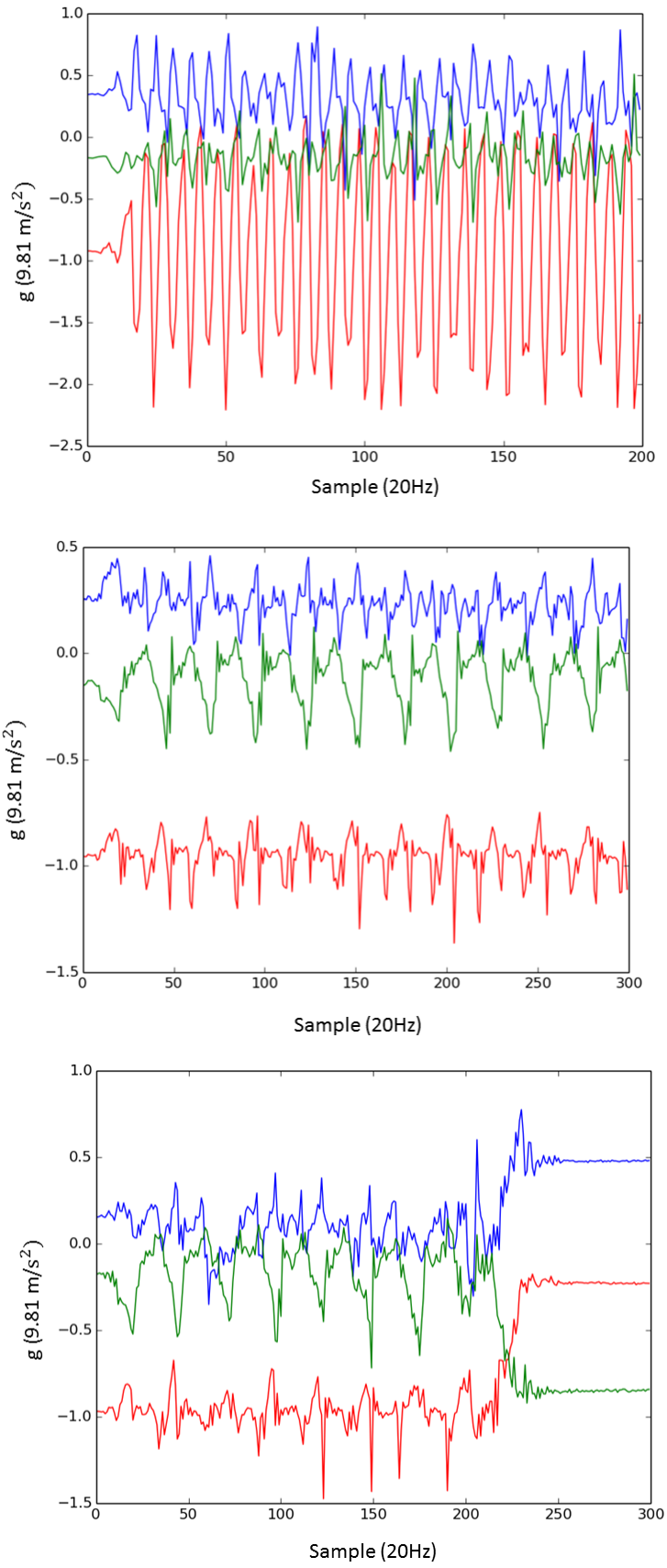

Figure 2-7: Sample IMU data of an individual jogging (Top), walking at a normal pace (Middle), walking and then stopping to sit (Bottom).

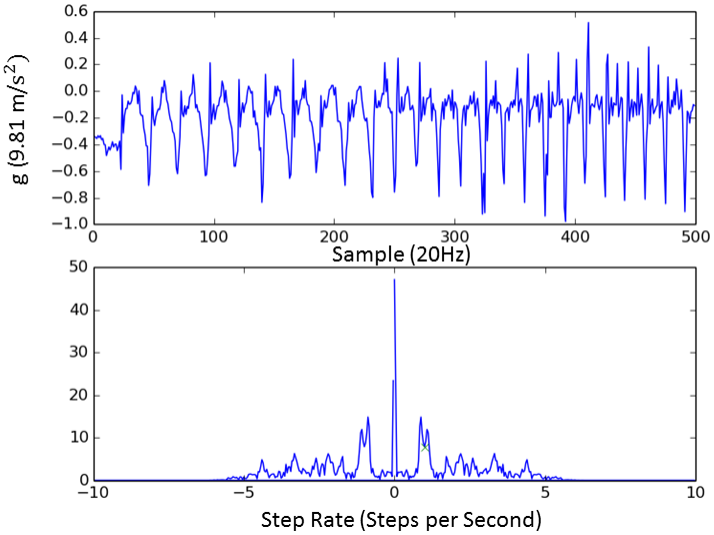

Figure 2-8: Raw data and FFT results for each window of sample IMU data run through the Hanning Windowed FFT approach. Identified peaks are marked with X's.

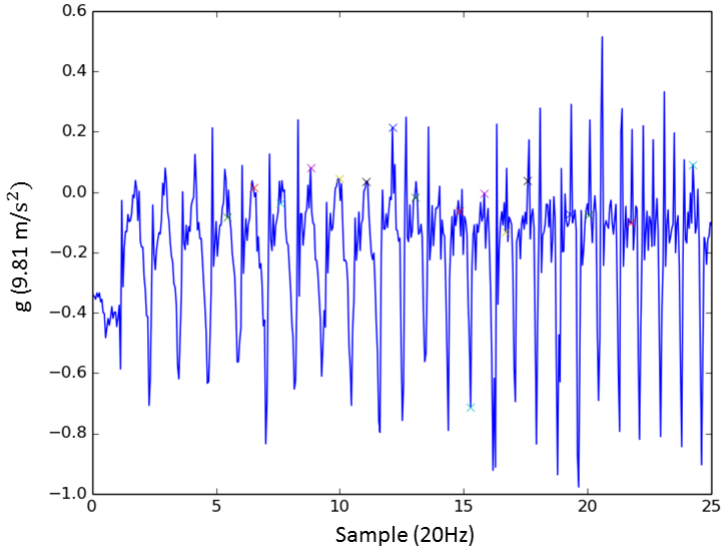

Figure 2-9: Temporal sample IMU data run through the CWT peak finding algorithm. Identified peaks are marked with X's.

is convolved with itself to identify peaks. The discrete autocorrelation function is as given below:

$$R_{yy}l = \sum_{n \in Z} y(n)\,\overline{y}(n-1) \tag{2.4}$$

A result from the application of the algorithm is shown in 2-10.

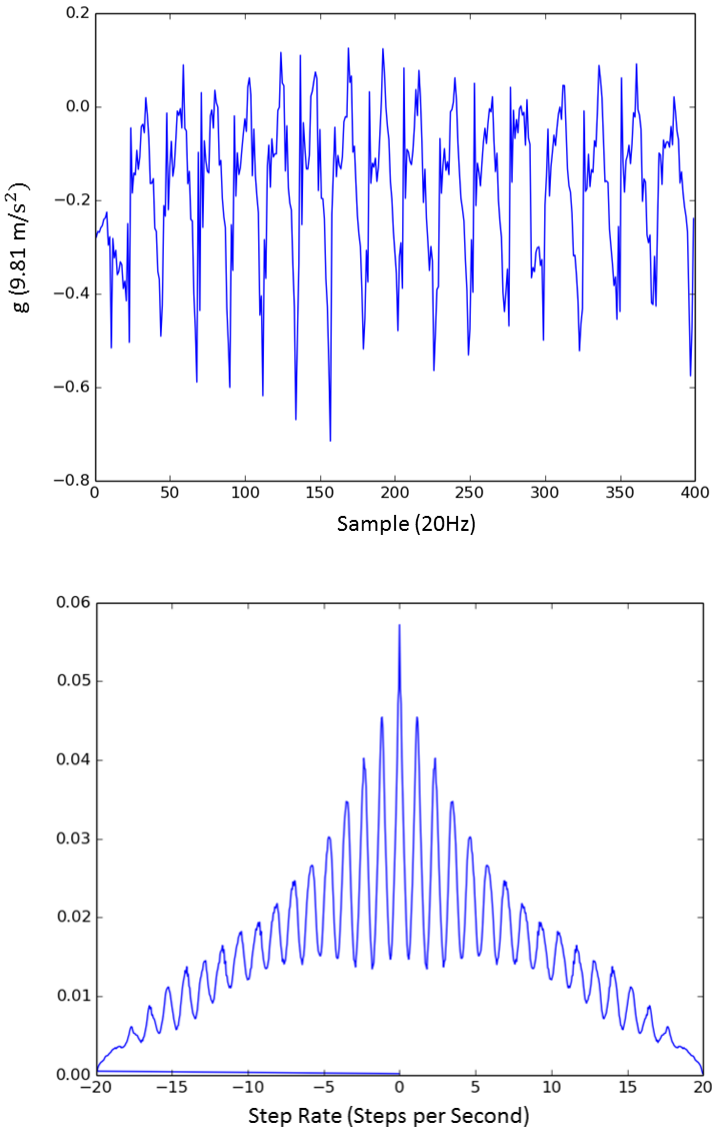

Figure 2-10: Raw Window sample followed by autocorrelated result. Amplitude of first non-centered maxima provides frequency estimate, which can be found by a CWT peak finder or naively.

After all algorithms were implemented in the simulated setup, the algorithms were evaluated for minimal computational requirements and ease/ robustness of implementation, first on a general purpose ARM processor, and eventually an FPGA. The first algorithm, a Hanning-Windowed FFT, was selected for implementation on the navigation system.

2.2.2 Scene Statistics

In this work, it is also suggested that it may be possible to garner information about the way in which system power should dynamically be scaled from the image data itself. This was approached by characterizing the general proximity of objects in a frame to the user wearing the device, and using this data to inform the amount of illumination power that is minimally sufficient for the point cloud computation to be accurate enough for the surface normal characterization and safe-distance identification phases.

While there are many particular cases that can be taken into consideration in designing a scene-analysis algorithm for illumination scaling, an intentional metric– that the largest, nearest objects would determine the minimal amount of illumination power sufficient to capture the scene's phase data– was imposed to guide the prototype.

We begin by characterizing the behavior of the system in terms of the returned amplitude (intensity of light, also referred to here as confidence [6]) for static fields-of-view at varying distances. A calibration experiment to verify the need for dynamic scaling of illumination was first performed, wherein large, flat walls occupying the entire field-of-view of the camera were placed at specified distances away from the lens, with the furthest distance being the maximum range of the system. At each stage, the per-pixel confidence of the 320 x 240 image was histogrammed into 10 bins. Select results are demonstrated by the figures 2-11, 2-12, and 2-13 below. Note that only the first six bins of the histogram are displayed in the figures below, as these are the regions of interest.

The data collected from the initial set of experiments implies that, while increasing

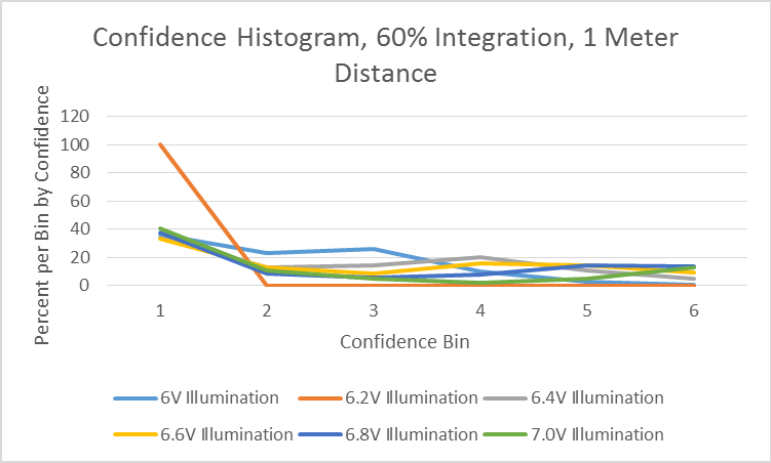

Figure 2-11: Confidence histogram for static object at a 1 meter distance. Notice that the confidence increases only slightly with higher illumination voltages.

