2018

Leveraging Eye Structure and Motion to Build a Low-Power Wearable Gaze Tracking System

Addison Mayberry

LEVERAGING EYE STRUCTURE AND MOTION TO BUILD A LOW-POWER WEARABLE GAZE TRACKING SYSTEM

A Dissertation Presented

by

ADDISON MAYBERRY

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2018

College of Information and Computer Sciences

LEVERAGING EYE STRUCTURE AND MOTION TO BUILD A LOW-POWER WEARABLE GAZE TRACKING SYSTEM

A Dissertation Presented

by

ADDISON MAYBERRY

Approved as to style and content by:

____

Deepak Ganesan, Chair

____

Benjamin Marlin, Member

____

Marco Duarte, Member

____

Ivan Lee, Member

____

James Allan, Chair of the Faculty
College of Information and Computer Sciences

DEDICATION

This thesis is dedicated to the families who kept me going along the way.
My biological family: Mom, Dad, Tanner, Caelan, Kiera.
My undergrad family: Jesse, Steve H., Jantzi, Mike, Phil, Decoy, Ruba, Nate.
My roommate families: Steve O., Z. Do, Dan, Seth, Austin, Isabelle, Elanor.
My church family at MercyHouse, who are too many to list here but whom
I will never ever forget.
And above all, the Father of those families, from whom all blessings flow,
including this work and these relationships that helped bring it to bear.
I love you all.

ACKNOWLEDGMENTS

No researcher is an island and no dissertation is put together without the input of many people. First and foremost I want to thank my wonderful advisor, Deepak Ganesan, for all of his advice, mentorship, hard work, and patience with me over the years. It was truly a privilege to be his student, and any and all success I may have in my career I owe in part to his influence and tutelage. Christopher Salthouse was also a huge source of mentorship and guidance, and I am very grateful for the brief time that we were able to work together.

I also want to thank the many other collaborators I have had over the years. Many of them were formal co-authors, but also many were not. Among graduate students, I particularly want to acknowledge Pan Hu, Yamin Tun, and Soha Rostaminia; I also had the honor of mentoring many excellent undergraduate students, most notably Duncan Smith-Freedman, Jose LaSalle, and Connor Pope. You are all exceptionally talented and it was an honor to work side-by-side with all of you in the trenches of the iShadow project. I look forward to the great successes that the future holds for each one of you.

I also want to acknowledge the incredible influences and support I received outside of graduate school. It is becoming less and less of a secret that getting is a PhD is an incredibly trying experience for most and takes a toll on mental and emotional health. I have been blessed with an incredible array of people in my corner who kept me sane and loved me through the roughest patches.

I have an incredibly loving and thoughtful family who never failed to stay in touch and ask how I was doing, especially Tanner and Caelan who never failed to include me in whatever online game they were playing and were patient in the long periods when I couldn't participate or was too tired to be much fun when I did. I also had the support of

far too many friends to list here by name. I particularly want to thank and acknowledge my undergraduate fraternity brothers who have stayed in touch and supported me year after year, especially Jesse Porch who offered me advice and a place to escape on far too many occasions to remember.

The single most influential group in seeing me through to the end of this thesis, by far, is the community of MercyHouse Church. If I were to list all of the things they have done for me, the acknowledgements would run longer than the body of this dissertation. Thus, it will have to suffice to say that they are what made Amherst my first real home. Robert Krumrey has done an incredible work in and for God in establishing and leading this community, and the echoes of my time with them will resound throughout the rest of my life.

The people I shared a living situation with were, by and large, wonderful friends to me and a huge source of encouragement and daily support in the daily struggle to get up and keep fighting the PhD fight. Stephen Oloo and Elizabeth Do carried me through one of the worst periods of my adult life. Steve and I later had the privilege of being joined by Dan O'Shea and Seth Berggren, and I have been changed and matured by being in proximity to all three of them. Last but absolutely not least, the Kopack family - Austin, Isabelle, and little Elanor - welcomed me into their home during the final stretch of this work and gave me light in a period when the darkness threatened to completely close in. I can only hope that someday I will be a part of a family that can extend as much love to another as you three have to me.

There are so many others who were a part of my life and in their own ways, big and small, kept me afloat with their friendship. I will not even attempt to list out their names for fear of leaving anyone out (or being accused of trying to inflate the page count of this document), but you know who you are. I will forever love and remember the place and these people who are behind this dissertation.

ABSTRACT

LEVERAGING EYE STRUCTURE AND MOTION TO BUILD A LOW-POWER WEARABLE GAZE TRACKING SYSTEM

SEPTEMBER 2018

ADDISON MAYBERRY

B.Sc., GROVE CITY COLLEGE

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Deepak Ganesan

Clinical studies have shown that features of a person's eyes can function as an effective proxy for cognitive state and neurological function. Technological advances in recent decades have allowed us to deepen this understanding and discover that the actions of the eyes are in fact very tightly coupled to the operation of the brain. Researchers have used camera-based eye monitoring technology to exploit this connection and analyze mental state across across many different metrics of interest. These range from simple things like attention and scene processing, to impairments such as a fatigue or substance use, and even significant mental disorders such as Parkinson's, autism, and schizophrenia.

While there is a wealth of knowledge and social benefit to be gained from eye tracking, the field has historically been restricted to laboratory use by crippling technological limitations - most notably, device size and power consumption. These issues primarily stem from the use of high-resolution cameras and heavyweight video-processing algorithms, both of

which induce extremely high performance overhead on the eye tracker. To address this problem, we have constructed a lightweight, ultra-low-power eye monitoring device in the form factor of a pair of eyeglasses. The key guiding design principle for its construction was saliency-aware resource minimization. Specifically, our design leverages the fact that close-up images of the eye are characterized by large salient features which provide a high degree of redundant information; we exploit this to heavily subsample the eye image and reduce resource utilization while performing effective eye tracking.

In the first part of this thesis, we present an initial design of a wearable system to enable ubiquitous eye tracking. By exploiting the fact that the eye has several large, visually redundant features such as the iris and pupil, we were able to develop a neural-network-based adaptive-sampling algorithm for predicting the gaze point while sampling a minimal number of pixels from the image. This enabled us to realize a power savings using specialized imaging hardware that would sample only those most salient pixels, which proportionally reduced the power and time cost of reading images for eye tracking. With these optimizations we were able to build a first-of-of its kind wearable eye tracker that consumed 40 mW of power and demonstrated a gaze tracking error of only 3° across multiple subjects. We refer to this device as the iShadow platform.

The second contribution and section of this thesis is a significant improvement upon the original iShadow design for the purpose of improving both power utilization and eye tracking performance. We constructed a new pupil-tracking algorithm based on lightweight computer vision features, which leverages the smoothness of the eye's motion to reduce even further the amount of camera sampling needed. To guard against very infrequent discontinuities resulting from blinks or reflections off the eye, we integrated this model with the previously-used one-shot neural network algorithm. Because the common case (smooth, uninterrupted eye motion) occurs 90% of the time, we were able to realize a dramatic increase in performance due to the efficiency of the smooth tracking algorithm. The new and improved system, labeled CIDER, enabled much more accurate eye tracking

- 0.4° error - with power consumption as low as 7 mW. This design also enabled a tradeoff between power consumption and eye tracking rate, in which it was also possible to draw higher power of $\tilde{3}0$ mW in order to do eye tracking at rates of up to 240 frames per second.

The final contribution of this thesis is a re-designed version of the iShadow glasses hardware that is suitable for "in-the-wild" studies on subjects in their daily living environment. A wearable device, especially one that is worn on the head, must be minimally obtrusive in order to be accepted and used in the field by subjects. This design goal conflicts with the ideal placement of cameras that is needed for achieving consistent eye tracking fidelity. We present multiple possible methods we explored for addressing these competing design challenges, and discuss the reasons that many proved infeasible. To conclude, we present a working design solution that appears to optimally trade off user comfort and convenience and against the technical requirements of the system.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

The human eye offers a fascinating window into an individual's personality traits, medical problems, brain abnormalities, behavioral conditions, cognitive attention, and decision making. These characteristics have made it the subject of decades of research by experts in cognition, ophthalmology, neuroscience, epidemiology, behavior, and psychiatry, who have not only enhanced our understanding of how the eye works, but also revealed new ways of diagnosing health concerns. For example, nearly every health condition that affects the brain causes substantial variations in eye movement patterns including ADHD [32], Autism [66], Williams syndrome [67], Schizophrenia [14], Parkinsons [4, 11, 15, 69], Alzheimers disease [54], Depression [18], and others. The eye also reveals a great deal about our current cognitive state, thereby providing surprising benefits for even healthy individuals.

In the landmark book "Thinking Fast and Slow" [40], Nobel laureate Daniel Kahneman eloquently describes how an individual's System 2, which is our slow, deliberate, analytical and consciously effortful mode of reasoning, tires after too much cognitive effort, resulting in greater reliance on the unreliable but less effortful System 1, leading to poor decision making (also known as ego depletion). The effects are wide-ranging: judges are more likely to deny parole at the end of the day [25], clinicians have been found to prescribe unnecessary antibiotics [51], soldiers make poor decisions in operational environments [33], people buy more junk food [9], consume more alcohol and cigarettes [10, 21], and so on. How would we detect such cognitive "fatigue?" By looking at the eye and measuring pupil dilation.

In addition to this, continuous real-time tracking of the state of the eye (e.g. gaze direction, eye movements) in conjunction with the field of view of a user is profoundly important to understanding how humans perceive and interact with the physical world. Real-time tracking of the eye is valuable in a variety of scenarios where rapid actuation or intervention is essential, for example, detection of unsafe behaviors such as lack of attention on the road while driving, or leveraging visual context as a signal of user intent for context-aware advertising. Continuous eye tracking is also useful in non real-time applications such as market research to determine how customers interact with product, and advertising placement in stores.

The increasing prevalence of ubiquitous and wearable devices offers yet another avenue in which eye tracking could be extremely beneficial. As devices become more compact and integrated into everyday objects (appliances, articles of clothing, etc.), touch input is becoming less and less viable and other input modalities will be pushed to the forefront. Eye tracking is one such modality that could benefit many categories of such devices. The most obvious is head-mounted displays, especially virtual-reality and augmented-reality headsets. The goal of these systems is to augment or replace the user's perceptual field, and so the ability to passively understand and react to user attention within that field via eye tracking promises significant performance enhancements. Other wearable and ubiquitous devices could realize benefits from eye tracking as well — consider, for example, displays and devices that only activate when a person is looking at them, or UIs that can organically respond to user intent as expressed through eye movements, thus reducing the need for potentially cumbersome touch-based UIs.

1.1 Challenge: Limitations of Current Technology

Bringing the benefits of eye tracking to bear on these areas, however, requires that it be possible to perform it continuously across a wide variety of contexts and locations. One option is to incorporate the necessary cameras and computation for eye tracking into every device that may benefit from it — however, most ubiquitous technology such as wearables and smart devices do not have the resources to do this locally, nor does it always make sense to distribute the eye tracking load in this way.

In fact, current state-of-art eye tracking hardware consumes 1 W or more of power [91], which is orders of magnitude more than most ubiquitous devices can support. Even in cases where the resources are available, wall-mounted displays for example, all eye trackers require explicit calibration for each user. Thus, individuals would not be able to simply walk up and begin interacting unless they do a calibration process as well. And lastly, implementing systems to do eye tracking under arbitrary situational variables such as relative head position, environmental lighting, etc. has proved a serious engineering challenge that is yet to be fully overcome [62].

All of the problems holding back existing eye tracking technology point to the need for an eye tracking regime that is centralized to the person instead of distributed among the devices around them — i.e, a truly wearable eye tracker. By bringing eye tracking close to the eye, many of the technical challenges are simplified and the calibrated models for eye tracking can be carried locally with the person as they move. This improves the overall performance and robustness of eye tracking and enables it to happen in cases where it's not feasible to incorporate it in other devices or in the environment.

1.1.1 Current Wearable Eye Trackers

Wearable eye tracking technology already exists and is commercially available; however, it has serious limitations to its use. Existing wearable eye trackers are effectively high-definition camera banks with strong infrared illuminators, which is all wrapped together into a head-mounted frame. The aforementioned high power draw (1+ W) is needed to power these components. In addition, these devices can only record data, not do live processing, unless an external source of computation is attached. Thus, additional power is needed for some sort of local storage for the HD video streams. Altogether, these requirements mean that the devices are bulky and must be tethered at all times to a large battery pack + hard drive combination in order to function. While this might be acceptable for small studies or in-the-lab experiments, these properties are considered extremely undesirable by the average person, for whom even the comparatively subtle Google Glass seems overbearing [75].

This obviously hinders commercial applications, but it is also a major problem for researchers. There is a wealth of knowledge to be gained from performing longitudinal studies on subjects in daily life environments. Psychological and social scientists could gain unprecedented data about natural responses to external stimuli by being able to instrument a person's eyes over long periods of time. However, for this to work, the device would have to both be comfortable for extended use and not introduce behavioral bias by constantly drawing attention or interfering with the wearer's normal behavior. These requirements are not met by existing systems.

Thus, eye tracking technology will never jump the gap into full ubiquity and actualize the major benefits it can offer in both the commercial and academic spheres while it is burdened with these complications.

1.2 Problem Statement: Wearable Eye Tracking Design Goals

Having established that current wearable eye tracking systems are unsatisfactory, we provide here a list of design goals that should be met in order to create an ideally usable wearable eye tracking system. We break out the goals into two broad categories: quantitative measures of technical performance, and qualitative statements on usability. The reason for the latter, as stated above, is that subjects must be willing to use the glasses for long periods or else they will not be viable for their primary purpose.

Quantitative goals:

1. Operates at power levels comparable to contemporary practical wearable devices: i.e., less than 10 mW

2. Able to measure most categories of salient eye features accurately in real-time – blinks, eye closure amount, saccades (fast eye movements), fixations, gaze target – at a similar fidelity to commercial eye tracking devices

3. Performance at the above eye tracking tasks without significant degradation due to situational factors (e.g., outdoors vs indoors)

4. Can perform all of its primary functions without relying on an external connection for additional power or data processing

Qualitative goals:

1. Can be worn for long periods without physical discomfort

2. Will not inflict mental burden on the user — i.e., does not require regular attention to operate

3. Does not obstruct the user's vision or in any way impair their ability to do their daily tasks

4. Appears "normal" to a casual observer and will not draw attention from other people or become an eyesore to the wearer

1.3 Thesis Contributions and Outline

Having established the broad utility of eye tracking technology, as well as the gap between current systems and the ideal, we ask: are the challenges faced by existing eye tracking implementations fundamental and unavoidable, or are they artifacts of specific design decisions? In this thesis, we demonstrate that the latter is true by designing and implementing a lightweight, ultra-low-power wearable eye tracking system.

Our formal thesis contribution is the design and implementation of a wearable eye tracking device that implements the design goals specified above. We here break out the specific contribution elements by chapter.

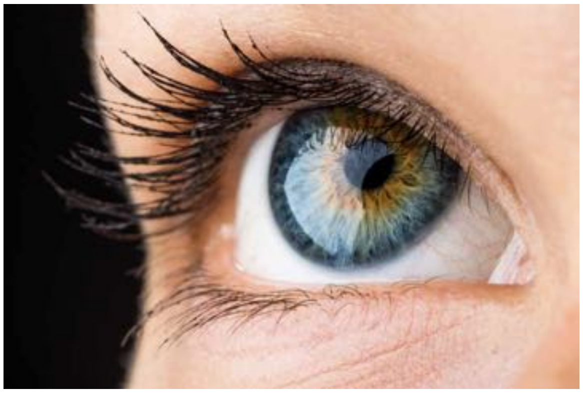


Figure 1.1: Idealized image of the eye

1.3.1 iShadow Gaze Tracking Platform

The first contribution of this thesis is given in chapter 3, in which we present a first pass at the design and implementation of a system for meeting the design goals specified above. The fundamental insight that makes all of this work possible is that an image of the eye, when taken from a close distance, is extremely visually redundant. This is very apparent when looking at an example, such as the one given in figure 1.1. This is an artificially colored image, but it makes the eye regions very easy to see. The eye is composed of several large contiguous regions, mainly the sclera (white of the eye), iris, and pupil. Since these regions are so well-structured, it seems very intuitive that it is possible to undersample them within the image and still be able to estimate their orientation and appearance accurately. You also can observe that the eye itself only takes up a portion of the image, with the rest being uninformative features such as the facial region — this data can be discarded entirely.

Thus, it should be possible to estimate eye parameters using only a small subset of the pixels in the image. This is an advantageous approach because there is an inherent cost to sampling and processing each pixel's worth of data, and in a low-power embedded system all resources are extremely scarce. Leveraging this key insight, we were able to design a system to take advantage of the visual redundancy within the most salient portions of eye images by intelligently subsampling those images in hardware, and doing efficient analysis of the resulting sensed data. The concrete implementation was a shallow neural network model trained to do gaze tracking from images of the eye. The model uses input regularization to determine which specific pixels of the eye image are most informative for the gaze tracking task, and only these pixels are sampled in the image sensor hardware.

This was a specific instantiation of the more general systems concepts of adaptive sampling and cross-pipeline optimization. This adaptive sampling strategy, built to leverage the specific domain opportunities afforded in the eye tracking problem, enabled us to realize a dramatic power and processing time savings over a naive approach. With these optimizations we were able to build a first-of-of its kind wearable eye tracker that consumed 40 mW of power and demonstrated a gaze tracking error of only 3° across multiple subjects. We refer to this device as the iShadow platform.

1.3.2 CIDER: Leveraging the Motion of the Eye

In chapter 4 we present the second contribution of this thesis, which is a significant improvement upon the original iShadow design for the purpose of improving both power utilization and eye tracking performance. The first iShadow design represented a crucial first step towards realizing our design goals, however, there was a still performance gap in key areas. Specifically, we needed to further decrease power draw to reach the level of actual commercial devices, and also to increase gaze / pupil tracking accuracy so as to match state-of-art eye tracking headsets. With this in mind, we used and improved upon the same general approach of application-driven resource allocation.

The major insight that enabled us to build a dramatically improved version of the platform was to leverage the smoothness of the eye's motion. In our previous work, we had developed a one-shot algorithm that accepted as input a single image of the eye with no context and extracted the gaze angle. This worked by leveraging the strong physical structure present in the eye, as discussed above. However, there is also a great deal of temporal structure to be leveraged — that is to say, there is a reasonable limit to how far the eye can move between image frames. Thus, if we are able to record fast enough, its motion can be modeled as smooth and we can make assumptions about its current position based on recent frames. There are discontinuities that can occur, most notably blinks or reflections off the eye that may appear to change its shape momentarily, but in the common case (90% of the time) the eye can be assumed to move smoothly.