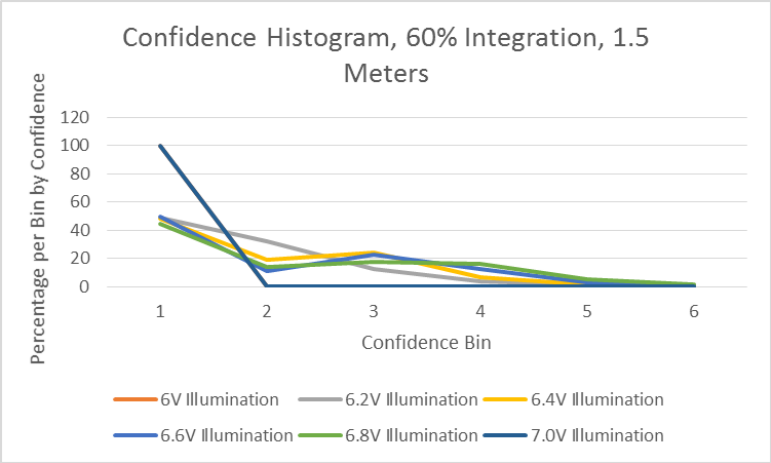

Figure 2-12: Confidence histogram for static object at a 1.5 meter distance. Notice that the confidence increases noticeably with higher illumination voltages.

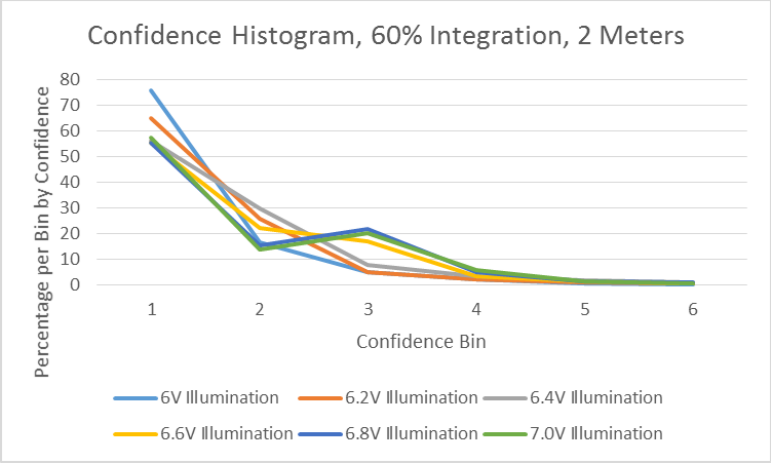

Figure 2-13: Confidence histogram for static object at a 2 meter distance. Notice that the confidence increases dramatically with higher illumination voltages.

the illumination makes little difference in the amplitude of returned light at a close proximity to an object, there is a noticeable decrease in confidence by up to 40 percent (by bin) for objects further away. It was also determined that, for closer objects, excessively high illumination voltages cause over-saturation in nearby pixels and reduce the confidence more than lower illumination values.

Furthermore, a set of experiments was performed in order to determine the minimally required illumination voltage and integration percentage using a rigid confidence metric. The previous system designers had imposed an experimentally determined "clean-up" metric that discarded any phase pixels beneath a threshold. To determine how this quantity of "dropped pixels" varied with illumination, integration, and distance parameters, the procedure above was repeated while this measure was collected over several hundred frames. Some results are summarized in the figures 2-14, 2-15, 2-16 below.

If this rigid confidence threshold is imposed with the number of "dropped pixels" as the quantity over which to optimize, the optimum illumination and integration parameters for a scene that is strictly at a certain distance can be extracted from the dataset partially pictured above. The results are shown in figure 2-17 below:

It is evident, of course, that most scenes captured by the ToF as the wearable system is in operation would not have all of the scene's pixels homogenously attributed

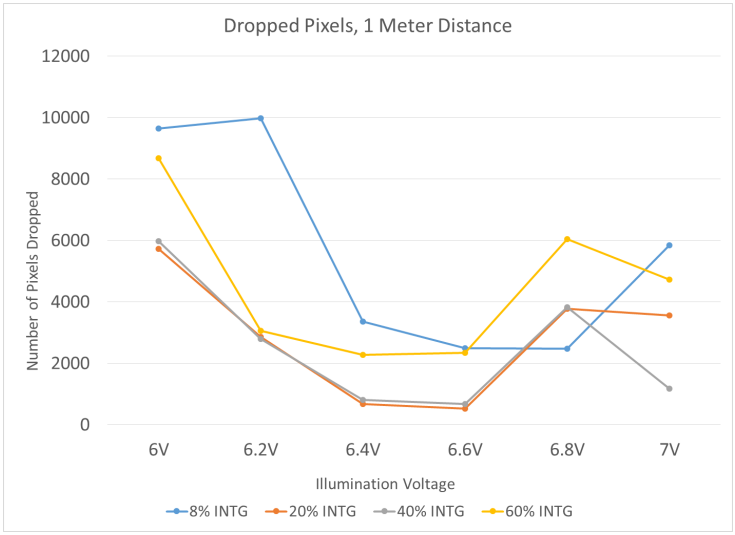

Figure 2-14: Average number of dropped pixed for a frame at a distance of 1M, for different illumination voltages and integration percentages

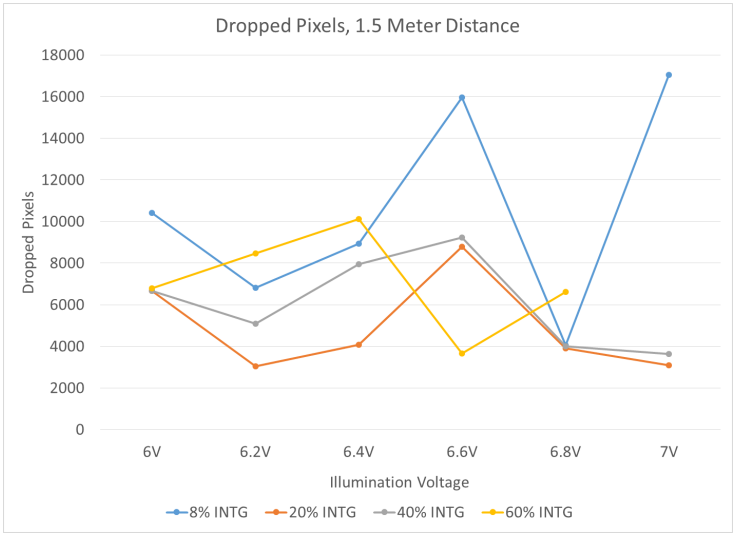

Figure 2-15: Average number of dropped pixed for a frame at a distance of 1.5M, for different illumination voltages and integration percentages

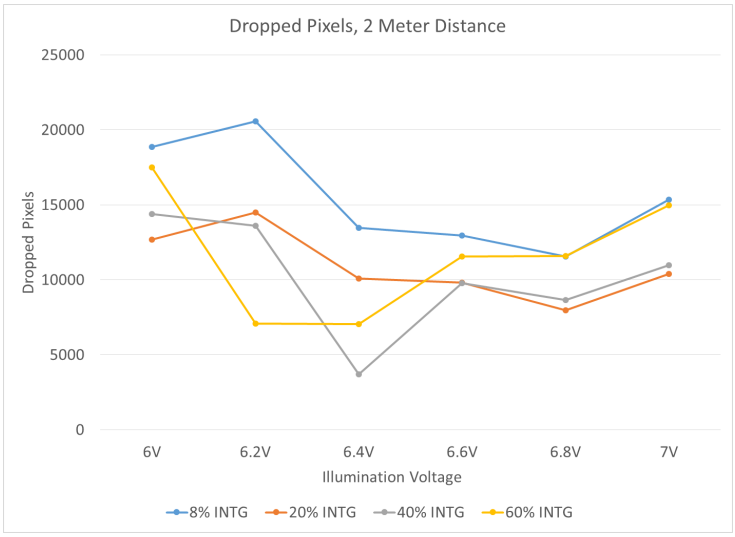

Figure 2-16: Average number of dropped pixed for a frame at a distance of 2M, for different illumination voltages and integration percentages

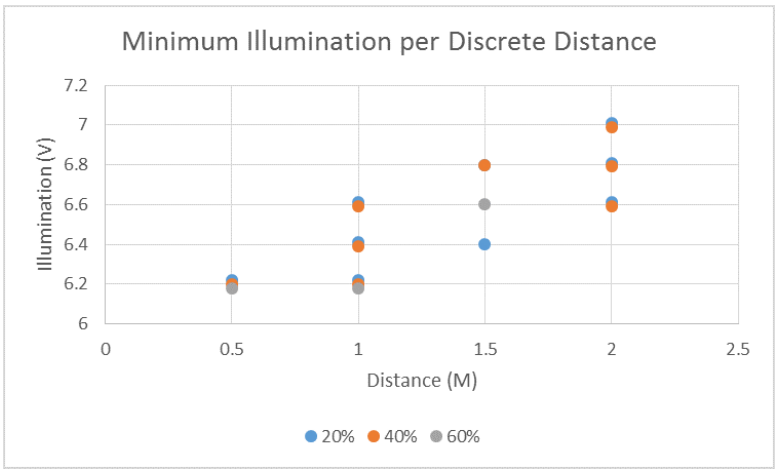

Figure 2-17: Minimum illumination and integration strictly required plotted at each distance.

to a single distance measurement value. Taking this into account, the following algorithm was used to assign illumination and integration parameter values to a particular scene.

Algorithm:

1. Compute a phase histogram, that assigns the phase value of each pixel in the 320 x 240 frame to one of five bins that partitions the camera's full range.

2. Search for the largest bin, if it contains more pixels than the noise distribution, and identify the bin's corresponding metric distance.

3. Extrapolating from 2-17, assign the current frame the appropriate illumination and integration values.

There are a few points to note about this algorithm. Firstly, notice the behavior of step 2. If a given scene is primarily at the edge of the camera's range, other objects that are still substantially large but closer in proximity will be sufficiently illuminated. If the given scene has a majority of pixels in close proximity to the camera lens, but a substantially large object exists at the periphery of the camera's range, it will not be sufficiently illuminated. This is an intentional tradeoff for this algorithm in order to reduce system level power consumption. Additionally, note that step 3 assigns the relevant parameters to each frame, but the camera's parameters may not actually be updated on a per frame basis. See the section on implementation for more details.

2.2.3 Frame Skipping and Feedback Control

One of the original components of the custom ASIC that was designed for this platform was a "frame skipping" scheme that implemented a traditional scene differencing algorithm, in order to detect if a current frame is sufficiently similar to the previous frames and can be ignored. The frame skipping algorithm [5] simply divides the input frame into nine regions and performs a pixel-by-pixel differenciation, and a voting

scheme amongst the regions determines whether or not the frame should bypass further processing for surface normal computation. Additionally, a "skipped frame" flag is raised as an output from the ASIC. The algorithm is outlined in the figure 2-18 shown below.

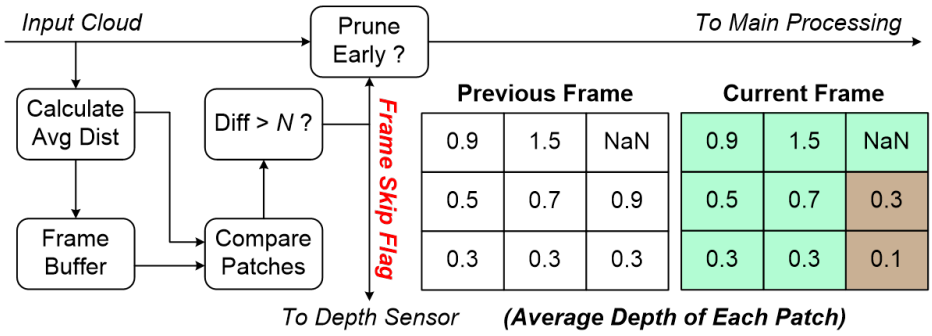

Figure 2-18: Illustration of the frame skipping algorithm as implemented in the Vision Processor ASIC [5].

In order to capitalize on this pre-existing feature, a feedback controller was written to dynamically tune the frame rate based on the skip frame output. The controller computed the "skip rate", or the number of frames skipped per ten frames delivered by the camera. The controller was then experimentally designed to keep the skip rate to thirty percent of all delivered frames. The proportional control function is as follows:

$$FrameRate_n = FrameRate_{n-1} + k \cdot (Desired - \frac{1}{N} \sum SkipRate_{n-1}) \qquad (2.5)$$

Notice that the algorithm implemented by the previous author's ASIC is a fairly primitive interpretation of a traditional rigid mean differencing approach. While the goal for frame differencing with regards to this initial prototype was simply to capitalize on and control existing infrastructure, a more robust approach has been taken for this feature on the next prototype using the TI OPT8320.

2.3 Implementation Details

2.3.1 Power Utility for a General ARM Processor

In order to test the effectiveness of the algorithms proposed above for dynamic power scaling, a custom power monitoring utility was written for the ARM processor that serves as host to the ToF and depth processing engine on the initial prototype. The power monitoring utility subscribed frame-by-frame to the point cloud data being generated by the ARM processor from the phase data and to the IMU data from the integrated MPU9250. For each frame of point cloud data, a real-world metric conversion was performed and each point was histogramed, as described in detail above. The largest bins were then computed and stored, and the distance associated with the maximum bin was assigned to the frame as an illumination constraint. This illumination constraint was computed as a moving average across ten frames. Similarly, the IMU data was processed to determine a user's average step rate across five second intervals. Lastly, skip frame flags from the ASIC were buffered and used to characterize the average skip rate per ten frames.

The illumination distance constraint was used to select the minimally sufficient illumination and integration parameters, as determined in the 2-17 above. However, in this implementation, the margins of the division indicated in 2-17 were adjusted by a linear factor of the step rate. In other words, the base illumination associated with a distance was increased or decreased slightly based on the step rate of the user. Notice that this implementation applies the step rate input to illumination control, whereas a different application approach is taken as a part of the second implementation in the latter half of this paper. Additionally, the proportional control scheme was applied to scale the frame rate, and this parameter was tuned on the camera itself by register file modifications at each time step.

2.3.2 Power Measurement Board Design

After the algorithms had been designed and implemented, it was of interest to capture the system level power expenditure with and without the application of these algorithms. Since we are interested in the power consumption of the entire wearable system as a whole, a method by which to capture the real-time power draw of the battery was needed while the user was in motion and the ToF camera was active. To do this, a small power monitoring PCB, with a current-sense resistor, amplifier, and microcontroller was designed and attached to the prototype system. The schematic, layout, and a photo of this component are presented in Figures 2-19, 2-20, 2-21 below.

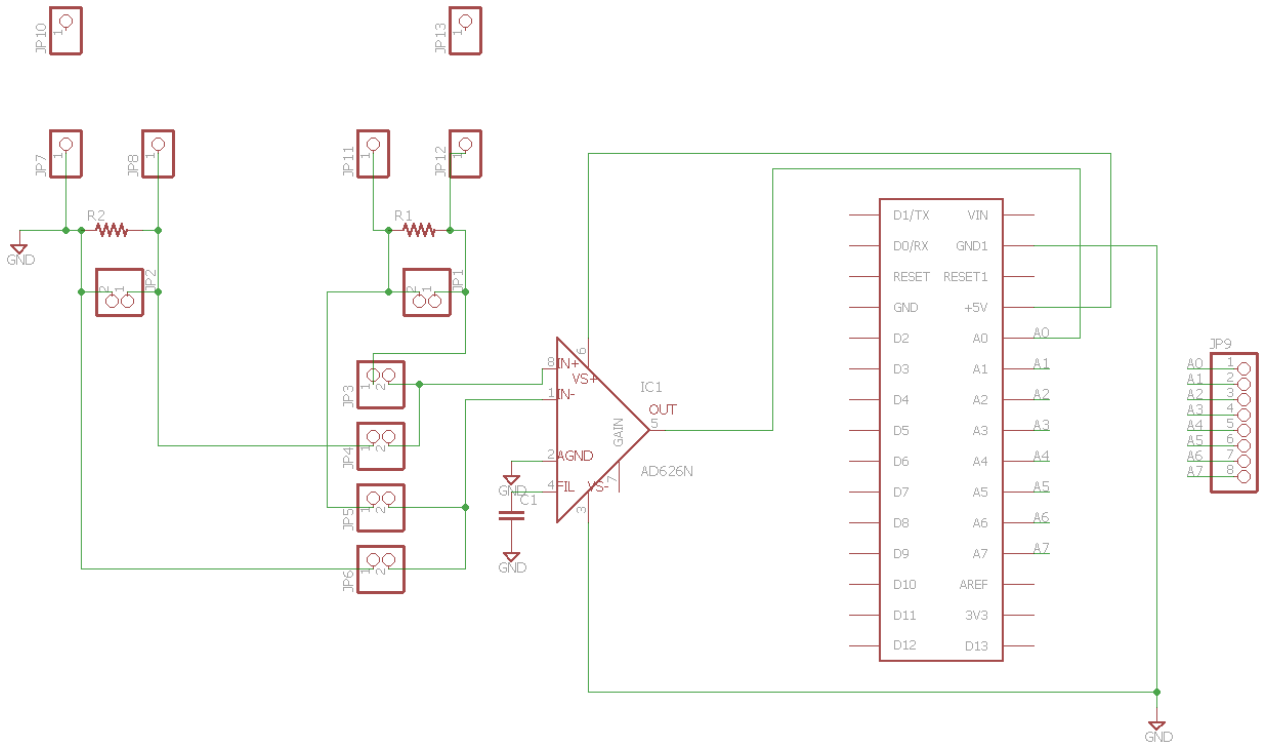

Figure 2-19: Schematic of the Power Monitor used for experimentation with the power algorithms applied to the navigation board.

The power monitor was attached to the wearable navigation prototype and mounted securely to the rear-surface for experimentation.

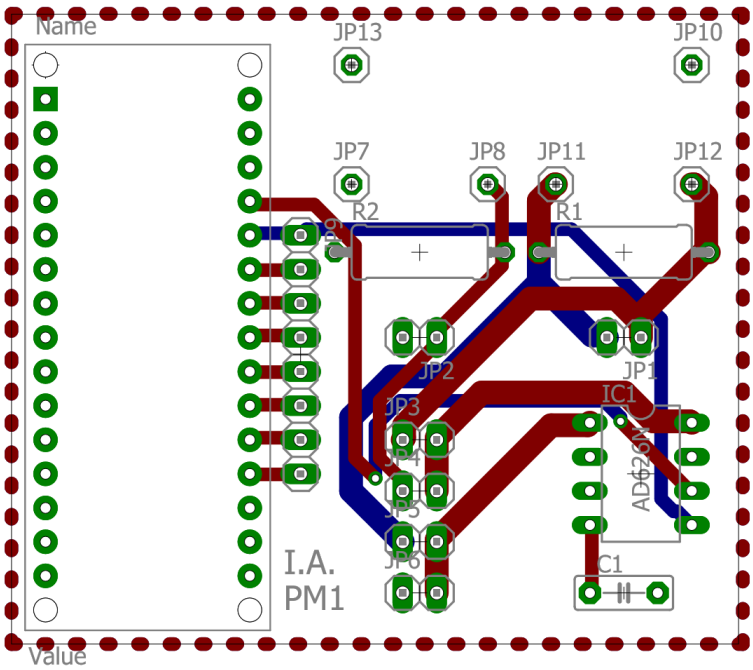

Figure 2-20: Layout of the Power Monitor used for experimentation with the power algorithms applied to the navigation board.

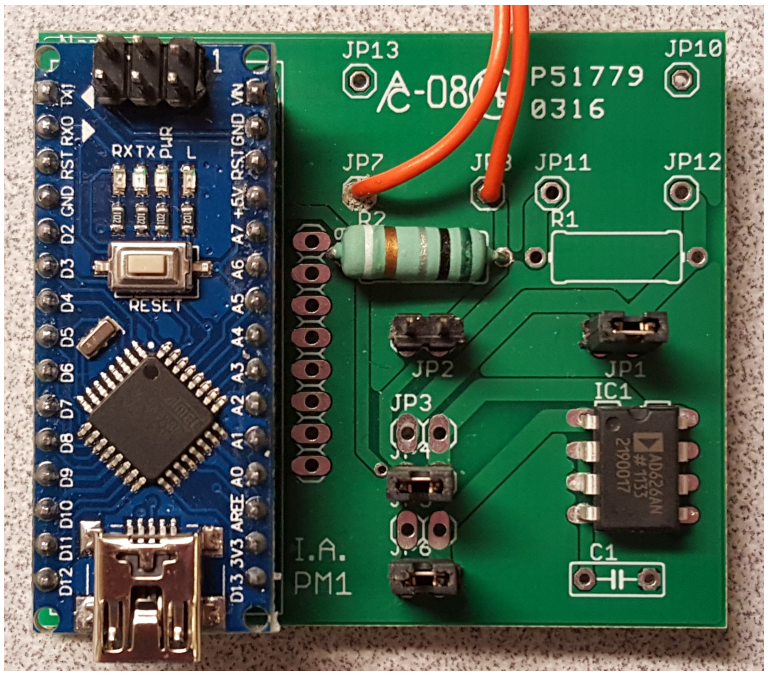

Figure 2-21: Photograph of the Power Monitor used for experimentation with the power algorithms applied to the navigation board.

2.4 Results

The experimentation was performed in the MIT CSAIL holodeck, with several obstacles arranged in the pattern of a guided maze. A layout of the space can be seen in the image 2-22 below.

Figure 2-22: Holodeck Experiment Layout. The correlated walls formed a maze which allowed for a diversity in step rate and environmental proximity between the user and the camera.

The maze was walked repeatedly by a user wearing the device, following an intentionally charted path along the obstacles that was initially informed by a trial with the braille feedback device. The course was charted in such a way as to create diversity in obstacle proximity and required pattern of motion (turns, straight paths, etc). For the purpose of relevant comparison, the user was permitted to be sighted and asked to follow the same course in every single trial. The user was asked to walk first at an average pace, then intentionally very slowly, and finally very rapidly.

2.4.1 Power Measurement Data

The plots in this section demonstrate the results of the experimentation detailed above, and it is important to note that power values plotted are measurements of all of the components on the navigation platform, together. A reduction in power due to the scaling of any feature implies that, at a constant system voltage of 5V, the current draw has decreased. First, system power consumption over the entire course without the application of any dynamic power scaling is shown in the figure

2-23 below. Notice the cyclic power consumption that results from the illumination behavior described at the start of this work.