We developed a new iteration of the iShadow platform, called CIDER, that leverages this common case in order to do even more efficient sampling while guarding against the aforementioned discontinuities. By taking into account the past position of the eye and doing very targeted pixel sampling in the common case, and then using the ANN model to bootstrap in the case of discontinuities, we were able to achieve even more accurate eye tracking (0.4° error) with power consumption as low as 7mW. This design also enabled a tradeoff between power consumption and eye tracking rate, in which it was also possible to draw higher power of $\tilde{3}$0 mW in order to do eye tracking at rates of up to 240 frames per second.

1.3.3 Engineering Considerations for Practical Applications

Chapter 5 presents the final contribution of this thesis, a re-designed version of the iShadow glasses hardware that is suitable for "in-the-wild" studies on subjects in their daily living environment. There is a non-obvious but inherent conflict among the design goals discussed above, namely, that a subtle and unobtrusive design of the device means placing the cameras at a position in which it is far more difficult to do consistent eye tracking at high fidelity. We present multiple possible methods we explored for addressing these competing design challenges, and discuss the reasons that many proved infeasible.

As contributions, we first present one working design solution that appears to optimally trade off user comfort and convenience and against the technical requirements of the system, thus fully realizing the wearable eye tracker design goals we have laid out here. It has two slightly different implementations with varying properties, with preference likely decided by the needs of the specific application. We present a comparison of eye tracking results between this design and the CIDER iteration of the platform. To conclude, we discuss a design for an even more advanced version of the system that uses flexible PCB material to integrate the cameras completely with the frame.

CHAPTER 2

BACKGROUND AND RELATED WORK

The problem of building a device to automatically observe and record features of the eye (especially the gaze angle) has been in active exploration for nearly a century [88]. Since the onset of the ability to record video and the computational means by which to process it, the field has taken enormous leaps forward. To provide a full context of even the most recent advances across the entire field of eye tracking is well outside the scope of this document.

Instead, we attempt in this section to provide background and context for this thesis as follows: we first give a summary of the major sensing modalities used for eye tracking, and then focus in on video-oculography, which can be divided into major classes of techniques depending on the type of hardware and algorithm. We give a summary of these classes and then do a more robust literature comparison on the class into which the product of this thesis falls: real-time wearable eye trackers.

We refer the interested reader to the following excellent resources for exploring the field more broadly:

- Early history of the eye tracking field, including early efforts at video-based methods [88]

- Textbook encompassing eye physiology and related neurological / psychological mechanisms, summary of eye tracking techniques, and guidelines for experimental design [28]

- More recent state-of-art summary of the eye tracking field, slightly dated but still an excellent resource for understanding the space [34]

- For the most recent work in eye tracking, the ACM Symposium on Eye Tracking Research and Applications (ETRA) [2]

2.1 Sensing Mechanisms for Eye Tracking

The gaze tracking problem has a long history and multiple sensing modalities have been developed to address it [88, 28]. We briefly address the different types of sensors that are currently in active use.

2.1.1 Video-oculography.

By far the most common technique at this point in time is the use of camera technology to record a video of the eye(s) and then apply video processing techniques to extract relevant information about the state of the eye on a frame-by-frame basis. This technique is referred to as "video-oculography" or V-OG, and the work of this thesis falls entirely under its umbrella since we exclusively make use of camera capture in order to sense the eye. V-OG has several major advantages: camera technology and video processing are well-understood and relatively easy to implement, there is little to no concern about noise in the data for most use cases, and the data is intuitively understandable to a human observer. In addition, though it is not the goal of this work, it is possible to do V-OG in a non-contact way by placing cameras in the environment — this is not possible with any other sensing modalities, which require the sensors to be in close proximity to the eyes.

The biggest disadvantages of V-OG techniques are data collection speed and processing time. This is simply due to the volume of data that is collected and must be analyzed. Other sensing modalities generate relatively simple time series data, which can be collected at a very high rate and potentially with a lower power draw. In addition, time series data, while often difficult to interpret, can be extremely fast to process due to the smaller dataset sizes. Contrast this with video data, in which a single frame could feasibly comprise multiple megabytes of data and usually cannot be collected at a rate higher than 60 Hz.

2.1.2 Electro-oculography.

An older technique that is possibly seeing a resurgence is the use of electrodes placed on the face to observe eye movements. This technique, known as electrooculography or EOG, works because the eyeball is actually an electric dipole with a positive charge on the front (cornea) and a negative charge on the back. Thus, changes in the position of the eye will register as changes in the electrical signal between pairs of electrodes placed on opposite sides of the eye regions [17]. The major drawbacks of EOG stem from the sensitivity of these electrodes — as with all forms of electrogram, achieving a good signal requires using "wet" electrodes with a liquid solution that ensures good electrical contact, and they must be placed at very specific locations on the body. These conditions must necessarily be relaxed for anything but the most physically restrictive in-the-lab studies, but relaxing them by using "dry" electrodes and allowing for variations in contact position make the signal very difficult to use and interpret [5, 62].

That said, there have been several recent efforts to use EOG techniques for wearable eye tracking in particular. The sensors are fairly small, inexpensive and low-power, and the data collected is fairly low-volume. These are all very desirable traits for an on-body sensor. Several groups have explored the space with custom-built EOG devices [17, 52], and in recent years a consumer product featuring EOG has been developed, the JINS MEME [39], which some groups have begun exploring for wearable eye tracking research [62, 27].

2.1.3 Photosensor-oculography.

More recently, a new branch of oculography has begun to emerge which uses simpler optical sensors rather than full cameras. Some works have demonstrated improved performance by using simple light sensors such as photodiodes fused with other, more feature-rich sensing modalities [38, 61, 90], and Li. et. al. explored it as the only sensing method [49]. Others have explored the use of sensors that track the surface of the eyeball. For example, Borsato and Morimoto have explored using an optical mouse sensors, which tracks changes of the target surface, to identify motions of the eye [16]. In industry, some companies have begun to explore the possibility of using MEMS distance sensors to map the surface of the eyeball and thus track its movements at high rate [65, 3].

2.2 Techniques for Implementing Video-Oculography

In this work, we are specifically interested in V-OG. We highlight the distinctions between video-based gaze tracking systems that are most relevant to our work in this section. This is the most active area of eye tracking research at the moment, and thus we can only give a brief overview of the major types of methods with sparse references — we again refer to the references at the beginning of this chapter for more details. We organize this overview around three important distinctions between gaze tracking systems, which are largely based on the categories described in [34]: (1) whether the hardware system is remote or wearable (typically head mounted), (2) whether the point-of-gaze estimation problem is solved offline or in real time, and (3) the category of algorithm used to solve the gaze inference problem.

2.2.1 Camera Type and Placement: Remote vs Wearable

The first major distinction is around the positioning of the cameras and light sources (if any) relative to the head and eyes. The two options are “remote” tracking, in which the cameras are placed in a static position in the environmeant, and wearable / head-mounted devices. Each has its advantages and disadvantages, and the choice is largely made by virtue of the application needs. In addition, the types of algorithms used depend in large part on the hardware configuration used — in many cases an algorithm is only suitable to be used in one situation or the other, in other cases the same algorithm can be used on data from either with some modifications.

2.2.1.1 Remote eye tracking.

Gaze tracking has traditionally been studied and applied in the remote tracking setting [88]. The subject sits facing one or more cameras that record video of the subjects eyes. The subject typically performs a task like reading or viewing images or video. The captured eye images are then used to infer the subject's point of gaze, giving information about what they were attending to while performing the different steps of a task. In early remote systems, chin-rests or other restraints were used to ensure the subject's head was completely motionless relative to the camera system. The gaze inference problem was also solved completely offline.

Subsequent advances have led to gaze tracking systems that can reliably infer gaze direction in real time while allowing subjects a much more comfortable range of motion while seated [20, 1, 68]. Modern instantiations of these techniques use either built-in cameras on laptops and smartphones to estimate gaze direction, or use additional add-on hardware [78] that provides more accuracy for gaze tracking. These approaches are particularly useful for measuring user engagement for online ads, and gaze-based gaming or computer interaction. While very useful, the gaze tracking hardware remains static, and doesn't provide continuous gaze information in natural environments when the user is not interacting with a computing device. In addition, there are complications arising from head pose and lighting conditions, which must be accounted for in the gaze tracking algorithm and tend to result in performance degradation outside of a controlled lab environment [62].

Remote tracking is well-suited for psychological studies involving analysis of the rapid eye movements known as *saccades*. It has been demonstrated that the exact behavior of saccades changes depending on many neurological factors, to include fatigue, neurological conditions such as ADHD, and even neurodegenerative diseases such as Alzheimer's and Parkinson's diseases [54, 15, 69]. However, saccades occur over very short time periods from 10 to 100 ms, and thus require a high-framerate camera in order to correctly measure [28]. Such camera hardware is generally large and consumes high power, and thus is only suitable for a remote tracking setup. This, in addition to the fact that head movements make saccades more difficult to accurately quantify, means that these types of studies can only be done with remote tracking.

2.2.1.2 Wearable eye tracking.

In recent years, more eye trackers are being used in a mobile context. Wearable eye trackers are almost always mounted on the head in the form factor of a pair of eyeglasses. The major advantages of wearable eye tracking over remote are (a) that the video stream is agnostic to head pose, as the cameras are always pointed directly at the eyes, and (b) the system can travel with the user, potentially allowing for eye tracking to occur even when the user is not seated in front of a device such as a laptop.

These same benefits, however, are hindered by the technical requirements of an eye tracking device. Typical eye tracking algorithms require high-resolution video data to function well, which has two problematic implications already discussed in chapter 1. The first is that the cameras require a large amount of power, and therefore most wearable systems require a battery pack of some sort. The second is that the high volume of data mandates a powerful processing unit to handle the data, so it must either be stored locally for later processing or connected to a laptop or powerful smartphone in order to do real-time eye tracking.

For many purposes, especially research studies, it is sufficient to record eye tracking data for later analysis. However, even in this scenario, we note that it is extremely beneficial to be able to do local processing in order to reduce the power and data storage needs of the device — even industrial devices, which function primarily in this mode, require carrying a battery pack + storage device that only lasts for 2 - 4 hours. For many practical applications and even some research purposes, however, it is strictly required to be able to compute gaze data in real-time, and systems with this goal are beginning to emerge. Since this is the primary category into which this work falls, we give a more detailed breakdown of the current state-of-art in section 2.3 below.

2.2.1.3 Near-Infrared vs Visible Light

One small note to discuss on the hardware side is the type of illumination used - the majority of eye trackers use Near-Infrared (NIR) illumination of the eye to make the pupil appear darker or lighter [56]. The advantage of NIR illumination is that it creates specific reflection patterns on the cornea and pupil of the eye, and these reflections can be captured by one or two imagers. The resulting images can be processed through advanced image processing techniques to robustly extract various aspects of the state of the eye. The disadvantage of this approach is that the device is considerably more complex, and requires an illuminator and reflector to direct the NIR light into the eye, in addition to one or more imagers. This makes the eyeglass design a much more complex engineering problem, with a variety of active and passive elements mounted at different points on the frame. While visible-light-based eye trackers have been proposed in the past [47], these are far less advanced than the NIR-based counterparts as they are generally much less precise and less robust to changes in lighting conditions.

2.2.2 Shape-based vs Appearance-based Gaze Estimation

In terms of gaze tracking algorithms, three primary approaches have been explored in the literature. They are commonly referred to as shape-based, appearance-based and hybrid algorithms [34]. The shape-based approach uses features of the eye image to fit an ellipse to the boundary between the pupil and the iris [34]. A key requirement of this approach is the use of NIR illumination sources, which, through their positioning relative to the camera, can make the pupil appear brighter or darker, thereby making it easier to detect the boundary [56]. When using visible light, shape-based techniques are harder to use since the boundary of the pupil is harder to detect. Shape-based approaches are extremely popular as they tend to have very high accuracy if the eye image can be fit correctly to the underlying eye model – thus, much of the focus of eye tracking development has been on (a) developing increasingly advanced models of the eye to fit and (b) controlling the sensing environment (i.e., the face region visible to the camera) as tightly as possible so as to facilitate the best fit to the model.

Appearance-based eye tracking algorithms attempt to predict eye features directly from the pixels of the eye image without an intermediate geometric representation of the pupil. It has generally been used only for the purpose of gaze tracking, which refers to the specific task of mapping the wearer's eye orientation to a gaze angle in the global reference. This class of eye tracking essentially treats the gaze inference problem as a regression problem where the inputs are the pixels of the eye image and the outputs are the vertical and horizontal components of the point of gaze in the outward facing image plane. Due to the generality of the gaze inference problem when formulated in this way, predictions can be based on essentially any multivariate regression approach. Two prominent approaches used in the gaze tracking literature are multi-layer neural networks [8] and manifold-based regression [73, 24, 23]. This approach is preferable in our scenario, since it is more robust to artifacts observed in a visible-light based image stream of the eye. While we leverage an appearance-based technique, our primary contribution at the algorithmic level is the development of sparse appearance-based models for gaze prediction that optimize gaze estimation accuracy while minimizing the pixel sampling costs.

2.3 Wearable Real-Time Eye Tracking Devices

We here briefly discuss other wearable eye tracking devices currently in existence. Note that, to the best of our knowledge, the system discussed in this work is the only one that is design to be able to do real-time eye tracking entirely on-board. All others are either intended to record video data for later processing, or must be plugged into a laptop or smaller device such as a smartphone or Raspberry Pi to do live tracking.

2.3.1 Commercial eye tracking devices

There has also been mounting commercial interest in these kinds of mobile eye trackers and the prospect of being to track eye parameters “in the wild.” Companies such as Tobii and SMI have produced devices which have shown great promise for opening up new avenues of research [55, 87]. However, existing industrial-grade mobile eye tracking devices are predominantly a condensed version of a standard remote tracking system, including carefully calibrated on-board illumination and multiple high-definition cameras.

While the engineering required to condense such a complex system to a wearable form-factor (usually in the shape of a standard pair of eyeglasses) is impressive, the power requirements of such systems are inordinately large by wearables standards - the user is required to carry a large battery pack, and even then their average run time is less than four hours. In addition, many of these devices are not truly real-time — many of them only perform video recording and storage, and even in more recent iterations can only do real-time tracking when connected directly to a laptop or smartphone. In these cases, performance will degrade, since processing the eye data often means running computations over thousands or hundreds of thousands of high resolution simultaneous image streams. Thus, a full desktop machine is preferred, and the processing can be a time-intensive task. Finally, the cost of these devices is often in the tens of thousands of dollars, making any kind of large-scale study infeasible to all but the most well-funded organizations.

The major exception to many of these caveats is Pupil Labs [59], which originally began as an academic project (discussed in more detail below). They currently produce a device featuring higher framerate than most others, at a pricing that is intended to be accessible to academic researchers. This device does still require the presence of a laptop.

2.3.2 Academic projects

2.3.2.1 COTS-based, laptop-connected devices

There have been a number of academic projects with the aim of developing a general-purpose eye tracking device. Many aim to be easy and cheap to build based on the idea of "open hardware," generally using off-the-shelf webcams and as a consequence being fairly bulky and always requiring a laptop connection [47, 42, 43, 7, 64, 60]. There are exceptions to these generalizations: [43] attempts to mitigate the bulkiness by attaching the cameras to a baseball cap, and [60] facilitates data transmission to a base station via wifi. These projects vary as to whether they provide a full algorithm for eye tracking as well as the hardware, some opting to use existing or simplified algorithms with the emphasis being on the hardware design.

2.3.2.2 iGaze

A particularly notable device of relevance to our work is iGaze [91]. The goal of iGaze is to detect gaze fixations and determine whether the user is looking at a networking-capable device, and if so, the user can initiate a wireless connection to facilitate some useful exchange of data between the iGaze platform and the device in question. The platform itself is comprised of a head-mounted camera monitoring the eye, very similar to this work, and a Raspberry Pi device carried by the user to do image processing and gaze computation. iGaze's predictive error and power consumption are relatively high compared to our work - they report average gaze error of 5° and average power consumption in excess of 1 W, whereas our error is below 1° and power consumption is 7mW.

2.3.2.3 iLid

iLid is an offshoot of this work which aims to estimate fatigue via the rate of eye blinks [62]. Using a similar platform to the one described in this thesis, the authors demonstrated the ability to accurately measure blink rate across a wide spectrum of situations known to be challenging for eye tracking devices — outdoor lighting conditions, physical activity (jogging), etc.

2.3.2.4 Pupil

As noted above, perhaps the most successful project intended to be a general-purpose eye tracking device is Pupil, which become the flagship product of Pupil Labs [41]. This project eschewed the use of COTS components, instead opting for custom-manufactured devices which were still cost-effective compared to those offered with other commercial wearable eye trackers. They developed a custom algorithm which was an extension of existing well-known techniques, but which still requires the presence of a laptop computer to run. This device has seen acceptance by other research groups [44].

2.3.2.5 Miniature-Camera Devices with Neural Network Algorithms

Following the publication of our initial work, at least two projects have presented devices similar in nature with regards to the camera hardware and tracking algorithm. ETracker [46] features a single miniaturized camera and uses a convolutional neural network (CNN) model for eye tracking, which is fairly common among recent works and computer vision problems in general.

The second relevant project is InvisibleEye [81], which explores the use of a bank of miniature cameras placed around the glasses frame and uses the data from all them in a CNN model for eye tracking. This bears some similarity to our final multiple-camera design as discussed in chapter 5. However, this device also requires a laptop attachment due to the complexity of running a full CNN, and therefore also does not have any considerations around power consumption — they use a bank of six cameras (three per eye) without subsampling pixels in the hardware, which is a key resource optimization in our platform that makes local gaze tracking feasible. They do, however, demonstrate that by doing software subsampling down to lower resolutions in all cameras and using that input to the CNN, that it is possible to achieve reasonable error rates. This presents the possibility of an interesting synergy with our platform, as discussed in future work in chapter 6.