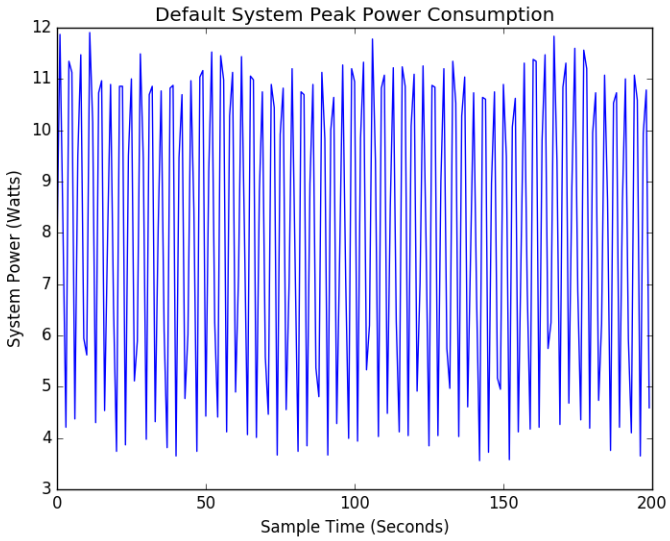

Figure 2-23: Sample power consumption plotted by power monitor at 7.5V illumination, 30fps.

Then, a comparison of the default performance with incremental application of the power scaling algorithms for each walking pace is shown in the subsequent figures. Each figure shows the average power consumption in Watts per one second across the guided maze, which resulted in a total walking duration of about two minutes. Each figure contains the power expenditure resulting from (1) the application of no algorithms; (2)the application of the frame skipping algorithm alone; (3) the application of the scene statistics algorithm alone; (4) both utilities combined and applied throughout the experiment.

2.4.2 Power Comparisons and Observations

There are several conclusions that can be drawn from the data presented above. First, the default power consumption for this demonstration setup is shown to be at an average of about 12 watts. In this demonstration setup, this is a maximum value for the average power consumption generated by the system and can be used as a point of comparison. Looking at both of the plots, it can firstly be seen that all three schemes

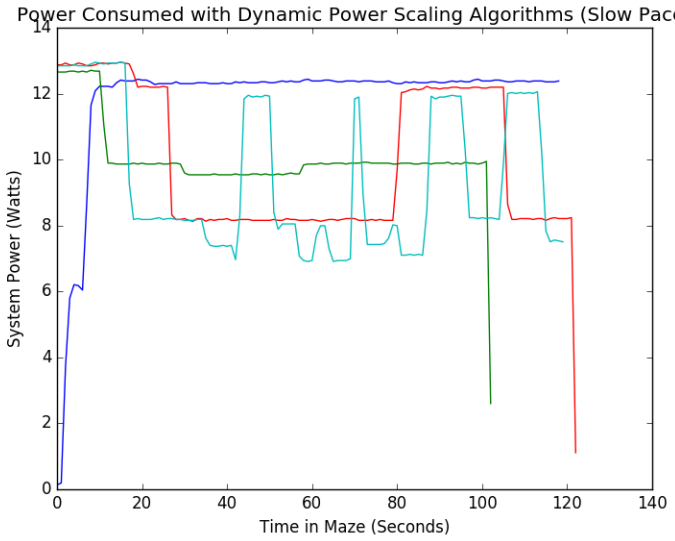

Figure 2-24: Power consumption with algorithms applied and a user walking at a slower pace. The blue line shows the default power consumption, the green line shows the application of the skip frame algorithm only, the red line shows the application of the scene analysis algorithm only, and the light blue line shows both algorithms combined.

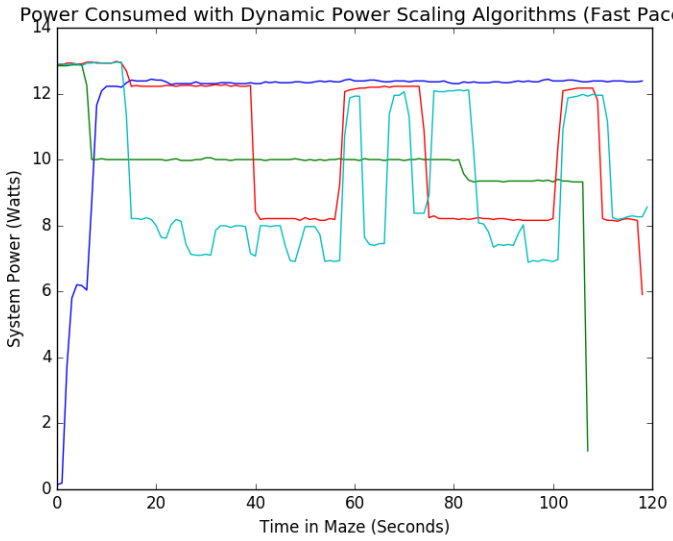

Figure 2-25: Power consumption with algorithms applied and a user walking at a faster pace. The blue line shows the default power consumption, the green line shows the application of the skip frame algorithm only, the red line shows the application of the scene analysis algorithm only, and the light blue line shows both algorithms combined.

do reduce the amount of power consumed across the path through the maze. The frame skipping control loop application, which is represented by the green line, proved to be sensitive to the rate of a user's motion, and could vary substantially across even a pre-determined path. The results of applying only the scene statistics algorithm, demonstrated by the red line on the plots, were sufficiently deterministic given a fixed path as the algorithm is largely a function of object proximity. Additionally, the scene statistics algorithm results in fairly rigid changes in power consumption across broader scales of time, without extremely gradual sensitivity to changes in the proximity of objects to the camera. As a result, the combination of the both the algorithms, shown by the light blue line, is the most effective response to the challenge of dynamic power scaling. As can be seen, it combines short-term sensitivity to user pace with long-term sensitivity to changes in proximity to the produce the most responsive scaling infrastructure.

While the first plot represents a user walking through the maze at a slow to average pace, the second plot represents a user walking at a much faster pace through the maze (and continuing on to an additional segment to maintain the same length of trial) and presents the sensitivity of the system to change in pace much more clearly. Notice that, for example, the scene statistics algorithm curve envelopes the combined algorithm curve at several locations in the maze.

Across the entire duration of the two-minute maze course, the user walking with a slow space and having applied the combined set of algorithms to the guidance device consumes 30 percent less power than in their absence, and 20 percent less power for a user walking at a faster pace. While these numbers may not seem dramatic, it is important to note here that in these experiments, a user is constantly in motion in an intentionally varied environment. Given that these results demonstrate reductions in power consumption even for slight decreases in pace or in environment proximity, it is evident that over the span of an entire day, where a user might primarily be in one place, the reduction is significant.

While the software implementation of these algorithms demonstrates potential for this application, it is next of interest to understand how these algorithms, as

well as any computationally significant features that have remained in software, can be translated to a hardware implementation for a fully integrated and miniturized platform. The following section will investigate this further.

Chapter 3

Implementation for Calculus Platform (TI OPT8320)

3.1 Background

3.1.1 Motivation for Next Generation Navigation Device

The previous chapter in this thesis has described the need and utility of power intelligent systems, or systems capable of running application specific algorithms for the dynamic scaling of power, primarily because the initial prototype system consumed power at levels that would render it practically inutile as a wearable device. The next step in this investigation of guidance wearables was to focus on the development of a system that consisted of a ToF camera and processor that was designed to inherently consume less power, trading off range and accuracy in its place. An additional goal was to determine whether the major portion of the computation being done currently with software in the old system– pointcloud processing and set of power scaling algorithms– could be translated into a hardware implementation. Understanding the requirements for a fully integrated system, without the need for an on-board CPU, is a critical step in moving towards a more practical, power friendly solution for the task of guiding the visually-impaired.

This section of the work focuses on the use of Texas Instruments' newest ToF

camera, the OPT8320 Calculus, and it's associated evaluation platform. The relevant technical details of the device and a comparison to the previous ToF chipset is presented in the section below.

3.1.2 Relevant Platform and System Specifications

The OPT8230 ToF camera is QQQVGA platform, with a frame dimensionality of 80 x 60. This is 16x smaller than the OPT9221/9220 platform, with a QVGA dimensionality of 320 x 240. It has a pixel pitch of 30 microns per pixels, where the 9221 had a pixel pitch of 15 microns per pixel. This implies that, though the 8320 camera has a smaller field of view, it contains more information per pixel than the 9221 family. The maximum range of the 8320 platform given the evaluation board that was provided for the work by TI is no more than 1.5 meter, whereas the maximum range of the standalone 9221 platform (without the lower power LEDs that were included in the first prototype) could be 4 - 5 meters. Additionally, the 8320 is capable of direct output frame rates of upto 4000 frames per second, rendering it much more suitable for rapid motion applications that the 9221 platform, which had a maximum of 120 frames per second.

Lastly, the 8320 chipset consists of only a single IC, which has the timing generator, controller, and depth engine integrated into a single processor. The 9221 family consists of a set of two chips, which ultimately drives the system power requirements up. The specifications regarding power consumption and comparisons can be found in the sections below.

3.1.3 Power Consumption and Power Model of the System

As mentioned above, Texas Instrument's OPT8320 chipset in nearly identical in operation to its predecessors, the OPT9220 and OPT9221. The illumination power of the evaluation system containing this chipset can also be governed by the model presented in Chapter 2. However, because these infrared LEDs require a smaller voltage (and hence, allow for a shorter range of detection) and because the camera frame has

a smaller dimensionality (80x60 in place of 320 x 240), the system has much lower characteristic power curves. The power of the system as a function of the relevant parameters can be seen in the figures 3-1 and 3-2 below.

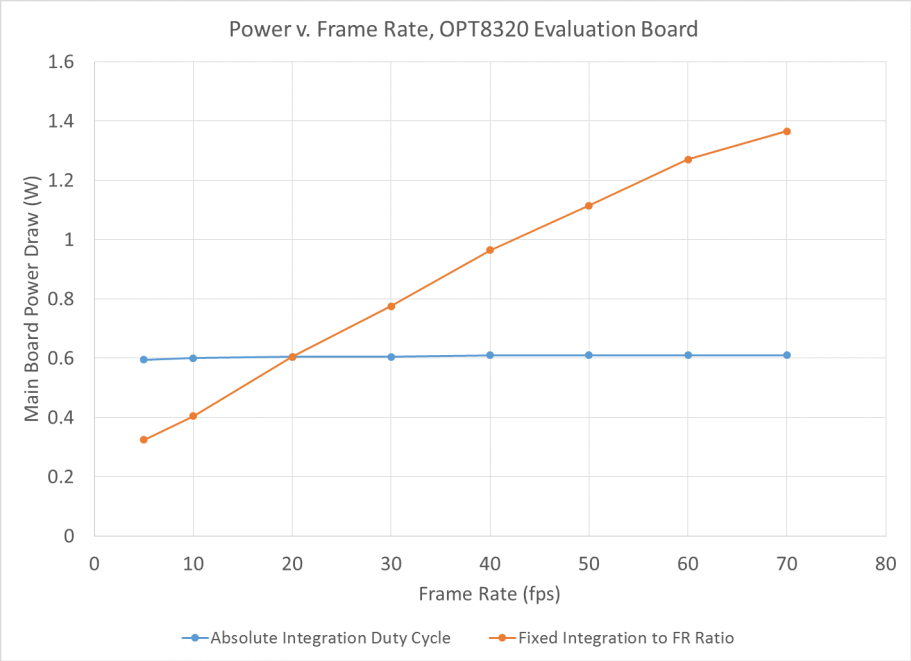

Figure 3-1: OPT8320 main board power consumption as a function of frame rate, at illumination panel current of 30mA. The blue plot shows a fixed integration duty cycle of 20 percent, where the orange plot displays a fixed ratio equal to the integration duty cycle over frame rate, set here at 0.01.

This system, as it stands in the evaluation platform, consumes approximately 2 Watts at peak. However, the camera is designed to be interfaced with custom illumination panels, depending on the range required and the power budget available, which are likely to consume much more power. These plots show the variability in the power that results from these features being scaled.