2.3.3 Eye Tracking for VR

There is a closely related effort in the eye tracking field to wearable eye tracking, which is eye tracking for virtual reality devices. Many of the same constraints exist as in the general wearable eye tracking problem: the cameras must be head-mounted and therefore have limitations on size and functionality. There are two major distinctions with the VR gaze tracking problem, however. The first is that the restrictions on wearability are seriously relaxed — a VR device is a much larger and bulkier system than the glasses-style form factor that most wearable eye trackers aim for. Thus, it is possible to use larger cameras with more optimal positioning than in normal wearable eye trackers. Also, because the processing power required to do VR graphics rendering is extremely high (compared to that available to most wearable devices), it is easier to implement complex algorithms with good runtime.

This last point is doubly important because of the major design goal inherent to VR — the latency and accuracy requirements of the gaze tracking system are extremely high. Regardless of the exact task, it is critically important that the VR system be responding to the user's eye movements with what feels to the user like perfect accuracy and real-time latency. Failure to do so will be extremely jarring, possibly to the point of becoming disorienting or even sickening. Thus, it is critical that there be a source of major computation power available to ensure that these requirements are met.

There are a number of benefits to exploring eye tracking in conjunction with VR displays. One of the biggest is foveated rendering, which leverages the fact that human eyes have very high spatial resolution near the center of the visual field (the fovea), and lower spatial resolution outside of it. Thus, it is possible to save rendering effort by rendering with high detail around the user's gaze point and at lower detail around the edges of the field of view where it won't be noticed [58]. There is also the possibility of using the gaze point as explicit input to whatever programs are running on the VR system, so that it could be used as menu input or other forms of interaction with the system.

Systems to implement eye tracking for VR have been under exploration in some form for about two decades [29, 45]. Due to the high technical requirements, much of the work that has been done on eye tracking for VR has used commercial systems with proven quality, as opposed to systems that are under active research development. SensoMotoric Instruments (SMI) presented an early eye tracking add-on for VR systems that has been used in studies [58]. There is some work to develop systems on the academic side, Li et. al. have developed a photodiode-based insert for doing gaze tracking in sync with a VR device [49]. Meanwhile, many of the major vendors of VR devices have been exploring the integration of eye tracking by buying up eye tracking startups [83, 74], while additional eye tracking vendors have begun developing VR add-ons [13, 79]. Hardware startup Fove is proposing a completely integrated foveated-rendering-based VR platform, with eye tracking fully integrated into the hardware [82].

2.4 Miscellaneous Related Areas

We here briefly describe other fields that are related to the systems-level techniques explored in this thesis, and are not specifically limited to eye tracking.

2.4.1 Dynamic energy-aware adaptation

At a high level, our work can also be viewed as an instance of runtime energy-aware adaptation. Many techniques can be used to achieve such adaptation, including techniques such as varying application fidelity to achieve desired battery lifetime (e.g. changing the fidelity of a speech recognizer or video playback on a mobile device) [30] and adaptive sampling and communication that leverages spatial and temporal structure in sensor signals [86, 26, 48], among others. Our work is a very specialized instance of such adaptation in the context of eye trackers, and proposes the use of two models that are optimized to extract eye parameters at different costs (cross model and neural network, discussed below), and techniques for switching between the models based on observed dynamics.

CHAPTER 3

ISHADOW GAZE TRACKING PLATFORM

The first major contribution of this thesis is the design and implementation of an eye tracker that dramatically reduces the sensing and computation needs for eye tracking, thereby achieving orders of magnitude reductions in power consumption and form-factor over existing commercial eye tracking devices. We refer to this platform as iShadow. In this first chapter we focus specifically on the task of gaze tracking — identifying the direction of gaze based on the angle of the eye — and we will expand to more general features of eye tracking in the following chapter.

The key idea in this first contribution is that eye images are extremely redundant, therefore we can estimate gaze by using a small subset of carefully chosen pixels per frame. We instantiate this idea in a prototype hardware platform equipped with a low-power image sensor that provides random access to pixel values, a low-power ARM Cortex M3 microcontroller, and a bluetooth radio to communicate with a mobile phone. The sparse pixel-based gaze estimation algorithm is a multi-layer neural network learned using a sparsity-inducing regularization function that minimizes the gaze prediction error while simultaneously minimizing the number of pixels used. Our results show that we can operate at roughly 70mW of power, while continuously estimating eye gaze at the rate of 30 Hz with errors of roughly 3 degrees.

3.1 Background and Motivation

While our understanding of the human eye and gaze has grown through decades of research on the topic [28, 36], eye tracking remains limited to controlled user studies and clinical trials, and has not bridged the gap to daily consumer use. The central challenge is that the sensing and processing pipeline is extremely complex: the eye-facing imager alone requires continuous operation of a camera at tens of frames per second, and compute-intensive image processing for each frame [6, 47, 80]. Unfortunately, these computational demands are very far from what can be accomplished on low-power microcontrollers and resource-limited embedded platforms. In addition, there are complex camera placement considerations and form-factor demands, making the eyeglass design much more intricate. In all, addressing the myriad technical and design challenges of wearable gaze tracking is a daunting task.

3.1.1 Problems with existing solutions

To understand the design challenges, consider an eye tracker equipped with two cameras, one facing the eye and one facing the external world. A VGA-resolution eye facing imager sampled at 30Hz generates a data rate of roughly 4 Mbps. Continuous real-time processing of such an image stream would require computational capability and memory comparable to a high-end smartphone, making the eye tracker both bulky and power hungry. An alternative design might be to wirelessly stream data from the eye tracker to a smartphone for leveraging the phone or cloud-based computational resources. However, the bandwidth requirements for such streaming is substantial — most low-power radios cannot support the demanding data rates of eye trackers, and streaming via WiFi is power-hungry and would greatly limit the lifetime of such a device. Perhaps as a result, many state-of-art eye trackers, such as the Tobii glass [80], operate as data recorders and continuously writes data to a disk that the subject carries in their pocket. (Google Glass, while not equipped with an eye tracker, has similar challenges - in continuous video capture mode, the device lasts for only a few hours.)

We argue that these problems are a consequence of the fact that the traditional approach to wearable eye tracker design is fundamentally flawed – existing systems separate image acquisition from the eye state processing. Consider, for example, recently available smartphones such as the Samsung Galaxy S line, which track gaze for eye scrolling; here, the entire image is acquired from the camera, after which it is processed through computer vision techniques to estimate gaze direction. By dividing the eye tracking pipeline into discrete processing stages based on a typical computer vision system (e.g., shuttered camera sensor, generic camera driver, camera API, vision processing software), this design prohibits any possibility of larger cross-stage optimizations that could dramatically improve eye tracking performance. It could be claimed that in the smartphone case this is simply a consequence of modular software / hardware design, however, even completely custom, state-of-art commercial and research eye tracking devices operate on a similar regime [91, 84].

3.1.2 Key design insight

The fundamental concept of this work is that employing a more holistic approach to eye tracker design enables system-wide optimizations that can cut across the divide between high-level system components (e.g., camera hardware, image processing). Specifically, we use application-driven sampling and processing of the eye image to reduce power consumption in the embedded hardware without compromising the effectiveness of the eye tracking task. The potential performance improvements to be gained by allowing more direct software control of camera hardware have been explored in the past [50]. We believe, though, that there are features specific to the wearable eye tracking problem that allow for even greater optimizations for this task in particular.

The specific property of the eye tracking task that we leverage for this work is that individual images of the eye, when taken from a close distance, are extremely visually redundant and thus it should be possible to estimate eye parameters using only a small subset of the image. This technique is generally known as adaptive sampling. Most eye tracking algorithms leverage this information explicitly, which cements this intuition [34].

If we can identify these salient regions of the image for the purpose of tracking the eye's orientation, it follows intuitively that we can allocate hardware resources more intelligently. As discussed previously, the majority of an eye tracking system's resource cost is a result of the large amount of images that must be acquired, digitized, and stored. By leveraging knowledge about which regions of the image field are most useful for gaze tracking, we can focus our limited resource budget towards capturing and processing only these regions. This is facilitated at the hardware level by using a custom image sensor which allows specific pixels from the camera to be sampled and others to be ignored. By doing this, the system can sample only pixels corresponding to the most salient regions of the eye image, thus not incurring any sampling and processing cost for the less informative pixels.

In summary, our goal is to identify the most useful pixels in the imager for the purpose of eye tracking and then only sample and process those. The twin benefits of this are: (a) a savings in resource utilization (mainly power consumption) that is directly related to the proportion of the image that is not sampled, and (b) minimal cost to the performance of the eye tracking algorithm, since the pixels that are left unused are those that have been identified as less relevant for the eye tracking task.

3.1.3 Design Considerations

This targeted sparse sampling of pixels has a ripple effect on almost all design choices in our system. From a resource perspective, fewer pixels per frame imply lower memory needs, and fewer image processing instructions per frame thereby enabling the entire gaze estimation pipeline to execute on a simple microcontroller. From a latency perspective, fewer pixels implies lower latency for image acquisition and gaze estimation, making it possible to achieve real-time high rate gaze computation despite limited processing capability. From a power perspective, we subsample pixels directly at the imager, and not after acquiring the image, thereby reducing power consumption for image acquisition as well as processing. Fewer pixel acquisitions and less processing translates to less energy consumed per frame capture and gaze computation, which enables longer term operation.

The design of a sparse acquisition-based gaze tracking system presents substantial challenges that we address in this work. First, imagers that operate at the milliwatt power levels need to sacrifice pixel quality for power, hence an important question is whether we can achieve high accuracy despite operating with low quality imagers. Second, we ask whether we can design a gaze estimation algorithm that provides a graceful resource-accuracy tradeoff, where the accuracy of gaze estimation gracefully degrades as the energy budget reduces. Third, the computational demands of gaze tracking are often substantial — for example, several gaze estimation algorithms require processing of the eye image to detect the iris and identify its boundary, which requires processing that is considerably higher than what can be accomplished with a microcontroller. Thus, we need to understand how to reduce the processing needs to enable real-time gaze estimation while operating on platforms with tens of kilobytes of memory and no floating point units.

3.1.4 Contributions

The main contributions of this first chapter are the following:

- First, we design a multi-layer neural network-based point-of-gaze predictor that uses offline model training to learn a sparse pixel-based gaze estimation algorithm. A key novelty of this work is the use of a state-of-the-art sparsity-inducing regularization function for identifying pixel subsets[89]. We show such an approach can reduce pixel acquisition by 10× with minimal impact on gaze prediction accuracy.

- Second, we design a real-time gaze estimation algorithm that implements the model learned by the neural network on a prototype computational eyeglass platform equipped with a low-power microcontroller and low-power greyscale cameras with random access capability. We show that our system can operate at frame rates of up to 30Hz with gaze errors of roughly 3 degrees, while executing in real time and at a power consumption of 72 mW.

- Third, we show that our methods can be easily calibrated to a new individual with a calibration dataset that is only one minute long. Our user study with ten users shows that, once calibrated, our system works robustly across individuals.

3.2 iShadow Overview

In this section, we provide a brief overview of the iShadow system. The first step in using iShadow is calibration, where a user looks at a few points on a monitor while keeping their head relatively steady, in a manner similar to commercial eye trackers. During this calibration phase, iShadow captures a full image stream from the eye-facing and outward-facing imager, and downloads this data to a computer either via USB or Bluetooth.

The second step is the neural network based sparse pixel selection algorithm. In this stage, the learner divides the calibration dataset into training and testing sets, and sweeps through the regularization parameters to learn a set of models that correspond to different gaze prediction accuracies. Each model specifies both the set of pixels that need to be acquired and the weights on the pixels to use for the activation function that predicts gaze coordinates. Depending on the power constraints of the platform and the accuracy needs of the application, the appropriate model can be downloaded to the iShadow platform for real-time operation.

The third step is the run-time execution of the model that is downloaded onto the iShadow platform. The run-time system acquires the appropriate pixel set and executes the non-linear weighted sum to predict gaze coordinates in real time.

Once gaze coordinates are obtained from the eye-facing imager, they can be used in different ways depending on application needs. For example, it could be used to detect rapid saccades (i.e. rapid eye movements) which correspond to an event in the external field of view of the user. When a saccade is detected, the outward facing imager can be triggered to capture an image or video of the event that could have caused the saccade. Alternately, gaze coordinates could be used to detect fixation and use this information to decide when to trigger the outward facing camera.

3.3 Gaze Tracking Algorithm

The first major component needed is the algorithm for gaze tracking, which has two requirements as discussed previously: (a) be able to accurately determine the gaze point of the eye — specifically, the gaze point within the frame of reference of a camera which covers the wearer's field of view, and (b) identify and use only the most salient regions of the eye image for this task. In this section we describe our framework accomplishing these goals. At a high level, the idea involves setting up the gaze prediction problem as a neural network where the inputs are the pixel values obtained from the imager, and the output is the predicted gaze coordinates.

To enable subset pixel selection, the neural network learning algorithm uses a regularizer that penalizes models that select more pixels; thus, the optimization involves two terms: a) an error term that captures how well the algorithm predicts gaze coordinates, and b) a penalty term that increases with the number of pixels selected. This optimization is done offline using numerical optimization methods, and the parameters are hard-coded into the microcontroller for real-time execution. Thus, the eyeglass is not making any real-time decision about which pixels to sample, or how to map from pixel values to gaze output — it is simply computing a function of the subsampled pixels based on hard-coded parameters from the learned neural network model. The online operation is therefore lightweight and easy to optimize in hardware. We describe this process in more detail in the rest of this section.

3.3.1 Model Specification

Our base gaze prediction model is a feed-forward neural network as shown in Figure 3.1 [12]. The input layer is a $D \times D$ array of values I representing the eye-facing image. The pixel at row i and column j is given by I_{ij}. The desired output of the system is the gaze coordinates in the outward facing image plane (X, Y). The hidden layer of the model consists of K hidden units H_k. The model includes input-to-hidden parameters W^{IH}_{ijk} for each pixel location (i, j) in the eye-facing image and each hidden unit H_k; a hidden unit bias parameter B^H_k for each hidden unit H_k; hidden-to-output parameters W^{HO}_{kx} and W^{HO}_{ky} mapping between hidden unit H_k and the horizontal and vertical gaze coordinates (X, Y); and output bias parameters B^O_x and B^O_y for the horizontal and vertical gaze coordinates (X, Y). The hidden units use a standard hyperbolic tangent $(\tanh)$ activation function. The output units use linear activations. The mathematical formulation of the model is given below.

$$\hat{X} = B^O_x + \sum_{k=1}^{K} W^{HO}_{kx} H_k \tag{3.1}$$

$$\hat{Y} = B^O_y + \sum_{k=1}^{K} W^{HO}_{ky} H_k \tag{3.2}$$

$$H_k = \tanh\left(B^H_k + \sum_{i=1}^{D}\sum_{j=1}^{D} W^{IH}_{ijk} I_{ij}\right) \tag{3.3}$$

3.3.2 Model Learning

Given a data set $\mathcal{D} = \{I^n, X^n, Y^n\}_{n=1:N}$ consisting of N eye images I^n with corresponding gaze coordinates $(X^n,\ Y^n)$, our goal is to learn the complete set of neural network model parameters $\theta = \{W^{IH}, W^{HO}, B^H, B^O\}$. We learn the model parameters by minimizing a regularized empirical loss function between the neural network's predicted outputs $(\hat{X}^n,\ \hat{Y}^n)$ and the true outputs (X^n, Y^n) [12]. In this work, we use squared er-

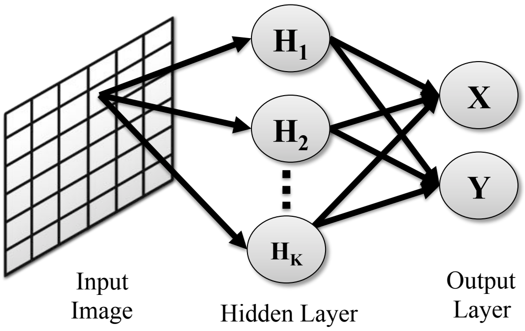


Figure 3.1: Illustration of the the neural network gaze prediction model.

ror as the loss function. The objective function $\mathcal{F}(\theta|\mathcal{D})$ is shown below for an arbitrary regularization function $\mathcal{R}(\theta)$ with regularization parameter λ.

$$\mathcal{F}(\theta|\mathcal{D}) = \sum_{n=1}^{N}(\hat{X}^n - X^n)^2 + (\hat{Y}^n - Y^n)^2 + \lambda\mathcal{R}(\theta) \tag{3.4}$$

The objective function $\mathcal{F}(\theta|\mathcal{D})$ cannot be analytically minimized with respect to the model parameters θ, so numerical methods are required. The gradients of the model parameters with respect to the loss can be efficiently computed using the standard backpropagation algorithm [63]. For standard, smooth regularization functions like the two norm squared $||\theta||_2^2$, the gradients of the regularization function $\mathcal{R}(\theta)$ are also easy to obtain. The base model can be learned using any numerical optimizer such as the limited memory BFGS algorithm [57].

3.3.3 Pixel Subset Selection

Given that the eye-facing images are extremely redundant, the central hypothesis of this work is that we can drastically sub-sample the eye facing images while preserving much of the accuracy.

The problem of choosing an optimal set of pixel locations is a subset selection or feature selection problem [35]. We refer to the pixels actually selected as *active pixels*. We can represent the set of active pixel locations using a binary mask A where $A_{ij} = 1$ if the pixel is active and $A_{ij} = 0$ if the pixel is not active. Given such an active pixel mask A, we can modify our neural network to base its prediction on the active pixel locations only. In Figure 3.1, this would correspond to simply removing all of the edges between the inactive pixels and the hidden units. The computation and communication complexity of image acquisition and gaze estimation are both linear in the number of active pixels in our framework. This means that we should expect a linear decrease in the energy cost of both image acquisition and prediction as the number of active pixels decreases.

To select a smaller active pixel set, we use a state-of-the-art sparsity-inducing group-ℓ_1 regularization function [89]. It is well known that the standard squared-error regularizer commonly used in machine learning and statistics shrinks parameter estimates toward zero, but it typically does not result in actual sparsity in the model coefficients. Tibshirani solved this problem for linear regression through the introduction of a method called the *lasso* for optimizing the least squares loss with a regularizer that penalizes the absolute values of the model parameters [76]. For linear models, setting a coefficient to zero is equivalent removing the underlying variable from the model and, as a result, such ℓ_1 regularization methods have been proven to be very effective at optimally solving subset selection problems.