3.2 Translating Power Optimization Algorithms into Hardware

The remainder of this section will focus on a hardware implementation of the point-cloud computation and dynamic power scaling algorithms, with some modifications

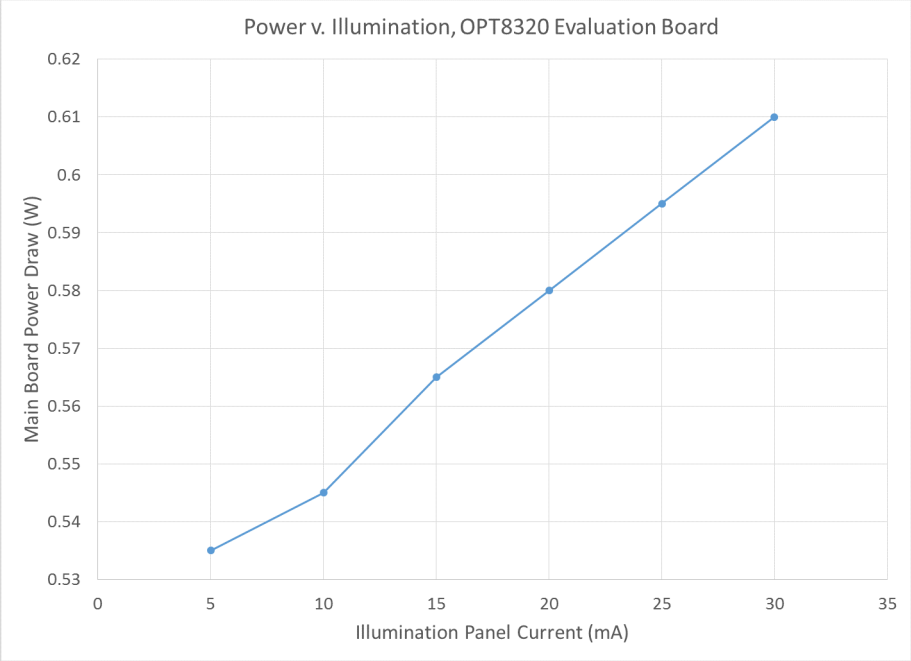

Figure 3-2: OPT8320 main board power consumption as a function of illumination, at frame rate of 30 fps and 20 percent integration.

to the original prototype implementation, as a model for the way in which such components could be completely integrated into a wearable system. The details presented in the sections below will aim to demonstrate that power can effectively and robustly be scaled by hardware automatically, through dedicated functionality as a part of an ASIC or through a dedicated ASIC itself.

As a demonstration, the system proposed and detailed in this section was implemented on a Zynq Miniitx100, a platform that consists of a standard Xilinx FPGA and an ARM processor running a full Linux operating system. It is important to note that, given the constraints on time and limited access to the evaluation hardware, data streaming from the camera and parameter configurations being written back to the camera were still implemented in software on the ARM processor, though this would extremely straightforward to integrate fully into hardware in a future wearable system. The software-to-hardware interface was written as a Connectal interface, while the FPGA RTL was written in Bluespec System Verilog and simulated initially in Bluesim.

While algorithms and architectures were constructed jointly, two of the four main

Bluespec module implementations (point cloud and scene differencing) in this work have been contributed by Andre Aboulian as a part of a joint final project for the Complex Digital Systems Design class. He has also been gracious enough to provide several of the figures, as noted below.

A high-level overview of the demonstration system can be seen in the system diagram shown in 3-3.

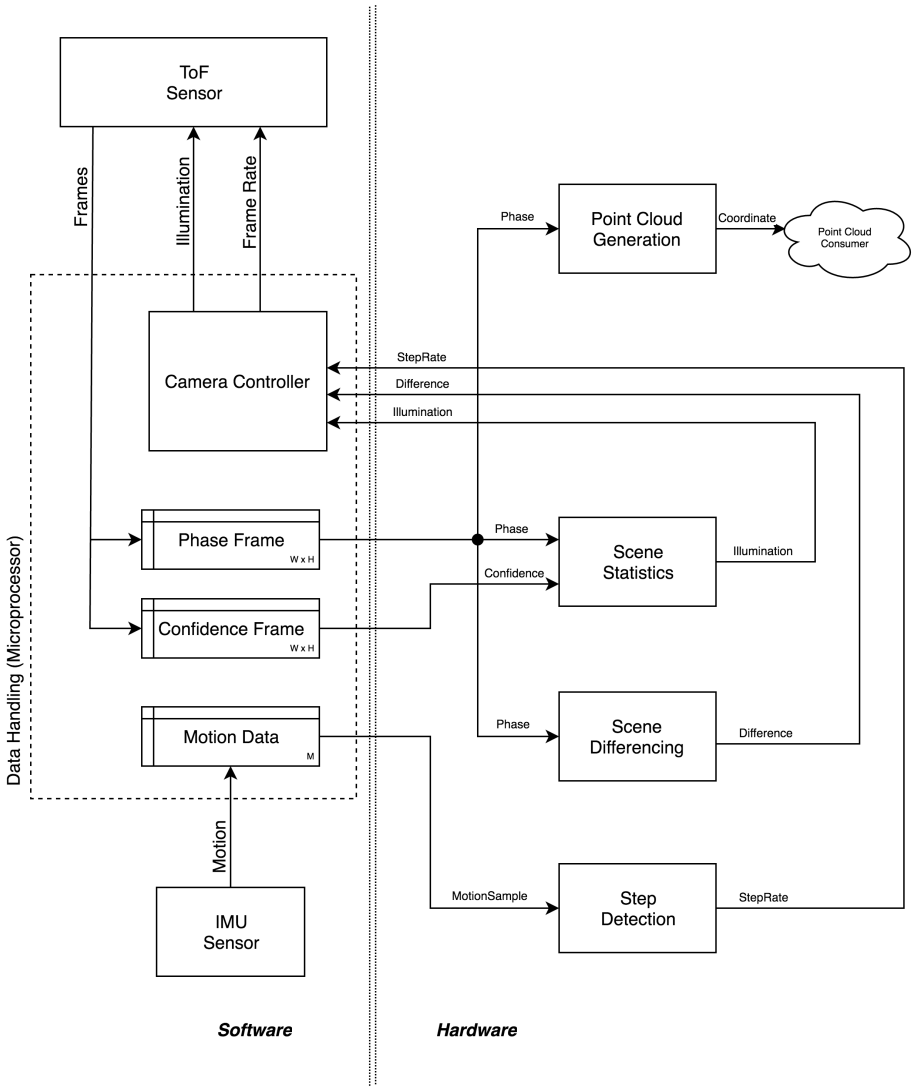

Figure 3-3: System overview of the hardware implementation for the dynamic power scaling algorithms. Courtesy of Andre Aboulian.

3.2.1 Point Cloud Generation: Theory and Architecture

In the initial navigation prototype, the most consistent and computationally hefty operation needed to be performed was the generation of the pointcloud. The point-cloud transformation is an operation required to performed pixel-by-pixel, frame by frame, and does not scale well with larger dimension frames. This operation, which is required for the guidance application and most other robotics applications that might use ToF systems, was made more efficient by an FPGA implementation that capitalized on redundancy within a pinhole camera model based geometric transformation. First, the geometric transformation that was architected for this project is derived below.

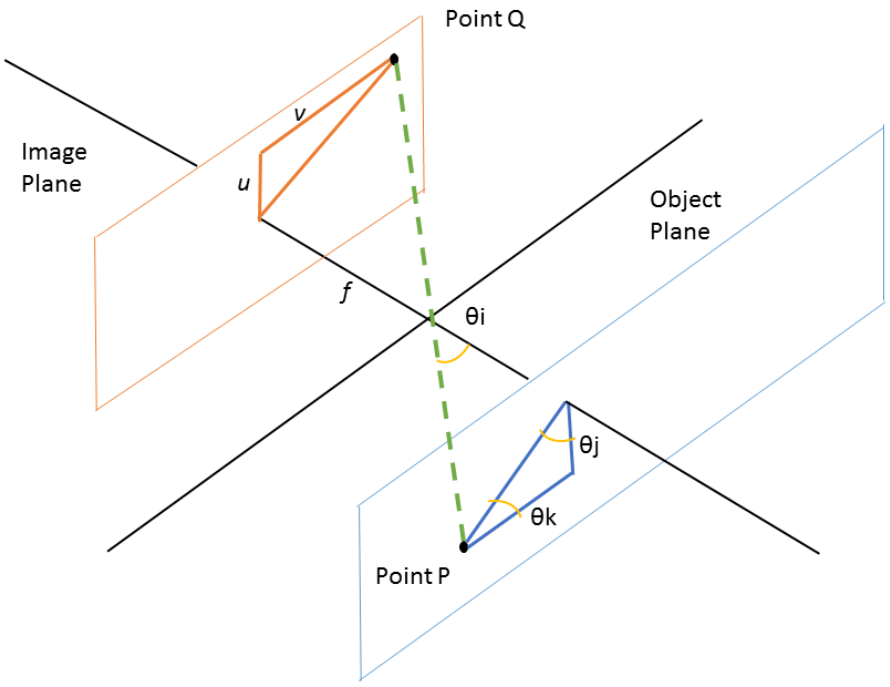

Figure 3-4: This figure shows the pinhole camera model that was used in the point-cloud transformation for this study. We find point P, the point in the real world, from point Q, the point on the image plane, and the depth ray that is read by the imager.

Using the sample model shown in 3-4, we can find the X, Y, Z coordinates of the Point P in the image by the following projections.

$$\begin{aligned} X_P &= Depth_{XYZ} \cdot cos(\theta_i) \\ Y_P &= Depth_{XYZ} \cdot cos(\theta_i) sin(\theta_j) \\ Z_P &= Depth_{XYZ} \cdot cos(\theta_i) sin(\theta_k) \end{aligned} \tag{3.1}$$

Expanding further with trigonometric substitutions we have:

$$\begin{aligned} X_P &= Depth_{XYZ} \cdot \sqrt{\frac{f}{1+u^2+v^2}} \\ Y_P &= Depth_{XYZ} \cdot \sqrt{1 - \frac{f}{f+u^2+v^2}} \cdot \frac{u}{\sqrt{u^2+v^2}} \\ Z_P &= Depth_{XYZ} \cdot \sqrt{1 - \frac{f}{f+u^2+v^2}} \cdot \frac{v}{\sqrt{u^2+v^2}} \end{aligned} \tag{3.2}$$

where (u,v) represent the coordinates of the Point Q on the image plane, and f is the focal length of the lens. Defining the depth in terms of the known quantity of phase $\phi(u,v)$, we have:

$$\begin{aligned} X_P &= \left[\frac{c}{2f_{mod}} \cdot \frac{\phi(u,v)}{2\pi}\right] \cdot \sqrt{\frac{f}{1+u^2+v^2}} \\ Y_P &= \left[\frac{c}{2f_{mod}} \cdot \frac{\phi(u,v)}{2\pi}\right] \cdot \sqrt{1 - \frac{f}{f+u^2+v^2}} \cdot \frac{u}{\sqrt{u^2+v^2}} \\ Z_P &= \left[\frac{c}{2f_{mod}} \cdot \frac{\phi(u,v)}{2\pi}\right] \cdot \sqrt{1 - \frac{f}{f+u^2+v^2}} \cdot \frac{v}{\sqrt{u^2+v^2}} \end{aligned} \tag{3.3}$$

Finally, since u and v are not incrementally spaced across the image plane due to curvature in field of view, and due to the fact that our model has the image plane centered along the focal length, we finally have:

$$
\begin{aligned}
X_P &= \left[\frac{c}{2f_{mod}} \cdot \frac{\phi(u,v)}{2\pi}\right] \cdot \sqrt{\frac{f}{1+u^2+v^2}} \\
Y_P &= \left[\frac{c}{2f_{mod}} \cdot \frac{\phi(u,v)}{2\pi}\right] \cdot \sqrt{1-\frac{f}{f+u^2+v^2}} \cdot \frac{u}{\sqrt{u^2+v^2}} \\
Z_P &= \left[\frac{c}{2f_{mod}} \cdot \frac{\phi(u,v)}{2\pi}\right] \cdot \sqrt{1-\frac{f}{f+u^2+v^2}} \cdot \frac{v}{\sqrt{u^2+v^2}} \\
u &= (x - \frac{Width}{2}) \cdot \tan(\frac{FOV_x/2}{Width/2}) \\
v &= (y - \frac{Height}{2}) \cdot \tan(\frac{FOV_y/2}{Height/2})
\end{aligned}
\tag{3.4}
$$

where FOV_x and FOV_y are the horizontal and vertical fields of view respectively, $Width$ is the width of the frame in pixels, $Height$ is the height of the frame in pixels, and x and y are the indices of the Point Q as read from the ToF sensor array.