In the present case, our model is a neural network with one parameter for each pixel in the image and each hidden unit. To solve the subset selection problem we need to simultaneously set all of the outgoing connections from a group of pixels to zero. This is likely not to happen when applying the standard ℓ_1 regularization function in a randomly initialized neural network model. However, the group-ℓ_1 regularizer developed by Yuan and Lin is designed specifically to drive groups of parameters to zero simultaneously. There are several different versions of the group-ℓ_1 regularizer. We make use of the group-ℓ_1/ℓ_2 regularization function as shown below. Note that in our case, we only regularize the input-to-hidden layer weights. The groups consist of all of the parameters from a given pixel to each of the K hidden units.

$$\mathcal{R}(\theta) = \sum_{i=1}^{D} \sum_{j=1}^{d} \left(\sum_{k=1}^{K} (W_{ijk}^{IH})^2 \right)^{1/2} \tag{3.5}$$

The neural network model parameters can then be learned by optimizing $\mathcal{F}(\theta|\mathcal{D})$ with the choice of regularizer given above.

3.3.4 Real-time Inference: Resource Usage vs Accuracy

It is important to keep in mind that all model learning described above happens in an offline setting. The real-time component of the gaze estimation system consists only of acquiring an eye facing image and performing a forward pass through the neural network. As shown in Equations (1)-(3), this forward pass requires only floating point addition and multiplication along with the computation of the $\tanh$ non-linearity. These are simple enough that they can be performed efficiently even on microcontrollers that do not have a floating point unit, and are limited in RAM.

One of the key benefits of our approach is that the sparse acquisition-based neural network model naturally lends itself to trading off energy consumption or resource usage for the accuracy of gaze prediction. Setting the regularization parameter to larger values will drive the parameters associated with an increasing number of pixel locations to zero while maintaining as much accuracy as possible. Fewer pixels implies a) less energy for using the imager, b) less computational needs for predicting gaze from the pixels, and c) less memory resources needed for storing the weights. All of these have advantages depending on the energy constraints on the eyeglass or the resource constraints on the microcontroller.

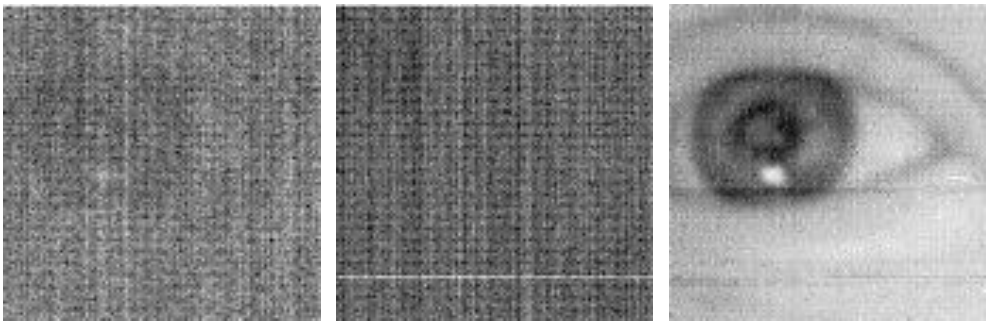
3.4 iShadow Calibration

One important practical aspect of using a gaze tracking system is calibration of the device to each user. We faced three challenges in designing the calibration procedure for the system. (1) There were many idiosyncrasies with the low-power Stonyman imager since it was an early version of a research prototype, and we had to perform extensive experiments across numerous parameters to understand its characteristics. (2) The image stream from the eye-facing imager changes depending on each individual's eye shape and eyeglass position, hence it was important to calibrate for each individual. (3) We needed to find ways to minimize burden on the participant and any manual overhead including adjustment of the imager position, manual labeling of images for learning, etc. all of which would make iShadow difficult to use for a new user. We detail the process by which we addressed these issues and calibrated the iShadow system for a new user.

3.4.1 FPN calibration

One drawback of the Stonyman camera is the noise issues inherent to the logarithmic pixels used by the camera. Such Fixed Pattern Noise (FPN) is typical in digital imagers, and results from small manufacturing imperfections including pixel size, material, interference with circuitry, etc which is unique to each individual imager. However, logarithmic pixels are particularly sensitive to these imperfections, meaning that most pairs of pixels have a noticeably different response to equal incident illumination. This effect yields an extremely noisy image if it is not dealt with.

FPN noise can be easily accounted for under the assumption that the noise remains stationary. In this case, it is simple to correct for the FPN by determining each pixel's response to uniform incident illumination and using this to generate an offset mask over the whole pixel array. The values in this mask are then subtracted from every image captured by the camera, which removes the effects of FPN completely. While FPN will remain constant under consistent lighting conditions and camera configuration settings, the FPN for the Stonyman camera is very sensitive to changes in outdoor lighting conditions. In addition, the dynamic range of imager is relatively low, resulting in saturation effects in bright sunlight.



(a) Image with FPN

(b) FPN Mask

(c) Image After FPN Correction

Figure 3.2: Stages of the fixed-pattern-noise (FPN) correction process. The raw image is collected with FPN present (a), from which the the static FPN mask (b) is subtracted. The resulting image (c) is mostly free of FPN.

Fortunately, we found that the imager's behavior under indoor lighting illumination tends to be relatively uniform, and the we could learn a mask that was reusable over a moderate range of lighting conditions. Under all indoor conditions, we have been able to use a single FPN mask per camera and generate consistently viable images. While this means that our system is not currently useful in outdoor settings, this is an artifact of the camera that will hopefully be resolved with further improvements to the hardware.

The process to learn a new mask for an imager is straightforward, and needs to be only done once prior to mounting the imager on the eyeglass. The imager must be exposed to uniform illumination so as to determine the relative offsets for each pixel in the mask. This should be done using a light-colored diffuser placed over the camera lens. A diffuser can be anything from a nylon mesh, as is often used in professional photography for illumination diffusion, to a thin sheet of paper. Once the lens is covered, the iShadow driver will capture and download an image to process as the new FPN mask for that camera. This process must be performed once for each of the two mounted cameras, and then FPN calibration is complete. See Figure 3.2 for an example of an image with and without FPN noise.

3.4.2 Collecting training data

Variations between subjects in the shape, position of the eye, and placement of the eyeglass mean that a new neural network model must be trained for each user, and therefore some amount of labeled training data must be generated per user.

The data collection process itself is quite simple. The subject sits in front of a display of some kind for a short period of time. We use a simple script that shows a black field on the display, and a white circle of some small radius on that field. Subjects are fitted with the iShadow glasses and seated in front of the display. To maximize the effectiveness of the training data, the user should be positioned so that the display extends to the edges or just beyond the edges of the scene-facing camera's field of view. This will ensure that there can be training data samples generated over the entire field of view. Checking that the system is in proper position can be done quickly by pulling a few images from the outward-facing camera, and displaying this to the user in real-time so that the user can adjust the position of the eyeglass.

When data collection is triggered to begin, the circle begins moving in a random pattern over the space of the display. Subjects are directed to focus their gaze on the circle as it moves across the screen, and iShadow begins collecting full-frame images from both the eye-facing and world-facing cameras. The exact training time needed depends on how accurate of a model is needed, however, as we show in our evaluation, after accumulating one minute of labeled training data, adding more does not yield a significant increase in the accuracy of the predictor. Thus, the training session can be very brief without compromising on the accuracy of the generated neural network model.

After training data collection is complete, the collected images need to be transferred to a computer to train the neural network model. This can be done by storing the images on an SD card during the session or live streaming via USB.

3.4.3 Labeling training data

One potentially time-consuming aspect about learning a model for each user is generating labels from the collected data. Since the user's head position depends on height and how he/she is seated, we cannot just use the pixel positions where the target circle is drawn on the computer screen. Instead, we process the image stream from the outward-facing imager, and use a simple computer vision algorithm to detect a light-colored patch on a dark field (rest of the screen). In cases where this process fails (often because the dot is on the edge of the field of view of the imager), the calibration algorithm asks for assistance from the human, but this is largely unnecessary for the training process. Depending on the amount of training data, the process generally takes only a few minutes.

3.4.4 Robustness to blinks and head movements

One question with calibration is whether there is a negative effect of the users blinking or possibly drifting from the target for brief periods. This is not a significant issue since we are able to generate a high volume of data over a short period of time through automatic labeling. As a result, even if there are periods where the user was blinking or the user's gaze moved off-target for a brief of period of time, these are treated as noise by the neural network learner as long as the majority of the samples in the training set involve the user being correctly focused on the target. Our experiments show that training under this assumption yields models that have a high degree of prediction accuracy.

3.4.5 Learning the model

In addition to images from the eye-facing camera, and a corresponding set of gaze coordinates from the labeling session, the model learner also requires a set of regularization parameters λ, each of which will yield a different model with differing sparsity values. The process for choosing the λ values depends upon the application. For our study we swept the training across a series of values ranging from very high to very low sparsity to generate curves for the effect of λ on gaze prediction accuracy as well as power consumption.

The suitable regularization parameter can be chosen in a few different ways. The default option is to find a good tradeoff between prediction accuracy and power. We generally find that there is a sweet spot for each individual i.e. there is a small range of sparsity that yields good prediction accuracy at a very low power cost, and this may be a good standard target for the model generator. Another option is dynamic generation based on the desired value of a certain system parameter, such as prediction accuracy or frame-rate. In this scenario, the model generator would sweep over a standard set of lambda values, generating a prediction model for each one. The model trainer estimates system error based on cross-validation during training, this value can be used to find an acceptable λ value and a corresponding model to meet a prediction accuracy criterion. If the goal is a certain prediction rate, then the same process can be performed using sparsity as the metric.

Once the λs have been decided (or if they are decided dynamically), the calibration program trains a neural network model for each and saves the corresponding model parameters. Once a final model has been chosen to be used for online prediction, it is automatically loaded onto the glasses. At this point, the system is prepared to do gaze prediction for the new subject.

3.5 iShadow System

Figure 3.3 shows a system diagram and photos a prototype version of iShadow; our current hardware implements all the design elements described in previous sections. We now briefly describe the key hardware sub-components used in the prototype and describe our optimized real-time gaze tracking implementation.

3.5.1 iShadow Platform

The iShadow platform features a standard glasses frame with two low-power cameras, an inertial sensor, microcontroller, and bluetooth, as seen in Figure 3.3. One camera is mounted in front of the user's right eye for acquiring eye-facing images. The other is mounted in the center of the frame facing outward to capture the center of the user's field of view. We use the standard optics on the image sensors, which give a 36° field of view.

Our hardware is designed with the constraint that needs to be thin and lightweight enough to be mounted on the side frame of a pair of eyeglasses, and roughly the same form-factor as a Google Glass. This required several hardware revisions and optimizations to make all components fit in the appropriate size. Our latest prototype is shown in Figure 3.3 and the components are described in Table 3.1. Of the components listed, the focus in this paper is primarily on the eye-facing imager, and computation of gaze on the MCU. Since the imager is central to our algorithm and implementation, we now turn to a more detailed description of its innards.

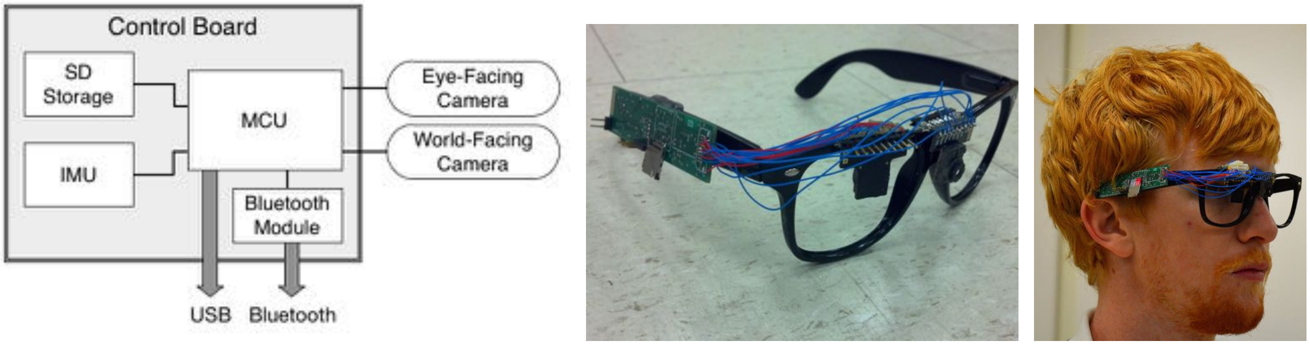
Eye-facing imager	Stonyman 112x112 greyscale [72]
World-facing imager	Stonyman 112x112 greyscale
Inertial Motion Unit	Invensense 9-axis IMU [37]
Processor	STM32 Arm Cortex M3 microcontroller [71]. 32 MHz processor; 48KB memory; 384KB flash storage
Storage	microSD card, 64GB max
Radio	Bluetooth

Table 3.1: iShadow platform components

3.5.1.1 Image Sensors

Our hardware framework is built around the Stonyman Vision Chip produced by Centeye, Inc.[1] This device fits our research purposes for several reasons. First, it is a low-power embedded camera, consuming approximately 3 mW when operating at full power. (see Table 3.2 for details). Second, the design of the pixel array allows for random access to

[1] http://centeye.com/products/stonyman-vision-chip-breakout-board



(a) Diagram (b) Platform (c) On Head

Figure 3.3: Figures show an architecture diagram and different views of the third-generation iShadow prototype. The prototype has two cameras, one at the center front and one facing the eye, with the electronics mounted on the control board on the side. Batteries, while not shown, are mounted behind the ear.

individual pixel values. The combination of low-power operation and random access capability makes the Stonyman unique among commercially available image sensor chips.

The Stonyman features a 112x112 square grid of pixels. These pixels are characterized by their logarithmic voltage response to lighting conditions. This allows for a greater dynamic range compared to a pixel that operates linearly with respect to lighting conditions. The use of logarithmic pixels allows a random-access interface, which the Stonyman provides via a register-based control scheme. It is this feature specifically that enables the energy-accuracy trade-offs that we explore in this work.

Like many contemporary mobile image sensors, the Stonyman sensor provides a sleep mode. We exploit this feature in iShadow to use power more efficiently. The majority of the power drawn by the camera while it is active comes from the pixel acquisition circuitry, and this can be powered down using a control register. In this low-power state there is only a small power draw - less than a half a microwatt - for the digital control circuitry that maintains the values in the control registers and allows the camera to be switched back to full-power mode. There is a small delay time, on the order of a few microseconds, for the camera to power up and back down. These delay times, as well as the power consumption of the camera in wake and sleep modes, are given in Table 3.2.

Finally, while we made the choice to use the Stonyman imager as the outward-facing camera in addition to the inward facing one, the outward facing imager can be replaced with a higher-end device if better quality images are needed for vision processing. The insight in this paper is that gaze coordinates can be obtained with a very low-power and low-resolution eye-facing imager such as the Stonyman camera.

Active to Sleep Delay	4 μs
Sleep to Active Delay	10 μs
Active Power Consumption	3.13 mW
Sleep Power Consumption	0.041 μW

Table 3.2: Stonyman Power Consumption and Transition Times in and out of Sleep Mode

3.5.2 Basic Operation Modes

At the lowest level, iShadow supports three basic modes of system operation — full image capture, real-time gaze tracking, and life logging. However, the system's functionality is not limited to the modes outlined here, as one of the primary benefits of iShadow over existing commercial eye trackers is its programmability. Users who have a specific application in mind that might benefit from a more complex operating scheme can implement and run it on the system themselves, allowing for a much broader set of usage scenarios than those outlined here.

3.5.2.1 Full image capture

In this mode, iShadow is continuously capturing full-frame images from the inward and outward-facing imagers and storing it into on-board flash. This mode is intended primarily for offline data analysis, and adjustment of parameters during the calibration phase (described above). By tightly optimizing the image capture method and interleaving control signals with ADC delays, we are able to drive both cameras at the same capture rate as if we were only driving one. The full-image capture rate for both cameras is 10 Hz.

These images can be stored to an onboard SD card or transmitted to another device via USB or bluetooth. For storage, we have designed our system to use a double-buffered scheme that allows the SD communication peripheral to read and transmit already-collected pixel data while more data is written to the second buffer. There is still some delay introduced by the SD card writes, as there are a small number of operations needed to trigger the SD card write process. However, this improvement hides the majority of the SD write latency. Since our available RAM is too small to hold an entire frame, let alone two double-buffered frames, we write data at regular intervals during image collect to prevent overflow.

3.5.2.2 Real-time gaze tracking mode

The most useful operating mode of iShadow is real-time gaze prediction, where it is continuously processing the pixels from the eye-facing imager to output gaze coordinates. In this mode, iShadow is solely processing image streams from the eye-facing imager. Once a gaze prediction has been completed and a gaze estimate generated, it can be used in several ways. One option is to store values for some form of post processing later - this is especially useful when run the eye-facing imager runs in conjunction with the world-facing imager, as in the final operating mode below. The gaze data can also be used for triggering more complicated on-board processes or for making higher-level inferences about the state of the eye or the wearer, as discussed in the following section.

3.5.2.3 Lifelogging mode

iShadow's third operating mode is a simple extension of real-time gaze-tracking. Simply put, it is capturing what in the outside world the user is looking at. By running the real-time gaze inference algorithm using the eye-facing imager and taking full images from the world-facing imager, the system records where the user is looking and what they are looking at. This type of ”lifelogging” is becoming more prevalent as wearable computing grows cheaper and more accessible.

In this mode, the bottleneck is the outward-facing image capture. By doing interleaving of prediction operations with ADC reads the predict time can be completely hidden, however, there is no way to significantly reduce the time needed to collect a full-frame image from the outward-facing camera. This operating mode is a simple but straightforward example of the types of applications that iShadow facilitates, and does not require any additional logic on top of the existing firmware.

3.5.3 Gaze-triggered applications

While the focus of this paper is not on designing gaze-driven applications, we briefly describe a few example applications that can be designed over the above operating modes. Several higher-level inferences are possible from a continuous stream of gaze coordinates: a) Eye fixations can be detected by looking for windows when the gaze coordinates are relatively steady with small changes due to head movements, b) Smooth Pursuit can be detected by looking for slowly changing gaze coordinates, and c) Sudden events can be detected by looking for a large saccade or change in gaze. These higher-level inferences could suggest different things about the visual state of the outward-facing imager — for example, eye fixations could be during conversation with an individual, smooth pursuit could be while reading from a book or screen, and sudden events could be an unexpected change in the surroundings. In the case of disease progression, abnormal gaze patterns may be detected, for example, abnormally frequent saccades. Once such a high-level inference detects an event of interest, it quickly triggers the outward-facing imager to capture the state of the world, and stores this data into the local SD card or streams the data over bluetooth to a mobile phone.