A microarchitecture diagram of the pointcloud module is shown in 3-5. To avoid excessive online computation when processing phase frames into point cloud data, lookup tables are pre-generated for all terms to the right of the depth term in equation 3.4. This is only done for a single quandrant, as the values are symmetric, and the assignment of sign is computed based on the indices assigned to the incoming phase pixel. The LUTs are stored in a BRAM and accessed pixel by pixel, as shown in the system overview.

3.2.2 IMU Pose Estimation: Theory and Architecture

The algorithm implemented as a part of the IMU pose estimation is identical to the algorithm presented as part of the initial prototype. The block operates by receiving as input FixedPoint accelerometer measurements from an external USB MPU9250 at a sample rate of 20 Hz. The data stream is chunked into frames of 128 data points, which allows each frame to contain approximately 6 seconds worth of step information. These frames are then passed through an Overlap-and-Sample block which delivers subsequent frames with 50 percent overlap to the previously delivered frame. The output is then passed through a custom super-folded, pipelined FFT implementation

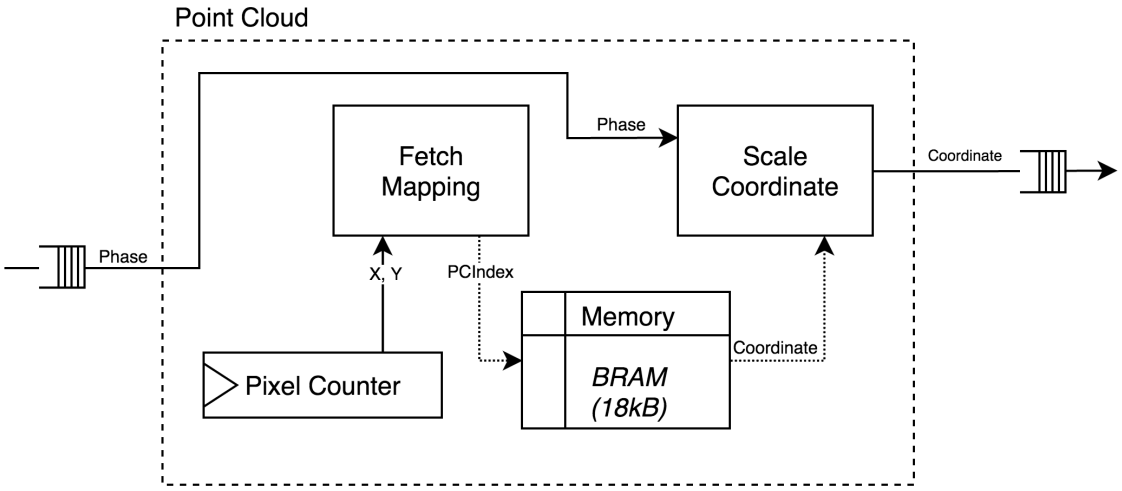

Figure 3-5: A microarchitecture diagram of the pointcloud generation module. Courtesy of Andre Aboulian.

and then through a Cordic block in order to obtain a magnitude representation of the data. Lastly, the magnitude spectrum of each frame is passed to a peak-finding block, which computes the step rate associated with the frame and returns the data to the ARM processor's camera control module.

A high level architecture of the Superfolded FFT can be found in Fig. 3-6. A radix-2, Pease transform based implementation is used. Note that the implementation uses only single Butterfly node to minimize logic and area utility. Additionally, the microarchitecture for the step rate block can be found in figure 3-7.

3.2.3 Scene Differencing: Theory and Architecture

In this iteration of the power scaling algorithms designed for the OPT8320, an algorithm that was more robust to sharp changes from frame to frame resulting from noise or vibrations between the artificially induced divisions was preferred. As a result, phase pixels in each frame are smoothed with a Gaussian filter to prevent false triggering. The gaussian convolution used in this approach is a 3x3 kernel.

The module operates as follows. The phase data enters the module as a single 12 bit phase pixel at a time. The pixels are first chunked into columns of height H. These vectors are passed to the convolution block, which applies a gaussian kernel

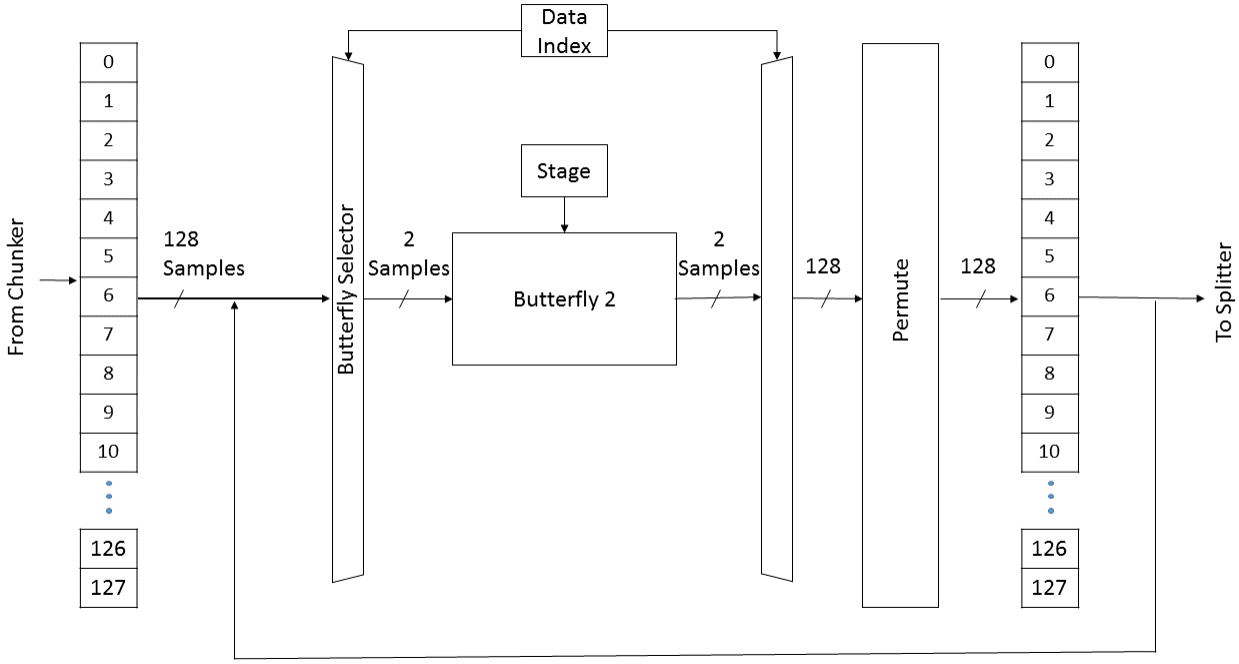

Figure 3-6: High level diagram of the superfolded circular FFT module used to compute the step rate in the IMU module.

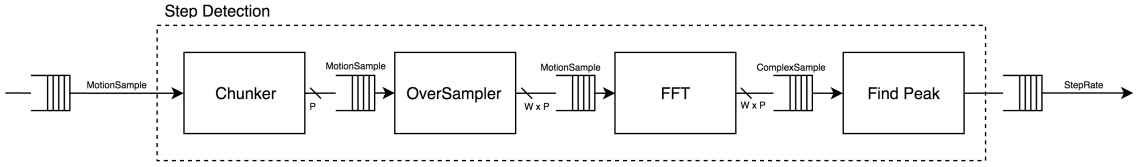

Figure 3-7: A microarchitecture diagram of the step rate module. Courtesy of Andre Aboulian.

on each pixel. The convolution block maintains a rolling K-width window in order to apply the K x K kernel. In other words, the oldest column is discarded as a new column is received from the chunker. The gaussian pixels are compared with the corresponding pixels from the previous frame. If they differ by more than a specified threshold, an accumulator value is incremented. After the comparison, the computed gaussian replaces the stored one from the previous frame. Once this occurs for all the pixels in a frame, the module returns the comparison of the accumulator value to a pixel quota threshold, providing N - 1 boolean operators ("skip flags") to the camera controller for N frames fed to the module's chunker. A detailed diagram of the scene differencing microarchitecture can be seen in the figure 3-8.

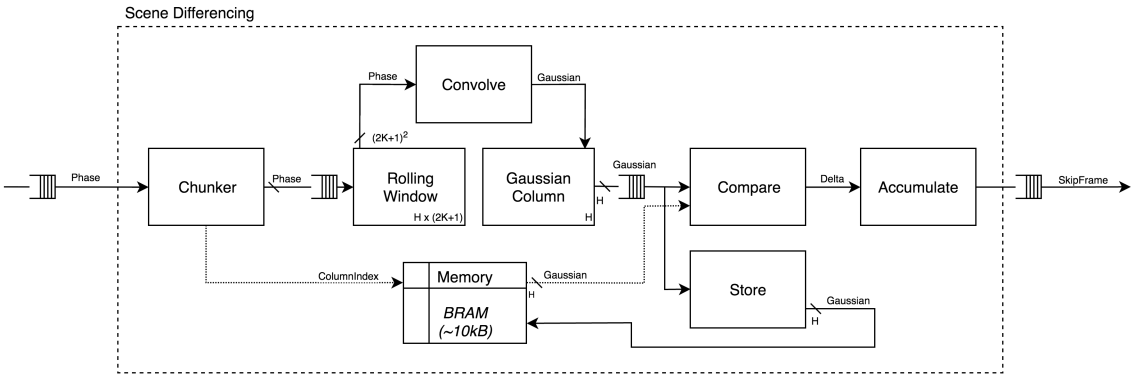

Figure 3-8: Microarchitecture of the scene differencing module. Courtesy of Andre Aboulian.

3.2.4 Scene Statistics: Theory and Architecture

The scene analysis algorithm follows a procedure similar to the previously mentioned approach, with a few differences. First, a priori data was collected regarding the minimum confidence that was required at discrete distance markers for accurate point cloud generation. This was completed with a simple table top experiment that included objects placed at consecutively farther and farther distances from the camera lens, while the illumination was gradually varied until a pointcloud could not longer be represented for a given object. This experiment returned a non-linear confidence threshold as a function of metric distance.

This threshold function is then imposed on a generated phase histogram, and the largest N bins that contain a set percentage (25 percent, in all experimental trials) are selected for analysis. The ratio of the number of pixels above the threshold to the total pixels per bin is computed for each of the N bins, and if the ratio is in the first or third quartile, a vote is assigned to increase or decrease the illumination. The final result is a weighted sum of the voting scheme that can be tuned for distance sensitivity. The algorithm can be summed up succintly as below:

Scene Analysis Algorithm (for OPT8320):

1. Compute the histogram H_p for all 4800 pixels, assigning bins with a resolution of 0.15m per bin

2. Impose confidence threshold function upon the histogram, $t_c(d)$, where d is distance

3. Select the N largest bins that comprise of M percent of the total pixels in the scene

4. Compute the direction of change in illumination as:
   $v_p(n) = \frac{t_c(n)}{H_p(n)} \leq 0.25$
   $V_p = \sum_N v_p(n)$
   $v_d(n) = \frac{t_c(n)}{H_p(n)} \geq 0.75$
   $V_d = \sum_N v_d(n)$
   $\Delta I = sign(w_1 V_p + w_2 V_d)$
   where v_p and v_d are postive and negative votes respectively assigned to a single bin, V_d and V_p are the frame's collective positive and negative votes, and w_1 and w_2 are tunable weights that can be set if, for example, a low-power mode values power reduction assignments more than power increase modes, or vice versa.