3.6 Evaluation

To test the effectiveness of iShadow at accurately and efficiently predicting the wearer's gaze location, we collected sample data from ten different subjects - eight men and two women from the ages of 22 to 31. We ran each subject through the calibration process outlined in section 3.4, excluding the FPN mask generation as it does not affect the per-individual results. We generated at least five minutes of labeled data for each subject in full-image-capture mode. To generate the ground-truth labels, we use the approach described in section 3.4. Running at the maximum framerate of 10 Hz, the resulting dataset includes at least 3000 images per user. We used this data to perform a variety of experiments to determine whether we had reached our design goals.

We present our evaluation in three sections. Section 3.6.1 details our experiments to determine whether the neural network model and the ℓ_1-subsampling method are capable of accurately estimating the wearer's gaze coordinates. In section 3.6.4, we evaluate the tradeoffs between the model parameters and the resulting performance of the hardware, especially the image capture pipeline.

3.6.1 Evaluation of Neural-Network-Based Pixel Selection

We first evaluate the benefits of our neural network-based method to select a sparse set of pixels. Using the data collected, we performed a number of experiments to investigate the accuracy of the neural network gaze prediction model as a function of the number of active pixels. We generated a set of 10 values of the regularization parameter λ, and trained each user's data over this same set of λs. Our per-user, per-λ experimental framework is based on a standard five-fold random re-sampling protocol. For each fold, we divided the available data into a training set and a test set completely at random using an $80/20$ split.

We use the training set to estimate the parameters of the model and use the learned model parameters to predict gaze locations for each test image. We compute the gaze prediction error for a given test fold in terms of the average squared distance (ℓ_2 distance) between each true gaze location in the test set and the predicted gaze location given each eye image in the test set. We average the prediction error over the five test folds to generate an error value for that user and λ. In addition, the size of the pixel selection mask varies with

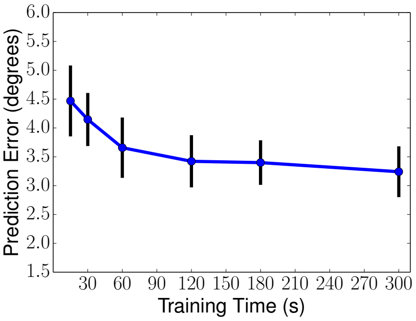
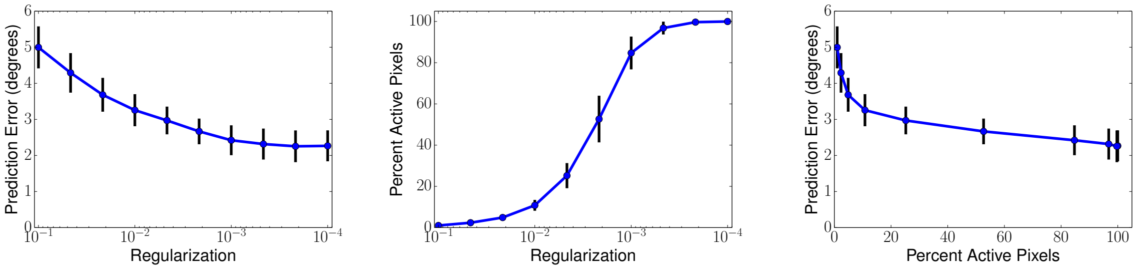


Figure 3.4: Amount of training time for the system versus the resulting gaze prediction accuracy.



(a) Lambda vs Accuracy

(b) Lambda vs Model Size

(c) Model Size vs Accuracy

Figure 3.5: Plots (a) and (b) show the effect of regularization parameter on gaze prediction accuracy and model size respectively. Plot (c) shows the net result, which is how the number of pixels acquired can be reduced dramatically (up to 10×) with minor effect on gaze prediction accuracy.

each fold as the optimizer finds slightly different solutions when using different training data sets. To account for this, we also average the size of the pixel mask generated over the five folds.

After generating per-user data, we average the prediction error and pixel mask sizes across all ten users to generate our results below. While there is moderate variation in the results between users due to differences in eye position and shape, the general trends across users are consistent and can be seen in the results we present here. For all of our results, we present the mean value and the 90% confidence interval unless otherwise indicated.

3.6.2 Accuracy – Model Size tradeoffs

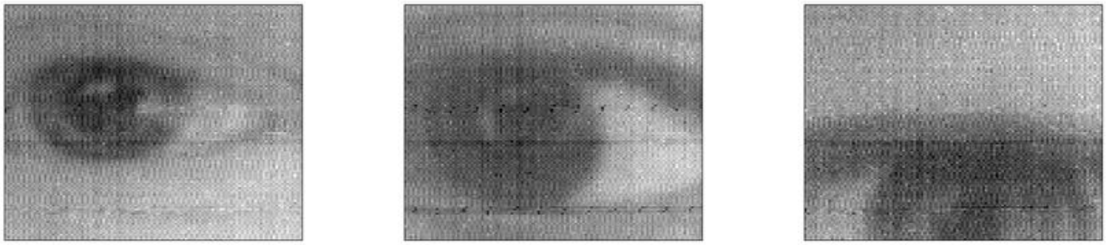
One of the benefits of our algorithmic framework is that it is able to provide a variety of models that offer different tradeoffs between the overall complexity of the model (i.e. number of pixels sampled, and number of weights for computation) and the accuracy of the model (i.e. the precision in degrees). This tradeoff is enabled by using different choices of the regularization parameter, which varies the penalty for model complexity compared to the accuracy.

First, we look at whether the regularization parameter is a useful knob for tuning model size and accuracy. Figures 3.5a and 3.5b show that varying the regularization parameter enables us to select a spectrum of models with different prediction accuracies and different fractions of selected pixels, respectively.

Figure 3.5c shows just the prediction accuracy vs model size — interestingly, we see that varying the percentage of activated pixels from 100% down to about 10% only has a minor effect on the prediction accuracy. This shows that there is substantial redundancy in the eye image, and the neural network is able to predict gaze just as well with 10% of the pixels activated as 100% of the pixels. On our imager, this means that sampling 10K pixels per image vs sampling 1K pixels per image has roughly the same prediction error, which in turn can translate to substantial reduction in power consumption.

We also see that the accuracy of the neural network model does not drop below about 2° even when the entire image is sampled. In contrast, high-end mobile eye trackers advertise accuracies of as low as 0.5°. The primary reason for this gap is that the eye-facing imager lens is mounted at a fixed position on our eyeglass in our current iShadow prototype, and has a limited field of view of 36°. As a result, for some individuals, the pupil was rarely completely within the field of view of the imager.

This can be seen in Figure 3.6, which compares sample eye images between users for whom the predictor performed well, average, and poorly compared to the rest of the set. In general, having the entire range of the pupil's motion visible to the camera, as well as



(a) 1.32° Error (b) 2.19° Error (c) 3.40° Error

Figure 3.6: Comparison of eye images from multiple users, giving the average prediction error for that user. Notice that the position of the eye in the image and iris / sclera contrast have a prominent effect on prediction accuracy.

a strong contrast between the iris and the white of the eye (sclera) seem to be the most significant indicators of good predictive performance.

Despite the gap, we argue that a 3° accuracy error is acceptable for a variety of real-world applications, particularly when the object being viewed is in relatively close proximity, for example, detecting the individual with whom a user is engaged in face-to-face conversation. Figure 3.7 provides a better understanding of the error in the outward image plane. Given that the imager already has a small field of view, gaze coordinates with roughly 3° error is reasonable enough to get a good idea of where the user is looking.

To get a better idea of how the neural network selects weights, we look at some example weights learnt by the hidden units. Figure 3.8 shows the weights for each hidden unit. Each example has approximately 10% of pixels active We can also see that the group-$\ell 1$ method learns to focus on the region of the eye image where the pupil and the white of the eye are most likely to appear, as one would expect.

3.6.3 Calibration time

To evaluate how much training data is needed to generate an effective model, we look at how quickly the gaze prediction converges as the amount of data that we use for training increases. Figure 3.4 shows the results for a particular choice of the regularization parame-

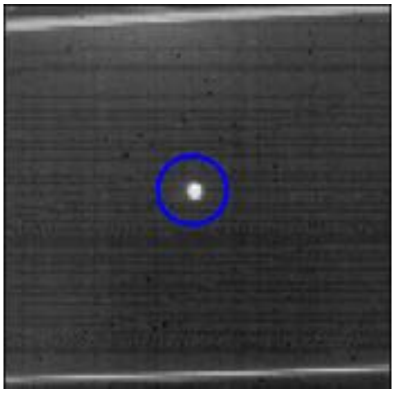


Figure 3.7: The circle gives an example of 3° of error in the outward imager plane around the white dot. The error is less if the eye is fully within the field of view of the imager, and higher when not fully in the field of view.

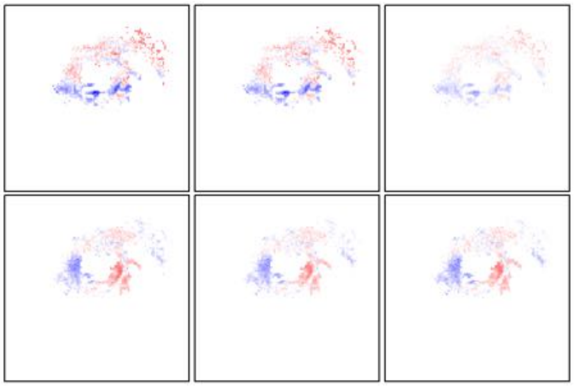


Figure 3.8: This figure shows the weights learned by each hidden unit in the neural network model for subsets of approximately 10% of pixel locations.

ter ($\lambda = 0.01$). We see that the convergence is very fast — even if there is only 60 seconds of data used for training, that is sufficient for the algorithm to determine the appropriate parameters with little to no improvement as the amount of training data increases. Similar results were seen for other values of λ. Thus, the time for calibration is not a bottleneck in system operation.

3.6.4 iShadow System Evaluation

We now turn to an evaluation of the iShadow platform predicting gaze in real-time with models trained using the neural network-based sparse sampling algorithm and appropriate regularization parameters, λ, as described earlier.

We wish to evaluate three key performance metrics — gaze tracking rate, prediction accuracy, and energy consumption. While gaze prediction accuracy and energy consumption are self-explanatory, gaze tracking rate requires a bit more explanation. Unlike a traditional eye tracker with an active pixel imager that involves exposure followed by frame capture and image processing, we use a logarithmic pixel imager where the pixels can be continuously read out at any time. Thus, we refer to gaze tracking time as the amount of time after sampling the first pixel of the pre-selected sparse set until we obtain the results of the gaze. Gaze tracking rate is the number of such gaze predictions we obtain per second.

3.6.4.1 Tracking rate vs prediction accuracy

By tuning the regularization parameter, λ, we can tradeoff between the three performance metrics. Let us first consider gaze tracking rate vs accuracy while fixing the power budget. Lets start with the case when all pixels are sampled — here, accuracy is high since all pixels are available, but gaze tracking rate is poor since fewer images are sampled per second. By increasing the level of regularization, we can progressively decrease the time needed to make a single prediction since: a) fewer pixels need to be read from the camera and there are fewer calculations to perform, and b) the number of feature weights is proportional to the number of pixels, hence the memory footprint of the model also decreases. Thus, we can increase gaze tracking rate to capture rapid gaze changes, but we suffer in the accuracy of tracking due to a coarser model with fewer pixels.

Figure 3.9 provides an empirical evaluation of the rate vs accuracy tradeoff on iShadow. We run iShadow in always-on mode and progressively reduce the model size from large (low error, low rate) to small (high error, high rate). The results are as expected — for large models, we get prediction errors of roughly 3° but low gaze tracking rates around 10 Hz. As the models reduce in size, gaze prediction errors increase to roughly 4°, but gaze tracking rates increase to 30+ Hz as well. For reference, commercial eye trackers operate at 30 Hz or higher, therefore, these results show that with a small reduction in accuracy, iShadow can achieve sampling rates comparable to commercial eye trackers.

3.6.4.2 Tracking accuracy vs Energy consumption

We now turn to gaze tracking accuracy vs power consumption. When evaluating energy consumption, we need to consider two hardware components, the micro controller (MCU) and the imager, and two software subsystems on the MCU, pixel acquisition from the camera, and the cost of running the gaze prediction algorithm. Thus, we do a three-way breakdown of energy: a) energy consumed for pixel capture by the imager, b) energy consumed for pixel acquisition by MCU, and c) energy consumed for executing the gaze prediction algorithm on the MCU.

Figure 3.10a shows the results for a fixed gaze tracking rate of 4 Hz. The system is duty-cycled whenever it is not actively performing gaze acquisition or prediction. We measured power between our on-board voltage regulator and the relevant circuit at a rate of 30 KHz, and report the energy breakdown for each gaze prediction across the different components. (Power consumption is just 4× the reported energy number since gaze tracking rate is 4 Hz.)

The general trend in the graph is as expected — a smaller model means less energy consumption but higher error. More interesting is the breakdown across the hardware and software components. The results show that the energy consumed for prediction is roughly the same as the energy consumed by the camera. But the main cost of acquisition is at the MCU, which needs to set appropriate control lines to select pixels, and read-out the pixels over a serial bus. The cost of gaze prediction is about a third or less of acquisition cost.

Figure 3.10b shows the time spent in each subsystem per gaze prediction — since a gaze tracking output is obtained every 250 ms, this shows what portion of this time period is spent per subsystem. Since the camera ON time is the same as acquisition time, we show only the plot for acquisition in the figure. The result shows that the least amount of time is spent in the gaze prediction operation, and more time is spent in pixel capture and acquisition by the MCU.

Overall, these results demonstrate that sub-sampling pixels is extremely important to reduce power consumption of the iShadow platform — gaze prediction consumes less time and energy compared to pixel capture and acquisition, clearly justifying the benefits of sparse acquisition. This is facilitated in hardware by the Stonyman's random-access pixel capability, without which the time and energy for the MCU to acquire all of the pixel data would dwarf the benefits of decreased gaze prediction time and camera sleep.

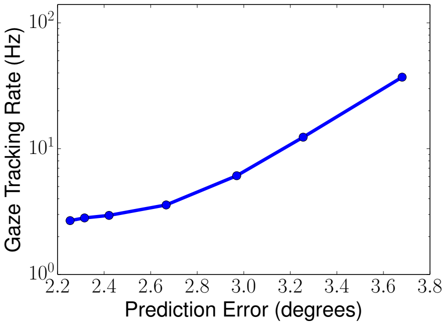
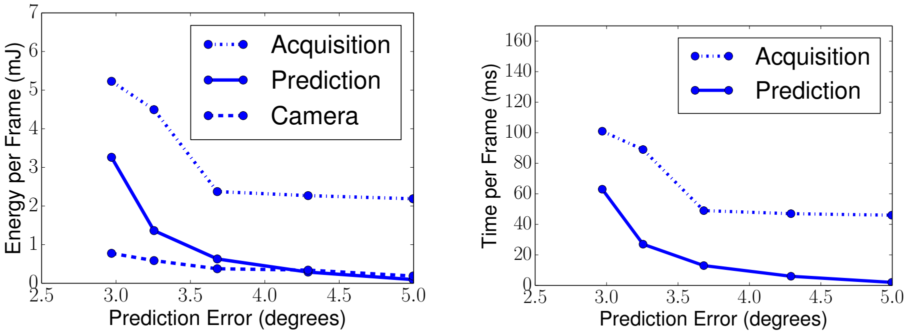


Figure 3.9: Smooth tradeoff between gaze tracking rate and prediction error as λ is varied. The MCU is always active in this experiment.



(a) Accuracy vs Energy

(b) Accuracy vs Time Elapsed

Figure 3.10: Breakdown of energy and time for different subsystems during the capture and predict process. (a) is the amount of energy consumed for pixel acquisition by the MCU, the gaze prediction computation, and cameras, and (b) is the amount of time spent by the MCU in acquisition and prediction.

3.7 Discussion

The design of a computational eyeglass platform is extremely complex due to the complexity of the sensors, the positioning and mounting on the eyeglass frame, and the demanding computational needs for processing image streams. While our results are promising, there are several steps to take before such platforms are ready for more widespread use. We discuss some of these steps in this section, as well as the types of applications that our platform is suitable for.

3.7.1 Imager + IMU fusion

While the techniques that we use in this paper can be used to identify gaze direction, we need additional information about head movement to precisely determine whether the individual is fixated at the same location in the external world. For example, the head might move about during conversation but the eye might continue to track the individuals face. Our platform incorporates an IMU in addition to imagers to enable such fusion, but our algorithms have not yet taken advantage of this capability.

3.7.2 Image field of view

As described in §3.6, one of the reasons why our gaze tracking performance varies across individuals is because of the field of view of the eye-facing imager. We currently use a lens that has a field of view of 36°, as a result of which the entire eye is not in the visual field of the imager for some subjects. We believe that this issue can be fixed using a lens with a wider field of view, such as a fisheye lens. One important advantage of the neural network-based gaze tracking algorithm used in iShadow is that it is more robust to distortions caused by different lens than canonical vision-based techniques. Even though a fisheye lens may be expected to distort the image, the neural network technique is not sensitive to such distortions compared to an algorithm that does shape fitting, hence our approach can be expected to retain or improve performance.

3.7.3 Inward-facing imager placement

An obvious consideration from an aesthetic perspective is how to place the inward facing imager such that it does not obstruct the field of view of the user. Several possibilities present themselves — for example, the pixels of an imager may be mounted all along the rims of the eyeglass, or the imager may be mounted on the side frame and observe a reflection of the eye on the spectacles lenses. While such design aspects are beyond the scope of this work, we note that a key benefit of the techniques presented in this paper is that it can be easily adapted to new eyeglass designs.

For example, consider the case where pixel imagers are mounted throughout the eyeglass rim. Depending on the shape of a user's eye, different sets of pixels may carry information that is relevant to gaze. These minimal set of pixels can be selected by applying techniques described in this paper. The same is true for the second example, using a reflected eye image. Thus, our methods generalize to a variety of other designs where subsampling of pixels may be useful.

3.7.4 Robustness across users

We have described the variability of per-user performance and the improvements we plan to implement to make performance better and more consistent across subjects that it has been trained on. However, our end goal is to provide a system that works with little or no per-user training required. This is critically important for accessibility and ease of use. The first step towards this goal is addressing the issue of eye localization in the image, either by altering the hardware to allow repositioning of the camera or by automatically adjusting for the position of the eye in the image. Once that issue has been addressed, we plan to collect data from a much larger number of subjects and begin building and testing "universal" models that will work at least moderately well on a new subject without requiring prior training.