5. Return ΔI

Notice here that the proposed scheme only allows the camera's illumination parameter to be increased or decreased one step at a time. This was intentionally

designed to suit the camera's fairly course granularity descernible for a small LED panel. This algorithm can be expanded to include a control loop at the last step for a camera with greater resolution in illumination control.

A microarchitecture diagram of the scene statistics module exhibiting the algorithm discussed can be found in figure 3-9.

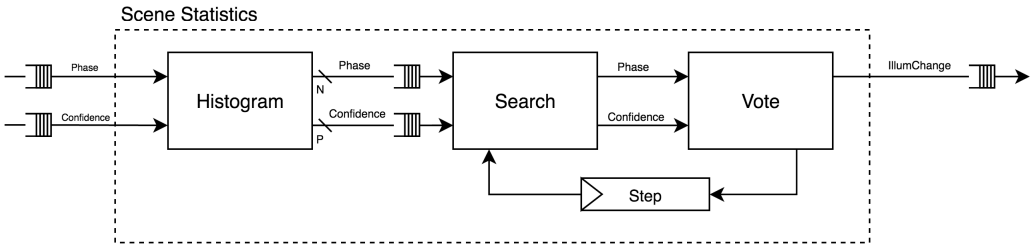

Figure 3-9: Microarchitecture of the scene statistics module. Courtesy of Andre Aboulian.

3.2.5 Camera Controller

The camera controller is the term in this work being used to describe any component of the demonstration system that is hosted on the ARM processor and that has infrastructure in software. As mentioned above, this camera controller is multifaceted, and includes the interface to deliver data from the camera, update the parameters of illumination, integration, and frame rate, and also a simple control algorithm to aggregate the results of the individual bluespec blocks before computing the required parameter updates.

The camera controller maintains a moving average of the illumination delta Δi_t that is assigned to each individual frame, with the ceiling of result appended to the current illumination voltage value to obtain I_t, the new illumination value.

$$\begin{aligned} I_t &= \frac{1}{n}\sum_{i=0}^{n-1} \Delta i_{t-i} \\ I_t &= I_{t-1} + \left[\frac{\Delta i_t}{n} - \frac{\Delta i_{t-n}}{n}\right] \end{aligned} \tag{3.5}$$

The camera controller additionally maintains a moving average of the skip frame binary values from the scene differencing module to compute the skip rate across

ten frames. This mechanism is not needed for the step rate detector, as the architecture already accounts from overlapping IMU frame data, and as each IMU frame is already several seconds long to prevent frame-cutting artifacts that induce noise in step detection. Using this information, the frame rate control is maintained in a similar fashion to the previously detailed algorithm, with a slight modification that capitalizes on the step rate. The modified control equation is shown below:

$$FrameRate_{n+1} = FrameRate_n + (k + q \cdot StepRate)(Desired - \frac{1}{N}\sum SkipRate_n) \tag{3.6}$$

where k is the proportional gain term, and q is a tuning factor regarding the degree to which the gain can be pertubed by the step rate. Intuitively, we seek to accelerate the change in step rate based on the presence or lack of rapid user motion.

3.2.6 Software Simulation

As a first step, a software simulation for each of the four blocks and the camera control algorithm was written in a python framework as a reference for the BSV implementation. The camera was simulated with a "frame server" that delivered pre-captured data to represent a particular frame rate and each of the specific illumination values. Each of the modules was bit accurate with the reference implementation and used for a comparison of the output results within a BSV test bench.

3.2.7 Potential Control Challenges

As with any control system, there are a few factors that could give rise to unstability in the power scaling functionality. For example, frequent oscillations in or perturbations to the input data due to a rapidly changing environment (such as if the user is on a busy street) or due to a user changing pace quickly would cause the power settings to oscillate. Additionally, the confidence metric might vary between an indoor setting and an outdoor setting, as there is more ambient light outdoors. This might cause

unpredictable behavior as the camera might assign inaccurate confidence values to depth pixels, which would result in a delayed or poor response from the scene statistics algorithm. Other similar factors might include noisy accelerometer data reflecting motions other than step rate, or short periods of high activity on the input to the controller that might be missed due to low frame rates from a previous setting.

3.3 Results

3.3.1 Timing Analysis, Testing, and Visualization

The entire system is clocked at 60MHz, where the critical path length results from the scene statistics histogram analysis that operates on an entire frame of 4800 phase values at one time. This suggests that this power scaling algorithm would not scale well as it stands for larger frame dimensions, and a principle of segmenting the histogram would need to be introduced for other applications. Given the number of clock cycles required for each module to process a single ToF frame delivered by the camera, the following table 3.1 summarizes the throughput of each of the modules.

Module	**Throughput**	**Frame Delivery Constraint**
Scene Statistics	12Khz	4Khz
Step Rate (IMU)	100Khz	20Hz
Scene Differencing	12.5KHz	4Khz
Point Cloud	12.5KHz	4Khz

Table 3.1: Table that shows the frequency at which each module outputs a result, constrained by the rate of delivery of new frames from each module's respective sensor shown in the column on the right.

It should be noted that frequency of frame delivery listed in the table above is the maximum internal frame rate of the 8320 chipset as per the device specifications– the maximum frequency for demonstration purposes through a USB interface is 2000 frames per second. Even at the maximum frame rate, the hardware implementation is capable of processing each frame into a point cloud and providing feedback to the camera controller without latency, demonstrating that the architecture is a viable solution.

For demonstration and testing purposes, strategic input test data was generated using the OPT8230 ToF camera. For example, to test the frame rate scaling algorithm, several test vectors of input phase frames during which the camera was moved rapidly were recorded. Additionally, to test the scene statistics output, test vectors that slowly varied the proximity of the camera sensor to a fixed wall in varying lighting conditions were also generated. A table summarizing the various test cases is shown in the table 3.2 below.

Test Dataset Label	**Tested Module**	**Test Condition**
A	Scene Differencing	Alternating steady and shaken camera data
B	Scene Differencing, Step Rate	Alternating steady and shaken camera data
C	Scene Statistics	Increasing the distance between the camera and the wall
D	Point Cloud	Sample distinctive objects for verifying pointcloud visualization

Table 3.2: Table that lists the four test cases for which strategic input data was generated. The resulting data is shown in later figures.

The resulting feedback provided by the power algorithms for each of the test cases and sample visualizations of the point cloud generated by the BSV implementation are shown below.

The results demonstrate the end-to-end functionality of the system. They are also indicative of the power savings that could be obtained in a fully integrated next generation prototype. For example, while the power quantity itself would vary with the chosen illumination panel, scaling the camera from 60 fps to 10 fps presents a 70% reduction in power. Similarly, even for the small default LED panel, there is 15% reduction from the maximum to the minimum possible illumination value. This suggests immense potential for a wearable device mounted on a user who might spend a majority of his or her day seated at an office or in an unchanging environment. Overall, these results demonstrate that this smaller ToF device, in conjunction with the custom power scaling algorithms, is a much more suitable alternative for the ToF

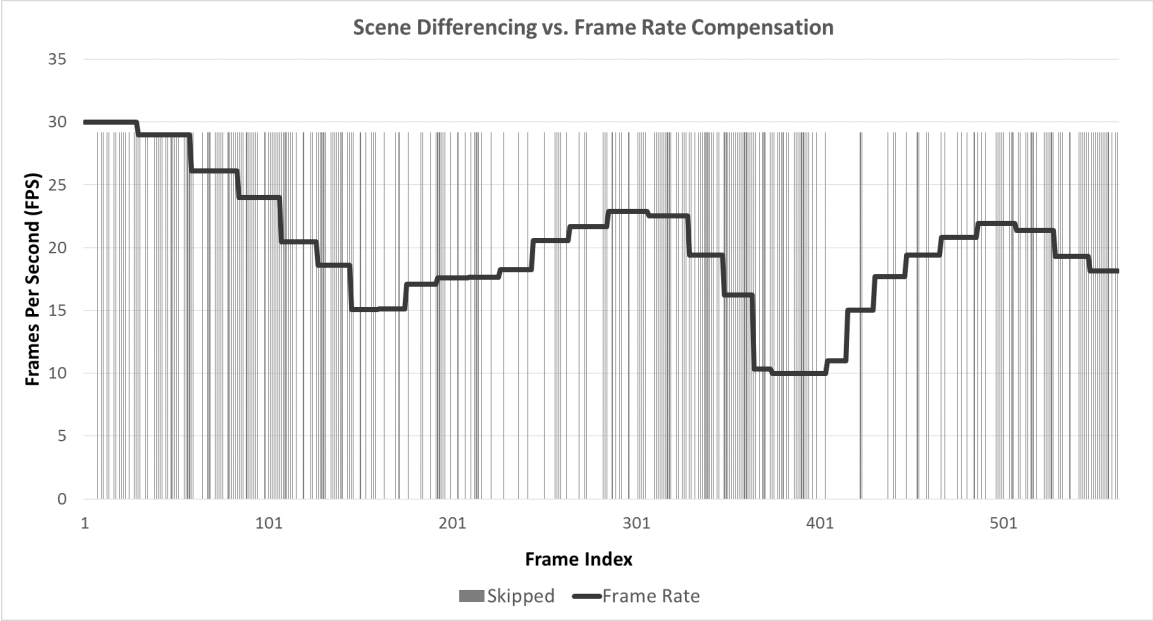

Figure 3-10: Test Case A: The frame rate compensation is plotted with the density of skip flags being produced by the scene differencing module. The camera is alternated between being held still and then shaken rapidly at 5 second intervals for a total of approximately 25 seconds.

camera in a future iteration of the guidance system.

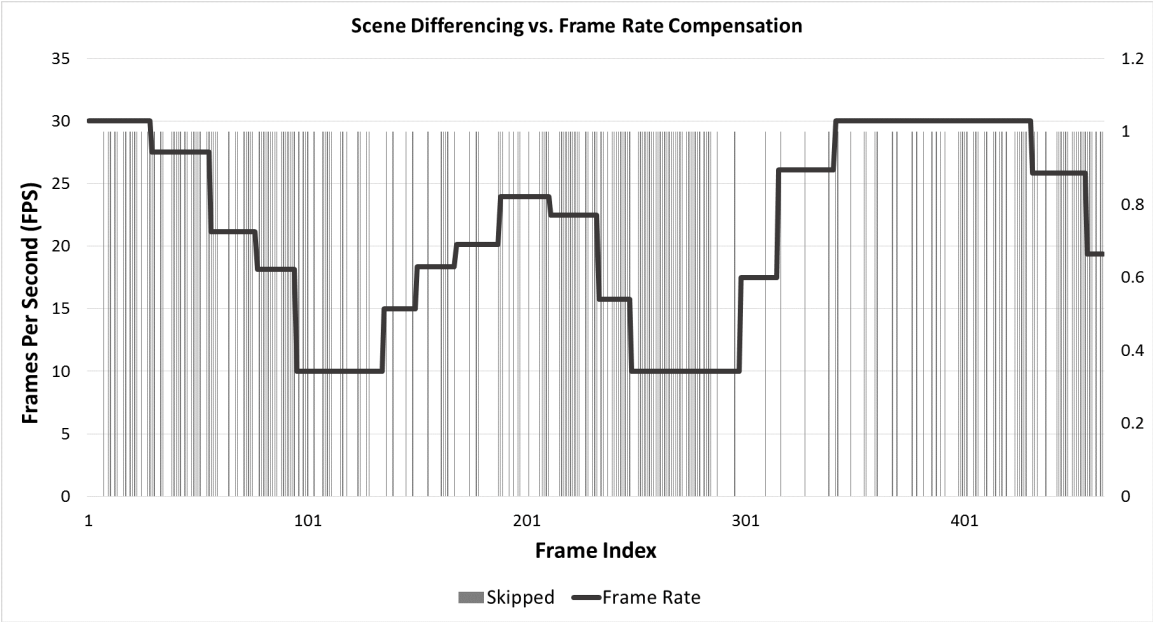

Figure 3-11: Test Case B: The frame rate compensation is plotted with the density of skip flags being produced by the scene differencing module. The camera is again alternated between being held still and then shaken rapidly at 5 second intervals for a total of approximately 25 seconds, but the gain of the control loop is now perturbed by the step rate, resulting in sharper increases and decreases.