3.7.5 Comparison to other eye tracking devices

The development of specialized eye tracking hardware has accelerated rapidly over the past decade, to the point where remote (non-mobile) devices are expected to have only a few tenths of a degree of predictive error [34]. Even mobile trackers report errors of 1° or less [6]. The fact that the error rates we report are moderately higher than this immediately begs the question of why we chose to introduce this system. Part of the discrepancy is the immaturity of the system - iShadow is still in early development, and accuracy can be expected to improve noticeably as we refine the system and tracking algorithm. It would be overly optimistic, however, to assume that such progression will bring us to the same order of magnitude of accuracy as industrial tracking devices.

However, current mobile trackers remain too bulky to be deployed comfortably in real-life scenarios and have too short of a battery life for even moderate-length studies of 4+ hours. Because of iShadow's low-power design, it offers a significantly longer run time and smaller form factor than existing mobile devices. Once the design of the system has been refined, we envision it being used for long-running experiments in the wild. For these types of applications it is unlikely that extremely fine-grained accuracy will be needed. iShadow's current accuracy is easily enough to identify, for example, what object the user is looking at in a scene. It is not enough to distinguish gaze between lines of text, but tasks of that nature are generally better suited to static test environments.

In short, all eye tracking hardware represents a tradeoff between accuracy and mobility. The current generation of mobile eye trackers give moderate mobility at the cost of some accuracy. We present iShadow as a further step in that same direction, giving excellent mobility at a slight accuracy cost over and above that of current industrial mobile devices. For tasks that require such mobility, we believe that iShadow is the best option currently in existence.

3.8 Conclusions

As the first major contribution of this work, we present a first-of-its-kind low power eye tracker that is designed to predict gaze in real-time while operating with a power budget of a few tens of milliwatts. This chapter presents a soup-to-nuts implementation of such a system, including a full hardware prototype, a new neural-network based algorithm for sparse pixel acquisition, a full implementation of a real-time gaze predictor on a microcontroller, and evaluation on several subjects. Our approach exploits the unique properties of random access pixel cameras to achieve a flexible energy-accuracy trade-off in the wearable/real-time setting. Our results show that we can dramatically reduce power consumption and resource needs by sampling only 10% of pixel values, without compromising accuracy of gaze prediction. These results are highly significant in that they offer a clear path toward ubiquitous gaze tracking for a variety of applications in computer vision, behavioral sensing, mobile health, and mobile advertising.

CHAPTER 4

CIDER: LEVERAGING THE MOTION OF THE EYE

In the previous chapter of this thesis we presented a first pass at the creation of a lightweight wearable eye tracking system. However, while it represents a major stride forward, its level of performance is insufficient to meet the standards of a realistic device that people could use on a daily basis in terms of power consumption and eye tracking accuracy. In this chapter we discussion a major redesign of the original platform with the goal of dramatically improving the utility / resource consumption tradeoff that was explored in the first chapter.

We explore the problem of pupil tracking so as to be able to more explicitly leverage temporal features of the eye. The ANN model from the iShadow work accepted as input a single image of the eye with no context, and leveraged the strong visual structure of the eye to great effect. However, there is also a great deal of temporal structure to be leveraged — that is to say, there is a reasonable limit to how far the eye can move between image frames. Thus, if we are able to record fast enough, its motion can be modeled as smooth and we can make assumptions about its current position based on recent frames. We demonstrate that a platform built around this concept, which we call CIDER, can estimate pupil center with error less than two pixels (0.6°), and pupil diameter with error of one pixel (0.22mm). Our end-to-end results show that we can operate at power levels of roughly 7mW at a 4Hz eye tracking rate, or roughly 32mW at rates upwards of 250Hz.

4.1 Background and Motivation

Our initial work on the iShadow platform represented a significant step towards realizing the goal of a wearable eye tracking device. However, in order for the system to be usable in the daily lives of individuals — such as for long-term in-the-wild studies — it must meet the design requirements of both a commercial wearable device and a state-of-art eye tracking device. In terms of being a wearable device, the primary concerns are size / form factor and battery life. We address form factor and its attendant engineering challenges in chapter 5, and for this chapter we focus on power consumption. While the initial work had minimal power draw in approximately the 70 mW range, realistic wearable devices such as the FitBit operate in the single-mW.

In addition, while we were able to develop an eye tracking method with an average error of 3°, commercial eye trackers advertise error on the order of 0.1°. Exacerbating the issue is the fact that the neural network algorithm's performance is very sensitive to shifts in the input data. Therefore, a major performance issue for the system is variation in lighting conditions, as this changes the values of the input pixels to the network and accuracy decreases dramatically. This lack of robustness represents a major practical problem for the use of the system in the real world. Thus, at the conclusion of the first work there was still a significant gap between the iShadow platform's performance and what it must be in order to be feasible as a research-grade tool in terms of both power efficiency and multiple aspects of eye tracking performance.

4.1.1 Proposed Solution: CIDER Eye Tracking Architecture

Our solution to these challenges, and the fundamental contribution of this chapter, is the design of a staged architecture for computational eyeglasses that can trade off power and robustness to illumination conditions, while improving eye tracking accuracy overall. The principle underlying our architecture is well-known to systems researchers — we optimize heavily for the common case but provide more power-hungry features to deal with the more difficult but uncommon scenarios that occur.

4.1.1.1 Common-case optimization

The “common case” for this platform refers to the presence of the eye within the camera’s view and its motion in between sequential camera frames. Specifically, it is reasonable to assume that in most situations there will be long periods of the the eye’s position experiencing little to no change from one image frame to the next. This is because (a) much of the time the eyes are fixating on a target and move very little, (b) given a sufficiently high framerate, even fast motions of the eye will result in little frame-to-frame change, and (c) eyelid blinks (which hide the eye and violate this assumption) take up a relatively small amount of time.

By taking advantage of the nature of this common case, it is possible to do eye tracking even more resource-efficiently than in the previous system design. This is because we can leverage temporal information (i.e., the position of the eye in the previous frame) to inform our sensing in the current image frame and use this to guide our adaptive sampling strategy. With this added information, it is possible to do dramatically more efficient tracking because the search space is made significantly smaller. In practice, we are able to do effective eye tracking with only a few tens of pixels sampled per eye position estimate.

However, there are situations that violate the “common case” assumption regarding smooth motion of the eye. As already mentioned, the most obvious is blinks of the eyelid, which block the eyeball from view momentarily. During that time, it is likely that the eye’s position will change slightly, so the temporal assumption cannot be safely held across a blink event. To deal with blinks and other artifacts such as reflections off the eye which may temporarily change its appearance, we include a one-shot algorithm (no temporal assumptions) which has higher sampling cost but also performs much more robustly when the temporal assumption breaks down.

4.1.1.2 Results

Despite the higher cost of the one-shot algorithm, the common case is so frequent (approximately 90% of the time, in our experiments), the increased efficiency gained from leveraging it improves the system's overall power-to-performance ratio dramatically. The overall system power consumption for the common case is about 7mW — compare to the 70mW power consumption of the original iShadow. In addition, this more focused sampling regime offers significantly better eye tracking fidelity due to the simpler nature of the problem. Whereas the original platform featured 3° average gaze error, the improved platform has an average pupil tracking error of 0.4°.

In addition, one of the interesting auxiliary benefits of our staged processing pipeline is that it can operate at very high frame rates during typical operation. Our optimized pipeline can operate at rates exceeding 200 fps, which is at the the high end of tethered remote eye trackers. This capability is particularly useful for detecting small fine-grained saccadic movements which happen while reading or when fatigued, providing further window into an individual's neural activities. Our algorithm, CIDER (CIrcle Detection of Edges with Reinforcement), is the first wearable eye tracker to achieve such high frame rates.

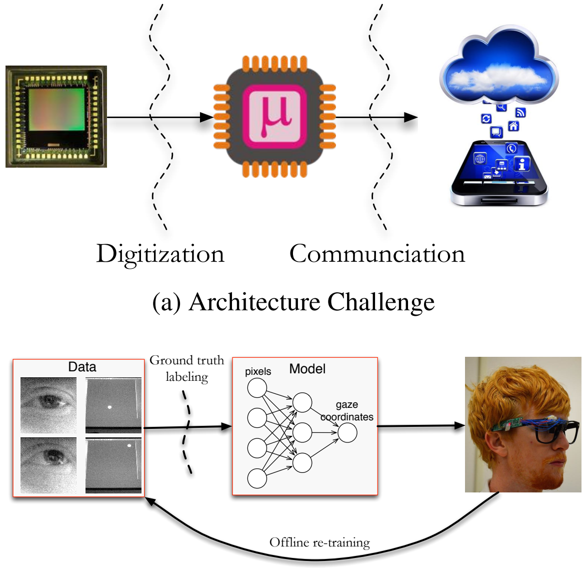
4.1.2 CIDER Performance

Our experiments show that

- CIDER can track pupil center with accuracy of roughly 1 pixel (0.3°) and pupil dilation with accuracy of approximately 1 pixel (0.22mm) in indoor lighting conditions.

- CIDER adjusts to indoor and outdoor illumination using an indoor-outdoor NIR detector in conjunction with different models and hardware settings. We show that the pupil center error increases only by a modest amount in outdoor settings (4 pixels or 1.2°).

- We operate end-to-end at a total power budget of 7.5mW when running at 4Hz, which is 10× less than our previous work, which was the former state-of-art in this area.



(a) Architecture Challenge

(b) Modeling Challenge

Figure 4.1: Design challenges. (left) Two major challenges in system design are the power consumed for digitizing pixels from the camera and the power consumed for high-rate communication to a mobile phone. (right) Another major challenge is calibrating the system during online operation without requiring explicit interaction with the user.

Alternatively, we can achieve eye tracking rates of upwards of 250 frames/second by scaling power consumption up to 32mW.

4.2 Fundamental Design Tradeoffs

As has been discussed, robust estimation of eye measures on a computational eyeglass presents a number of technical issues that cut across many aspects of design. In this section, we take a look at the most significant of these issues in greater detail and discuss possible options for improving overall performance. We separate each component of the system and identify the key robustness–power tradeoffs that they present.

4.2.1 Sensing

Continuous operation of a camera is power-hungry. The energy cost of sensing primarily arises from digitization of pixels — while the analog electronics of a CMOS camera has very low power consumption (few milliwatts), digitizing hundreds of pixels per image at high frame rate (upwards of 30fps) consumes orders of magnitude more power, resulting in a power consumption of several tens to hundreds of milliwatts for typical cameras.

To reduce power consumption of such a camera, we would need to throttle the rate at which pixels are digitized. This can be done in one of several ways, including sub-sampling the image, reducing the resolution of the image, or acquiring fewer frames. However, reducing power consumption in this manner can be detrimental to dealing with variations — for example, in the presence of variable illumination or shadows, acquiring more pixels and more frames is better since it provides more contextual information and facilitates more robust de-noising methods.

4.2.2 Computation

Once pixels are acquired from the imagers, we need to process them to estimate eye parameters. The computational demands of continuous high-rate image processing (filtering, feature extraction, and detection) are significant, and require CPUs that have more resources and are more power-hungry than low-power MCU-class processors. One way of addressing this issue is simply to use an MCU with a higher clock rate that can meet the processing requirements. However, as with using a more advanced image sensor, such capability would come with a significant increase in power needs. This tradeoff makes higher-end processors generally infeasible for use with wearables.

If using a more powerful MCU is not an option, the most natural alternative is to trim the computational requirements by using a specialized model. For example, our original platform uses a neural network rather than a typical image processing pipeline, which greatly reduces the amount of computation. However, this often comes at the cost of robustness since such a one-size-fits-all model is not always able to deal with variations in illumination conditions and still achieve high accuracy.

4.2.3 Communication

Can we solve some of the computation issues by offloading it to the phone and cloud? The challenge is dealing with the power-efficiency of radios at high data rates — while radios like BLE, Zigbee, and even WiFi Direct are all relatively low power when used intermittently in a duty-cycled manner, continuous transfer at 30fps from a camera requires always-on radios and increases the power consumption to several tens or hundreds of milliwatts. However, offload has the benefit of being able to leverage vast computational resources to re-train models, adjust parameters, and deal with noise in sensor data, so when judiciously used, it can be effective.

4.2.4 Illumination

One of the key challenges in CIDER is how to deal with the peculiarities of indoor and outdoor lighting to enable robust estimation of eye parameters in both environments. Existing techniques rely either on natural illumination [91] or artificial illumination using a near infrared light source (e.g. [77]). Both techniques have significant advantages and drawbacks. Natural illumination has power savings since typical near-infrared LEDs consume many tens or hundreds of milliwatts during continuous operation. However, this technique is highly sensitive to ambient lighting, and has poor performance when operating in low lighting conditions such as while driving a car at night. In contrast, illuminating the eye via near-infrared (NIR) can provide higher signal to noise ratio, but NIR illumination does not work outdoors since sunlight has significant infrared content, which overwhelms the illumination from the NIR LEDs. Thus, the conditions observed by the IR camera vary significantly, and require more robust methods to process images and extract eye parameters.

4.2.5 Calibrating to new users

An under-explored challenge in designing eye trackers is how to deal with calibration to a new user. The training process is cumbersome and typically performed each time a user wears the device. This was also the case in our prior work, where the neural network model

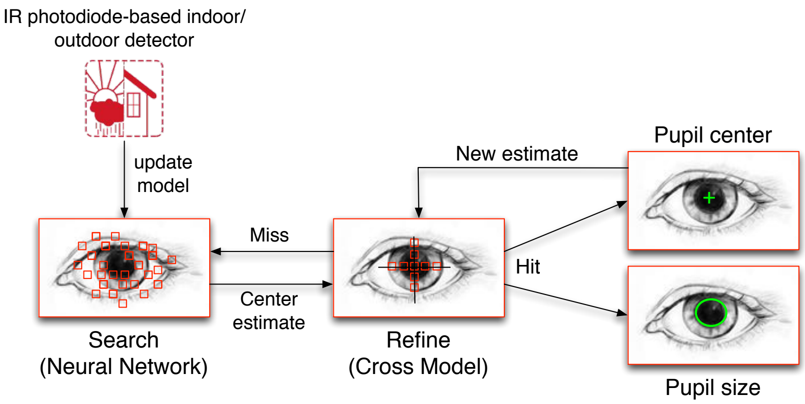


Figure 4.2: The CIDER pipeline: a) search stage using a neural network to get an initial estimate of pupil location, b) refine stage to zone in on exact pupil center and perform rapid tracking unless the pupil is missed, and c) NIR-photodiode-based detection of indoor/outdoor mode to update neural network model.

is learnt via a calibration procedure requiring the user to look at dots on a computer monitor before wearing the glasses. One question that we ask is whether we can train these systems for a new user with zero-effort i.e. no user involvement whatsoever. Such capability can enable us to train the system without asking the user to alter their normal behavior. This also opens up the possibility to automatically retrain the system when needed, thereby allowing greater flexibility in dealing with robustness issues.

In the following section, we outline a design that provides power efficiency while also creating robustness to variability.

4.3 CIDER Overview

At a high level, CIDER uses two different approaches to trade off between robustness and power — the first is a two-stage rapid eye tracking controller, and the second is indoor-outdoor model switching to deal with different illumination conditions and noise. Under typical indoor illumination, CIDER relies on a "Search–Refine" two-stage controller and a small amount of NIR illumination of the eye to estimate eye parameters in a fast, efficient, and accurate manner.

The search stage operates with no prior knowledge of pupil location, and uses a neural network to obtain an estimate of pupil center and size from a sub-sampling of pixels. The refine stage takes an estimate from the search stage, and uses a very fast and accurate procedure to locate and track the eye. When the refine stage loses track of the pupil due to specular reflections or other unpredictable variations, it reverts to the search stage to get another estimate of the eye location. The two stages differ in terms of the amount of sensing required (i.e. number of pixels acquired per frame from the imager) as well as the amount of computation performed to extract eye parameters from the pixels.

In outdoor settings, CIDER turns off NIR illumination (since there is too much ambient infrared), switches to different camera hardware parameters to deal with outdoor lighting, and switches the neural network model to one that is trained for outdoor conditions. We detect indoor vs outdoor conditions using an NIR photodiode that tracks the level of ambient infrared light. In the outdoor mode, CIDER does not use the refine stage and only relies on the neural network to track the eye. We generally find that there is significant variability, which makes it difficult for a more optimized model to operate in a reliable manner.

Overall our pipeline achieves a graceful tradeoff between robustness and power — under typical indoor illumination, CIDER spends most of its time in the fastest and most efficient stage while occasionally using the neural network to provide estimates. In outdoor illumination, CIDER spends all of its time in the slower but more robust neural network stage.

CIDER also addresses robustness issues by designing a model training pipeline that operates with no input from the user. Whenever the eyeglass is fully charged or has good connectivity, a block of images can be communicated to the phone, and a new model trained offline. The enabler is an offline image processing pipeline that generates accurate labels of pupil center and dilation from noisy image data (collected either indoors or outdoors), which makes it possible to learn the neural network models with zero effort from the user.

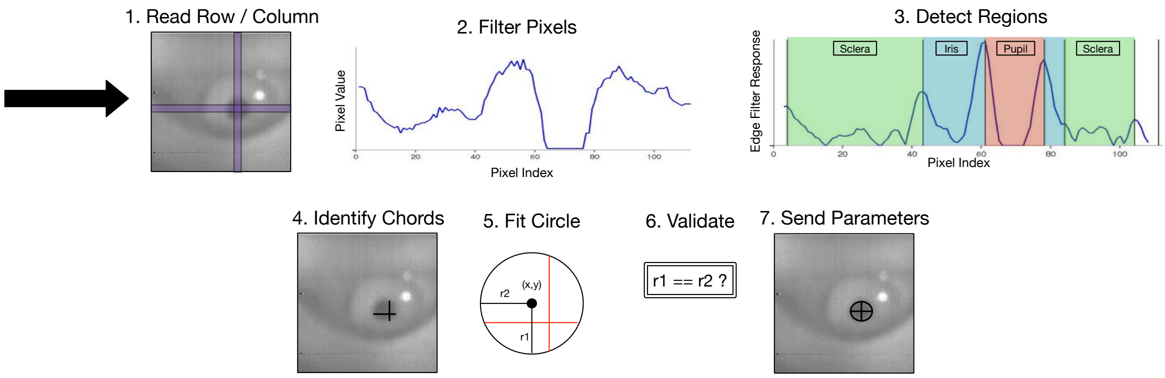


Figure 4.3: The CIDER cross-search model: 1) a row and column of pixels near the pupil center are sampled, 2) values are median filtered, 3) regions of the eye are detected using edge detection, 4) chords within the pupil are detected, 5) circle is fitted to the chords, 6) consistency check is performed to detect hit or miss, 7) the pupil center and size are estimated.

Finally, CIDER also improves on prior work in that we can simultaneously estimate pupil center and pupil dilation, thereby providing two key measures of the eye in real-time. In principle, CIDER's estimate of pupil center can be used to determine the gaze direction of the user by leveraging geometric mapping methods from the inward-facing image plane to outward-facing image plane as done in iGaze [91].

4.4 CIDER Design

In this section, we discuss the details of how CIDER works. We first discuss operation in the indoor case, for which CIDER is highly optimized, and then discuss how we handle the more variable outdoor case.

4.4.1 Search–Refine Controller

CIDER achieves high speed, high accuracy, and low power by using a rapid switching loop between the search and the refine stage.

4.4.1.1 Search stage – neural network model

The search stage is an artificial neural network (ANN) prediction model that operates over a subsampled set of pixels, based on the design outlined in the previous chapter. We provide a high-level overview for completeness, and refer the interested reader to the previous chapter for a more thorough overview. The process involves setting up the prediction problem as a neural network where the inputs are the pixel values obtained from the imager, and the output is a predicted (x,y) coordinate pair. The problem is set up as a bi-objective optimization, where one objective is to minimize the set of pixels that need to be sampled to reduce power consumption, and the second objective is to minimize loss in pupil center prediction accuracy. This is achieved using a neural network learning algorithm together with a regularizer that penalizes models that select more pixels. The optimization problem has two terms: a) an error term that captures how well the algorithm predicts gaze coordinates, and b) a penalty term that increases with the number of pixels selected. To promote sparsity i.e. to select a small active pixel set to sample, the algorithm uses a sparsity-inducing ℓ_1 regularization function, which minimizes the number of pixels sampled. The optimization function is solved offline using labeled training data, and the parameters are hard-coded into the eyeglass platform for real-time prediction of the pupil center.

The only major change to the ANN model for this work is the output target of the model. The goal of the original iShadow work was to predict the gaze location of the subject based on the orientation of their eye in the image. For this work, we instead want to identify the pupil within the eye image and report relevant parameters - center xy-coordinate, and radius of the shape. Since the original ANN model has several desirable properties, including input feature reduction and good accuracy, we chose to use the same model to estimate these parameters. The input to the neural network is still the subsampled pixels, however it is now trained to minimize error over three target values- center x, center y, and radius.

4.4.1.2 Refine stage – cross-search model

The refine stage is a cross-search model (shown in Figure 4.3) that leverages the estimate from the neural network to track the center of the pupil and pupil size with minimal sampling overhead. The first step of the cross model is to sample one row and one column of pixels at the estimated location provided by the neural network. There is substantial fixed pattern noise from our camera, so we need to first remove this noise (described in more detail in §4.5.2). Once the fixed pattern noise has been removed, the pixel values are median filtered and segmented into several regions — Sclera, Iris, Pupil, and Sclera. The segmentation process convolves the pixels with a box filter to detect edges. Since we run the same operations on a column and a row of pixels, we have two chords corresponding to the pupil along the vertical and horizontal axes. We assume that the pupil is a circle for simplicity of computation, and then it is straightforward to compute the center of the circle from the mid-point of the two chords.

4.4.1.3 Rapid switching between stages

Switching between the two stages works as follows. The ANN executes once, and then hands control over to the cross model to see if it can handle further refinements without requiring the ANN. The cross model is extremely fast, and takes a fraction of the time of the ANN, so it can execute quickly and check if further tracking of the eye can be handled entirely by using the cross model. To determine when to switch back, the cross model performs an internal validity check to see if the results it has obtained are consistent. Specifically, the cross model checks if the two chords (horizontal and vertical) result in a consistent solution. If there is too much error, it falls back to the ANN model. Since the cross model is fast, any misses are quickly handed by the ANN within a short time window, so the time window during which we do not have an estimate of the eye parameters is tiny.

The speed at which the cross model operates means that it is not only refining the estimate from the ANN, but is also tracking the eye. The cross model can operate at frame rates of several hundreds of Hz, which is much faster than the speed at which larger saccades occur. As a result, even if the eyeball is moving, the cross model makes small adjustments each frame, thereby tracking the eye. The only occasions when the cross model fails is when there are blinks, specular reflections, shadows, or other artifacts, in which case it switches to the neural network.

4.4.1.4 Optimizing NIR power consumption

One of the key enablers of the rapid switching controller described above is NIR-based illumination of the eye. Even though indoor lighting can vary significantly, there is virtually zero infrared content in the light emitted by lightbulbs (FL, CFL, LED, etc). This gives us an opportunity to use a small NIR light source to illuminate the eye, and use an NIR-pass filter on the camera to make sure that only the NIR illuminated content is captured. This gives us very controlled lighting conditions despite potential changes in the indoor lighting level.

One issue that we face is that typical NIR LEDs have high power consumption (the one we use consumes 180mW at peak power) — this is small compared to the overall power budget of typical eye trackers that consume watts of power, but it is exorbitant when we are attempting to operate at a few milliwatts of power. Thus, one question that we faced is how to reduce this power consumption.

There are two ways to reduce NIR power consumption — one is to duty-cycle the NIR photodiode, and the other is to reduce the operating voltage of the LED. NIR duty-cycling can be done between frames, therefore the reduction in number of pixels acquired using the cross-search model plays a significant role in the duty-cycling benefits. Reducing the operating voltage of the LED is effective as well — we found that NIR LEDs operate down to about 1.15V, and while reducing the voltage results in increased noise, there is sufficient signal for the neural network to learn a robust mapping. A small downside of this approach is that we lose some efficiency since NIR LEDs are typically most efficient at the high-end of their voltage range, however, this is balanced by the substantial power benefits that can be obtained. The combination of duty-cycling and low voltage operation reduces the NIR power budget by roughly two orders of magnitude, from 180mW to less than a milliwatt.

We note that our use of NIR is very different from methods used by commercial eye trackers. Typical eye trackers use multiple narrow NIR beams, and process the image data to locate these NIR beams, before combining this information with an eye model. However, this process requires several NIR LEDs, more complex geometric methods for estimating eye parameters, and does not generalize to outdoor settings. We operate with just two NIR LEDs, very simple models, and our computational methods continue to work in outdoor settings albeit at higher cost (as we describe below).

4.4.2 Indoor-Outdoor Model Switching

A second switching mechanism in CIDER is between indoor and outdoor modes of operation. Indoor and outdoor operation are very different for two reasons: a) NIR illumination is useful in indoor settings since it provides a controlled environment for eye tracking, but not for outdoor settings where there is too much ambient IR, and b) camera gain parameters need to be adjusted for outdoor settings and this requires modification of the neural network parameters.

Our idea is to track ambient IR conditions using a separate infrared photodiode that is built into our eyeglass (facing outward rather than inward). We use the IR levels to switch between different camera parameters (gain settings), as well as different neural networks trained for different conditions. This mechanism can be viewed as a camera gain control mechanism that is tightly integrated with the eye parameter estimation pipeline. Typical cameras use automated gain control (AGC) to deal with lighting variations, but a downside is that the pixel values are continually changing depending on the gain parameters. This makes it difficult to run a specialized computational function such as neural network, particularly when subsampling pixels to operate with as few pixels as possible. Rather than

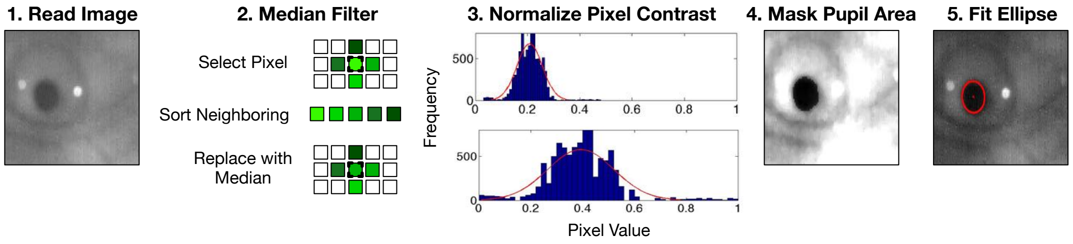


Figure 4.4: Labeling pipeline with select stages shown

continuous adjustments, our method can be viewed as a discrete approach, where we have two models corresponding to specific ambient IR settings, and we switch both the hardware parameters of the camera and the model based on the observed settings.

The switching process itself is extremely simple from the perspective of the firmware, requiring only a few MCU instructions to sample the photodiode at regular intervals. Since lighting conditions can be reasonably expected not to change with high frequency (i.e. more than once every few seconds), this sampling can be done as infrequently as once a second or less. If the MCU detects a significant change in lighting conditions, altering the camera gain parameters also only requires a small handful of instruction cycles. Thus, the overall power and time cost of the switching process is negligible.

This indoor–outdoor model switching can be viewed as a form of power vs robustness adaptation. In indoor settings, we consume more power due to NIR illumination of the eye, but save much more power by reducing the number of pixels sampled and associated computation. In outdoor settings, we shut off the NIR LED and opportunistically leverage ambient IR to save power. We rely on a more complex neural network model, which implies more pixels and more computation, but gain robustness in the process.

4.4.3 Zero-Effort Model Training

One remaining question in our system is how to train the models for each user. Ideally, we would want such training to be completely automated to minimize burden on the user. This problem often goes unaddressed in existing approaches, most of which are designed to require a period of explicit user participation in order to generate model training data. Zero-effort training could greatly increase the likelihood of broader applicability of our system.

The core question in training is how to develop a robust offline method for generating labels from noisy images collected by the camera. The offline procedure that we use for training the neural network is shown in Figure 4.4. The raw image is processed through a median filtering stage, from which the region corresponding to the eye is extracted. This region is further contrast-adjusted and filtered, and segmented to extract dark regions in the image. In good conditions, only the pupil shows up as a dark region, but we faced two additional challenges.

First, we see specular reflection of the NIR LED from the eye, and when the specular reflection overlaps with the pupil, the dark region can look like a disk, or like a disk with a bite on the side. To address this, we fill holes that we might observe in the segmented shape using standard image-fill techniques that identify distinctive regions of color within a larger area (the pupil) and adjust them using the surrounding pixels. Since the specular reflection is small relative to the size of the pupil, these simple techniques work extremely well in practice. Second, in outdoor conditions, we often see shadows caused by the sun's position relative to the eyeglass frame, and these shadow regions are also picked up by the segmentation block. To isolate the pupil, we look for the roundest segment to detect the pupil.

Given the target pupil location from the above image processing pipeline, we then divide the data into train and test sets and learn the neural network parameters. The new model is then uploaded to the glasses.

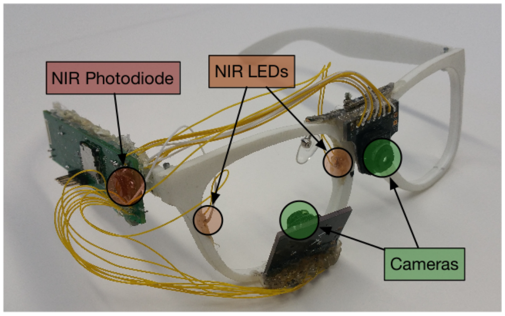


Figure 4.5: Eyeglass platform

4.5 CIDER System

In this section, we describe the main components of our eyeglass system and our implementation of CIDER, and the improvements over other prototypes that have been designed in past work.

4.5.1 CIDER Platform

Our eyeglass platform has a low-power camera that is mounted in the lower part of the frame facing the eye, as well as an NIR illuminator, as shown in Figure 4.5. We also have another outward-facing camera, as well as other sensors, but we do not discuss them in detail since they are not pertinent to the methods in this paper. We use the standard optics on the image sensors, which give a 36° field of view. The eye-facing camera has an NIR filter to capture the illuminated eye.

There are three major changes in the platform from the original design in chapter 3. First, we mount the eye-facing camera at the bottom of the frame rather than the top as in iShadow. The difference in how we mount the camera has implications on the tracking accuracy, as well as robustness to different conditions. One major consideration in this decision is that people naturally tend to look down with their eyes much more frequently than they look up. If the camera is mounted in the lower position we observed that, when looking down, the user's pupil is pointed nearly directly at the camera, making detection easier. In addition, when a person looks down for any reason, the upper eyelid naturally lowers a little. This can obscure the eye features when viewed from a higher vantage point (we have observed this in practice). Note that the lower eyelid does not noticeably raise when looking up, so the eye does not become obscured even when viewed from a lower angle. Thus, we concluded that mounting the camera on the lower portion of the frame was a strict improvement over the upper portion.

The second major change from iShadow is that we illuminate the eye with a pair of NIR LEDs shown in Figure 4.5. The placement point for the LEDs was chosen after careful characterization of what location would provide best illumination while minimizing issues due to specular reflections. Similarly, the choice of NIR LED was made after a rigorous measurement study involving more than a dozen types of NIR LEDs to understand their power-illumination profiles. The third and final difference is that we have an NIR photodiode that detects the level of ambient NIR and allows us to detect indoor conditions vs outdoor conditions.

Our platform is very different from other prototypes such as iGaze [91]. The iGaze device consumes more than a watt, and uses a Raspberry Pi attached to a glasses frame with cameras and sensors. The difference in power between our prototype and iGaze is between two and three orders of magnitude, and virtually every component in our system from algorithm to hardware components is optimized to achieve the power reduction.

4.5.1.1 Microcontroller

The iShadow platform uses an MCU with an ARM Cortex M3 core [22]. Our implementation of the platform uses an STM32L151 microcontroller, which is manufactured by STMicro Corporation [70] and is an implementation of the Cortex M3 standard. The STM32L1 family emphasizes low power consumption and includes a wide variety of processor sleep modes that are useful for reducing power draw by inserting timed sleep cycles where possible. It also includes several built-in peripherals for handling common communication protocols such as USB, reducing the firmware development burden significantly.

4.5.1.2 Image sensors

Our hardware framework is built around the Stonyman Vision Chip produced by Centeye, Inc.[19]

The Stonyman camera has a resolution of 112x112 pixels, each of which is characterized by a logarithmic voltage response to lighting conditions. These pixels have a high dynamic range, and more importantly, allow a random-access interface which the Stonyman provides via a register-based control scheme. Besides the extremely low power consumption compared to off-the-shelf cameras (3mW), the main advantage of the imager is that it allows for random access to individual pixel values. This feature allows us to subselect specific pixels that we need for CIDER, and results in significant reduction in the digitization cost.

Another important characteristic of the Stonyman imager that is the fact that the camera gain parameters are controlled programmatically rather than automatically (i.e. there is no automatic gain control like in other cameras). While this could be viewed as disadvantage, we find the ability to control gain to be beneficial for us in that we can adjust gain parameters and the model parameters in tandem when triggered by the NIR photodiode.

Finally, the Stonyman camera also provides features such as a sleep mode, during which the pixel acquisition circuitry can be powered down. The low-power state has power consumption less than half a microwatt since only a few control registers are running to maintain camera state.

4.5.2 Handling Camera Noise

We faced several implementation challenges, particularly in how we deal with the camera noise and identify camera gain parameters, which we briefly list in this section

4.5.2.1 Fixed pattern noise

One of the biggest challenges that we face in designing CIDER is dealing with low-level noise and the way in which the noise is intertwined with the hardware circuitry of the

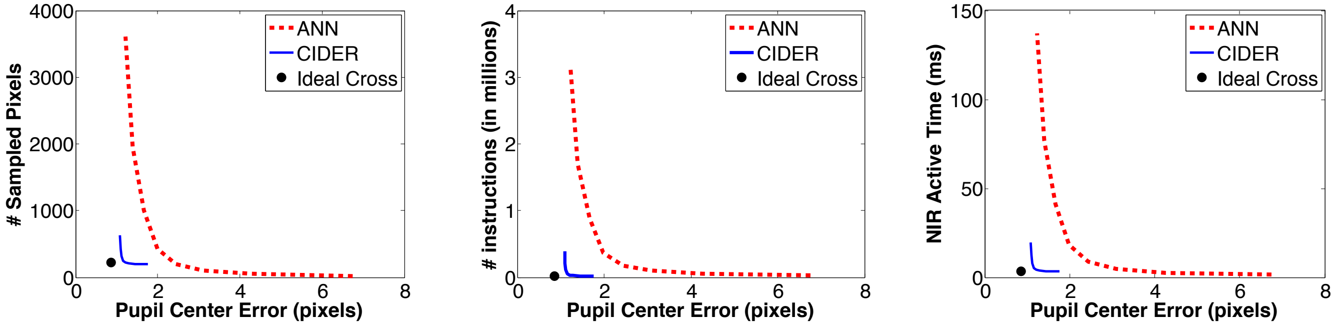
Stonyman camera. For example, one issue was that the fixed pattern noise of the pixels looked different when we were reading pixels along a horizontal line vs along a vertical line for the cross model. We identified that this issue was related to the way in which the pixel readout circuitry is designed on the Stonyman camera. The pixels along each row are daisy-chained to a single readout circuit, therefore, once we started reading out pixels from the beginning of the row, all the pixels along that row were activated. This resulted in varying noise along the row since the pixels at the end of the row had higher magnitude signal and noise. This issue does not occur when sampling pixels along a column since each only one pixel is read from each row. To address this problem, we learned a different fixed pattern noise mask per column and per row through offline calibration, and we subtracted the mask from the measured values to obtain the actual signal.

4.5.2.2 Gain settings

Another issue that we faced is that the Stonyman camera provides four gain parameters, each of which has roughly 20 settings. This results in a huge search space (20^4) to determine which is the best parameter setting for indoor and outdoor conditions. The search process is largely mechanical but time-consuming since an image has to be captured for each setting, and the values checked to see if it is appropriate. The setting is important, however, since indoor gain values simply do not work outdoors and result in the pixel values saturating. While the settings we identified works well under different outdoor conditions, it may be possible to perform further fine-tuning to specific conditions and get better results than those which we have reported.

4.6 Evaluation

We first describe the datasets that we have collected and the evaluation metrics we use and then describe our experimental evaluation.



(a) Sensing pixels vs center accuracy (b) Computation cycles vs center accuracy (c) NIR time vs center accuracy

Figure 4.6: Cost vs Accuracy

4.6.1 Datasets and Ground Truth Labeling

We evaluate CIDER with four datasets that correspond to different environments and dynamics. All data collection experiments involving human subjects received approval from an institutional review board.