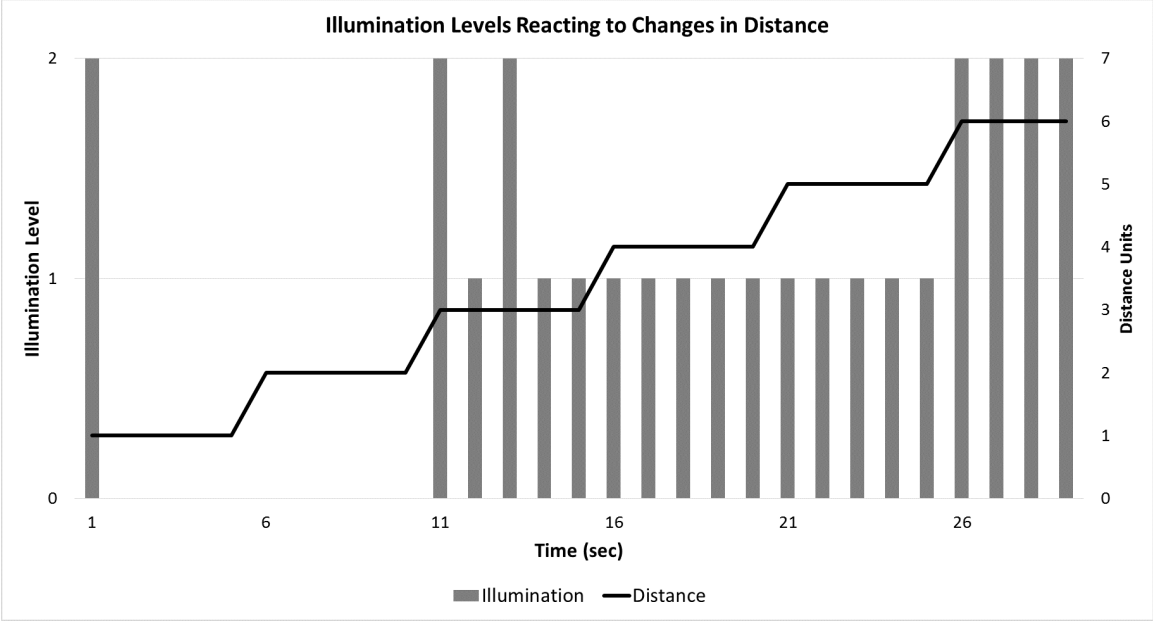

Figure 3-12: Test Case C: The camera is slowly drawn further and further away from a wall at 6 different stages of distance in 5 second intervals. As can be seen, the illumination scales appropriately by selecting one of the three stages of illumination – 0, 1, or 2. The noise at the transition points likely results from the unpredictable motion of the camera as it was drawn away, or the presence of an artifact such as a hand, etc.

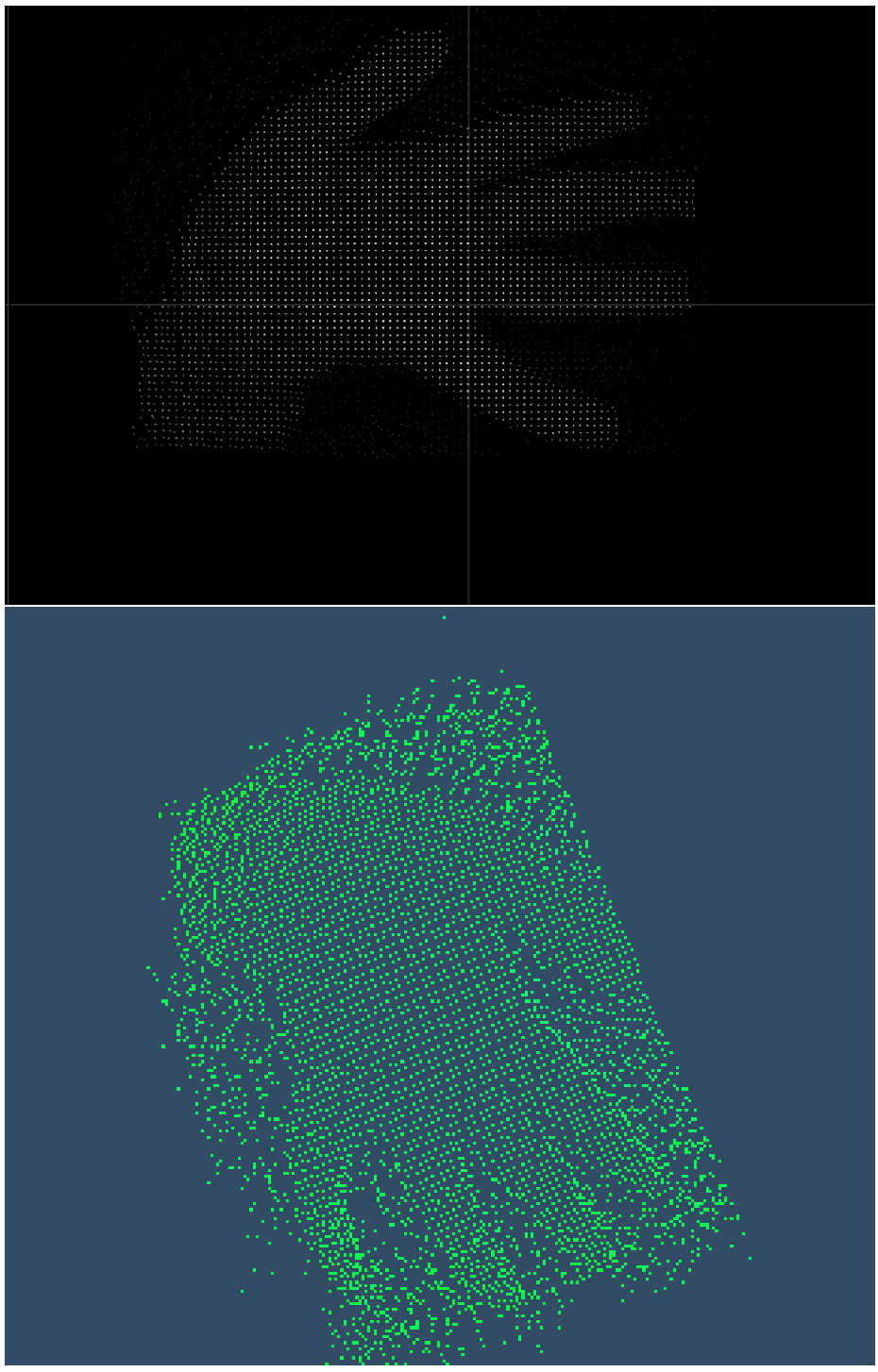

Figure 3-13: Test Case D: Visualization of the pointcloud generated by the pointcloud module. The top image is the pointcloud rendered by TI's VoxelViewer software from the phase data. The bottom image is the pointcloud produced by the module in this system from the same phase data.

Chapter 4

Conclusion

4.1 Impact

The biggest impact of this thesis is the application of simple, application specific image processing and signal processing techniques to dynamically reduce analog power consumption system-wide. While this work has targeted a wearable navigation platform as an example implementation, the key takeaway is a concept that can be applied broadly to wearables and future generations of consumer electronics– that domain specific algorithms that are fully integrated into a system so as not to be significant computationally can bridge the gap between the need for frequent battery charging or replacement and uninterrupted usability throughout the day.

4.2 Summary of Contributions

The work began with an introduction of the challenges and historical approaches to the development of such a device, with an emphasis on limitations resulting from excessive power consumption. A detailed explanation of the system prototyped in the Energy Efficient Circuits group, the first wearable navigation platform, was presented as necessary background.

In chapter 2, this platform was used as experimental grounds for developing a set of algorithms that capitalized on input available from the device to dynamically

scale power. The features most relevant to the scaling of power in this system were studied and exposed, and simple algorithms that utilized accelerometer data from the on-board IMU, performed scene analysis on phase data frame-by-frame, and tuned frame rate based on existing scene differencing infrastructure from the custom ASIC were designed and implemented in software. Physical experimentation was conducted with these implementations and results were presented.

In chapter 3, additional goals to further the research on such a device were introduced. Firstly, the newer OPT8320 ToF camera was characterized and employed as a potential alternative to the OPT9220/9221 family that is more power friendly and appropriately dimensioned for the input data that is required. Next, the most computationally hefty features that were previously computed in software, including pointcloud generation and the three power scaling algorithms, were architected and implemented in Bluespec System Verilog for FPGA operation. Operation of each individual module was verified, and test infrasture was built to allow strategic input data to be read, processed, and returned for offline verification. Sample results from this test infrastructure were presented, and it was concluded that, since common functionality could be obtained between the old and the new system, the OPT8320 is a suitable camera for a future iteration of the guidance prototype.

4.3 Future Work

The next most critical step in this work is to combine the results from several months of experimentation with the two systems into the next integrated prototype. This system would be capable of consuming no more than 2-3 watts at peak, with a much lower average power consumption due to the presence of similar power scaling algorithms. It would also be important to consider the elimination of unnecessary experimentation features such as an on-board processor (since it has been demonstrated that critical features can easily move to hardware).

A further area of exploration would entail the inclusion of hardware based CNNs for object classification and the necessary pre-processing/ feature extraction for object

detection; or the incorporation of a priori space knowledge that might result from the coupling of WiFi based localization data or manual panoramic scene stitching.

Bibliography

[1] E Bishop and Qingguo Li. Walking speed estimation using shank-mounted accelerometers. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 5096–5101. IEEE, 2010.

[2] Dimitrios Dakopoulos and Nikolaos G Bourbakis. Wearable obstacle avoidance electronic travel aids for blind: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(1):25–35, 2010.

[3] Sevgi Ertan, Clare Lee, Abigail Willets, Hong Tan, and Alex Pentland. A wearable haptic navigation guidance system. In *Second International Symposium on Wearable Computers, 1998. Digest of Papers.*, pages 164–165. IEEE, 1998.

[4] D Jeon, N Ickes, P Reina, I Ananthabhotla, HC Wang, DL Rus, and AP Chandrakasan. A navigation device with 3-d computer vision processor for visually impaired people. *Circuits and Systems for Information Processing, Communications, Multimedia, Energy Management, and Sensing*, page 13.

[5] Dongsuk Jeon, Nathan Ickes, Priyanka Raina, Hsueh-Cheng Wang, Daniela Rus, and Anantha P Chandrakasan. A 0.6 v, 8mw 3d vision processor for a navigation device for the visually impaired. 2016.

[6] Larry Li. Time-of-flight camera–an introduction. *Technical White Paper, May*, 2014.

[7] Koji Tsukada and Michiaki Yasumura. Activebelt: Belt-type wearable tactile display for directional navigation. In *UbiComp 2004: Ubiquitous Computing*, pages 384–399. Springer, 2004.

[8] F van der Heijden and PPL Regtien. Wearable navigation assistance-a tool for the blind. 2005.