- **Indoor-Stable data (fixed pupil, fixed illumination)** We collected data from 16 users, 12 male and 4 female. Each subject performed a video calibration routine where they looked at a high contrast dot moving on a computer monitor for several minutes. This gives us good coverage of eye positions, and allows us to train a good model as well as determine robustness to position of the eye. The illumination was held constant during this period, and subjects' pupils were roughly 5–7 pixels wide in this illumination. We generated approximately 2500 eye images for each user. The subjects involved in the data collection represent a range of ethnic groups with different eye shapes and iris colorations. We refer to this dataset as `indoor-stable`. All subjects in the other datasets were also in the `indoor-stable` dataset.

- **Indoor-Variable data (variable pupil, variable illumination)** We collected this data for 14 users, 10 male and 4 female. We varied the lighting conditions in five discrete levels using a combination of different sets of ceiling lights as well as target spotlights. The subjects pupils dilated between 5–15 pixels during this period, which gives us a fairly large range of pupil dilations that is representative of what one would observe in real-world settings. The screen brightness was kept low enough to not impact dilation much. The above computer-based calibration routine was executed for each setting to obtain data. We refer to this dataset as `indoor-variable`.

- **Outdoor data (uncontrolled illumination)** We collected this data for three users, all male, under outdoor settings. The conditions were generally bright. We obtained several minutes of data from each participant generally gazing at the outside scene under different orientations.

- **Indoor-Outdoor switching data** Our indoor-outdoor data was collected for one user, who walked between indoor and outdoor conditions repeatedly for four iterations, while spending roughly a minute in each environment. This dynamic setting helps us evaluate whether the NIR photodiode-based model switching algorithm works effectively with real scene changes.

4.6.1.1 Ground truth labeling

All data collected above was labeled for pupil center and pupil size using the process described in §4.4.3. Once labeled, we trained the neural network to identify the pupil center and radius of the best-fit circle approximating the pupil shape using a standard five-fold cross-validation scheme. We averaged the test set error over the five folds to get an average score. Pupil center error is computed as the L2 (Euclidean) distance between the estimate and the label, pupil size error as the difference between the estimated radius and the label. The errors were averaged over all subjects per model size to get a final set of error estimation accuracies over a range of neural network model sizes.

4.6.2 Evaluation Metrics

We use several performance metrics to evaluate our system. Since our power numbers are specific to our platform, we provide both a more general metric that could generalize to any platform as well as a more specific metric for our platform given the hardware components that we choose.

4.6.2.1 Cost metrics

We use three performance metrics to evaluate our system.

- **Sensing cost** We measure sensing cost in two ways: a) the number of pixels subsampled from the imager, and b) the power consumed for sampling the pixels for the Stonyman camera. The former measure generalizes to any camera that can be subsampled, while the latter measure provides a real measurement that includes constant overheads of switching the camera from sleep to active mode, sampling the pixels, and switching back to sleep mode.

- **Computation cost** We measure computation cost in two ways as well: a) the number of instructions that need to be executed for each model, and b) the power consumed for executing instructions.

- **NIR cost** Similar to the above two metrics, we measure the NIR cost in terms of active time (i.e. time for which the NIR is turned on), as well as the power consumed for our NIR LED with duty-cycling and voltage optimizations described in §4.2.

4.6.2.2 Accuracy metrics

We measure accuracy of estimating eye parameters using two metrics

- **Pupil center** The accuracy in measuring pupil center is measured in pixels in the image captured by the eye-facing imager. This measure gives us an idea of how far we are from the best-case performance given the sampling granularity of the imager. From an application perspective, the key metric of interest is the degree error in estimating gaze. We estimate that each pixel corresponds to roughly 0.3°, so this gives us a mapping from pupil center error measured in pixels to gaze error. Commercial (tethered) gaze trackers achieve errors of roughly 0.5° [85], which is slightly less than two pixel error on our system.

- **Pupil radius** Similar to pupil center, we measure pupil dilation in pixels. Each pixel in in the camera's visual field corresponds to roughly 0.22mm when measured on the pupil, which is similar to the resolution of high-end gaze trackers.

4.6.3 CIDER Performance

As we have outlined, the search and refine stages of CIDER are intended to maximize estimation accuracy and power efficiency over a range of environmental parameters. Our first set of results evaluates the performance of CIDER by comparing it against the two stages (search and refine) independently. We use the `indoor- stable` data in this evaluation, which gives us an understanding of best case performance under limited dynamics. We compare the following schemes in this evaluation:

1. **Neural network** The neural network model is learnt as described in §4.4.1 — we vary λ (regularization parameter) to learn various models that have different tradeoffs between accuracy and pixels (which translates to power). This gives us a pareto optimal set of solutions i.e. a set of solutions that shows the tradeoff between the two objectives.

2. **Idealized cross** The idealized cross method is initialized by the pupil center estimated by our offline algorithm. The cross model then estimates the pupil center and pupil size, and we compare the accuracy against ground truth. Clearly, this is an idealized scenario where the cross model should perform very well, but it still helps us understand how well the edge detection and parameter estimation methods work in the best case.

3. **CIDER** The CIDER method is the fast switching technique. Since the CIDER pipeline involves switching between the ANN and cross model, we expect performance to be in-between the above models.

4.6.3.1 Sensing, computation and NIR cost

Figure 4.6a shows the curve of sensing cost (in number of pixels sampled) against pupil center estimation error (in number of pixels). The curve is obtained by tuning the neural network regularization parameter, which allows for the generation of a number of network sizes with varying power needs and corresponding accuracy. The result clearly shows that there is a significant gap between any pareto optimal solution that can be obtained for the neural network vs the solution provided by the idealized cross model. CIDER operates between the two but closer to the idealized cross model. This can be explained by the fact that the neural network is triggered only about 10-15% of the time whereas the cross model operates the remaining 85-90% of the time.

The performance difference in terms of computation cost is substantial as well, in fact, even more than in the case of sensing (Figure 4.6b). The neural network computation is much more involved than the cross model, so there are significantly more operations per pixel. In addition, since the cross model requires fewer pixels, the number of times the computation needs to be performed is also much lower. Thus, the number of instructions that need to be computed for the cross model is orders of magnitude lower than for the neural network.

Finally, the time spent with the NIR LED on is also substantially lower for the idealized cross and CIDER models (Figure 4.6c). Since the cross model needs very little time to sense, the NIR LED needs to be turned on for a minuscule amount of time for each frame.

4.6.3.2 Energy savings

We now look at how the benefits in sensing, computation and NIR translate into energy savings on our platform. We measure the average power over a 10 second window of

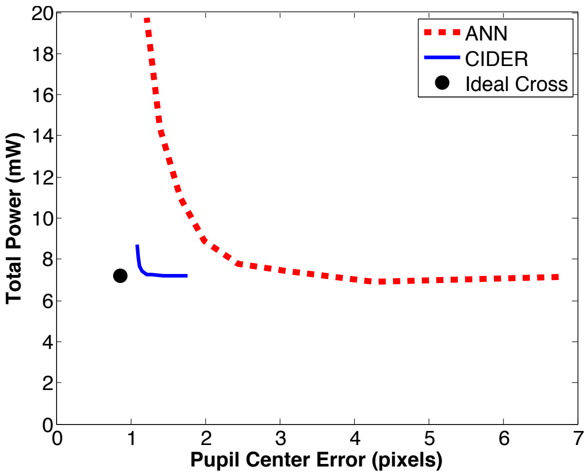


Figure 4.7: Aggregate power vs accuracy

operation using a DAQ running at a 10 kHz sampling rate. To measure power consumption for all three models, we fix the pixel capture + predict rate of the system to 4 Hz by inserting MCU sleep periods as needed. The 4Hz rate is chosen to enable us to measure a sufficiently large range of neural network model sizes to plot the pareto optimal graph.

Figure 4.7 shows the aggregate power consumption of CIDER and compares against the two other baselines. We see similar trends as we saw earlier in that CIDER operates in between the idealized cross and ANN model with roughly a 3× reduction (compared to neural network models that have low error). The overall power budget for CIDER is roughly 7mW, which is a huge improvement over state-of-art (order of magnitude less power consumption than the original design), and a substantial achievement considering that the system is operating a camera, estimation algorithm, and NIR LED.

One curious feature of the graph is that the baseline for all schemes is shifted by about 6mW. The baseline shift corresponds to constant overheads incurred by our platform, and for configuring various parameters for the camera upon wakeup and shutdown. We suspect that there are various sources of power leakage that contribute significantly to the baseline, but we have not yet been able to fully debug these issues. Looking forward, we expect that

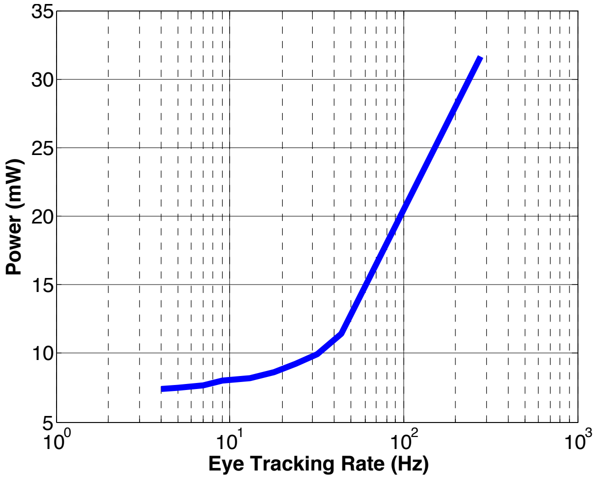


Figure 4.8: Aggregate power vs eye tracking rate (log scale)

some of this constant overhead can be eliminated with a more optimized computational block such as an FPGA rather than a general-purpose MCU.

4.6.3.3 Power vs tracking rate

Another benefit of CIDER is that it can achieve high tracking rates. We plot the power vs pupil tracking rate in Figure 4.8, which shows the total system power consumed as the tracking rate is varied. To generate this graph, we used the same model as was used for the measurements in Table 4.1, and inserted sleep periods of variable length between each single execution of the CIDER pipeline. The measurements were again taken using a DAQ sampling at 10kHz.

Component	**Power (4 Hz)**	**Power (278 Hz)**
Camera	7.30 μW	30.8 μW
MCU (digitization)	2.67 mW	11.3 mW
MCU (computation)	4.79 mW	20.2 mW
NIR	8.24 μW	34.8 μW
Overall	7.48 mW	31.6 mW

Table 4.1: CIDER power breakdown

Table 4.1 shows a finer-grained breakdown of the power vs tracking rate for each component of CIDER (with a moderately large neural network chosen to use 10% of the pixels). We give two power measurements - one taken at the maximum eye tracking rate possible for this model size, namely, 278 Hz, and one taken at the 4Hz rate used for the rest of the evaluation results. There are several useful observations that can be made from this result. Interestingly, the camera and NIR consume virtually no power compared to other components since they are turned on for a very tiny amount of time. The acquisition consumes a significant amount of power — this is because digitization of the analog signal output from the camera is expensive. One of the major improvements that CIDER provides is reduction of the digitization overhead. The MCU computation is also expensive, however some of this cost could be reduced by using a more optimized computation block such as an FPGA.

4.6.3.4 Estimation accuracy

The above discussions emphasize power consumption, but it is instructive to look at the absolute accuracies that can be achieved by CIDER. The above results show that CIDER achieves pupil center estimation accuracy within 1.2 pixels. The neural network method cannot achieve such accurate estimation even when consuming considerably more power and resources.

This result may seem surprising at first, since it is natural to expect a more power-hungry technique to have a corresponding increase in performance. The main reason is that the NIR-illuminated eye (indoors) presents very strong edges that are easier to accurately identify using edge detection techniques (the cross model) than using a neural network. So, the accuracies tend to be higher for CIDER even though the power consumption is much lower. This is not the case in the outdoor environment, however, hence the need for the indoor-outdoor switching model. Thus, not only are we able to achieve substantially reduced power consumption, we also do so while simultaneously improving accuracy to within a small amount of the lower bound of what is achievable with our camera.

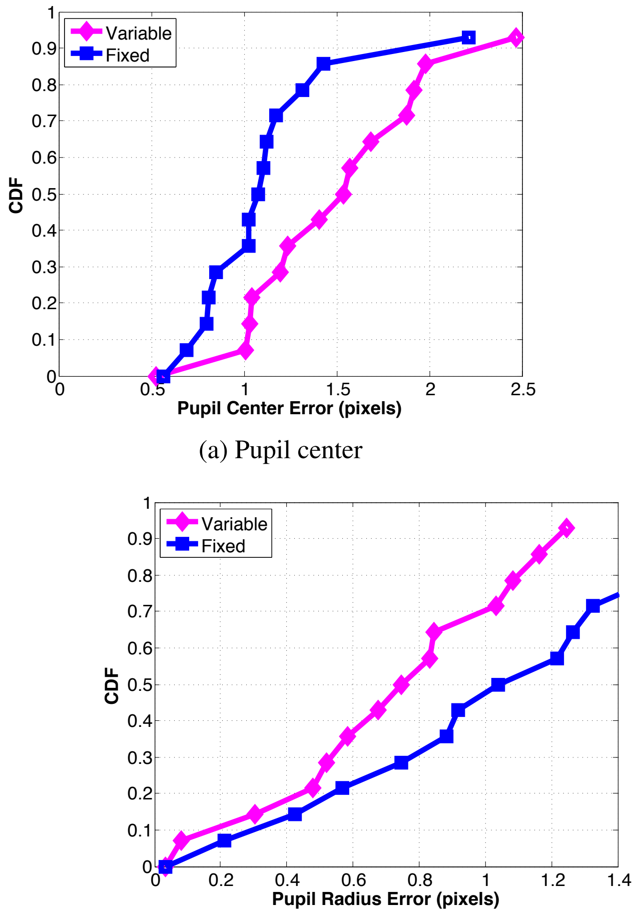
Table 4.2 shows the results for pupil size estimation when using only the neural network and when using CIDER. We do not show the entire power-accuracy profile for pupil size since we find that even the smaller ANN models perform well in estimating the pupil size, and there is not much difference in using a larger model. So, we present only the mean performance across all model sizes. We see that the pupil size estimation error is typically less than one pixel, which suggests that both stages can do an excellent job in estimating pupil size. Indeed, we find that the error for CIDER may be over-estimated since we often see that the cross model's estimates are closer to the real value than even ground truth labels.

4.6.4 CIDER Under Variable Conditions

Having evaluated CIDER under relatively stable conditions, we turn to situations that have more variability. Specifically, we look at three cases: a) variability in the pupil dilation of the user, b) an outdoor setting with variable illumination, and c) the user moving from an indoor to an outdoor setting.

4.6.4.1 Variable pupil dilation

The results in §4.6.3 were taken under fixed pupil dilation, so one question is whether the results are robust to varying pupil sizes. Figure 4.9 compares the pupil center and pupil size estimation errors of CIDER for the 14 users in `indoor-variable`, all of whom are also in the `indoor-stable` dataset. Figure 4.9a compares the pupil center prediction results for fixed and variable illumination conditions, each as an error CDF, and Figure 4.9b gives the same comparison for size prediction. The center prediction accuracy under varying pixel sizes is marginally worse than the accuracy under fixed pixel sizes, but the difference is not significant. For the size prediction task, CIDER actually generated slightly better estimates on the variable pupil size dataset. This seems counter-intuitive at first, however, in the `indoor-variable` dataset, the pupil size is generally larger than in `indoor-stable`, as the lighting conditions were darker for most of the experiment. This makes accurate



(a) Pupil center

(b) Pupil size

Figure 4.9: Performance comparison - fixed and variable pupil size

detection of the size slightly easier for both the ANN and the cross model. Overall, we see that performance of CIDER is robust to variation in pupil size.

4.6.4.2 Outdoor dataset

The outdoor scenario represents another high variability situation for CIDER. The cross model does not work in this situation, so the system relies primarily on the neural network that is trained for outdoor settings. We find that the accuracy with CIDER under outdoor settings is roughly 4 pixels (for moderately sized ANNs). The results are worse than accuracy in indoor settings, but not far off. In fact, the accuracy that we obtain in outdoor settings is better than the results that were obtained in the previous chapter under indoor

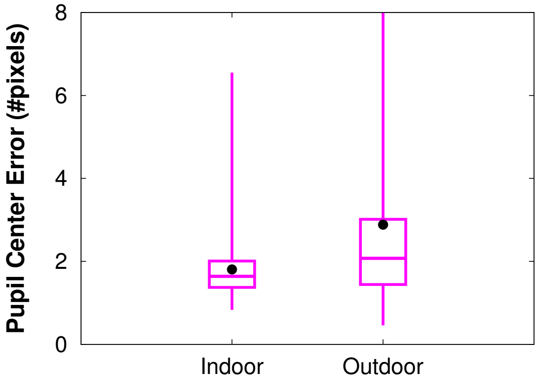


Figure 4.10: Indoor-Outdoor switching

settings. One of the main reasons for the performance difference is the vastly improved labeling pipeline that we have developed, which allows us to label noisy data quite well.

We get about 1 pixel pupil dilation error, but we find that that this is an over-estimate of the real error for reasons described above. There is about a 1 pixel offset between the radius estimated by the offline labeling algorithm (which performs filtering), and by the cross model. For transparency, we have reported the error as observed, but we think the error is about one pixel smaller than that reported.

4.6.4.3 Indoor-Outdoor switching

We now look at a situation where a user is moving between an indoor and outdoor environment, and show how well our IR photodiode-based model switching performs. Figure 4.10 shows the error distribution during the indoor segments vs outdoor segments. This is shown as a box plot, where the three lines in the box corresponds to the quartiles (25 percentile, median, and 75 percentile), the whiskers correspond to the max and min, and the dot is the mean. We truncate the max error whisker for the outdoor case since there are some cases where CIDER returns more than 10 pixel error.

We also verified from the traces that the NIR-based switching works effectively, and switches models between the indoor and outdoor modes whenever the user changes environments. As observed in §4.4.2, the instruction cycle and power cost of the detection and switching process itself is negligible. The error distribution of the predictions is higher for the outdoor case, but it is still relatively low with a mean of less than three pixels. The error when indoors is lower with a mean of less than two pixels.

4.6.5 CIDER High-Speed Eye Tracking

One of the major benefits of CIDER is the eye tracking speeds that it can achieve. High-speed eye tracking is useful for understanding fast saccadic movements of the eye, which is one of the neural mechanisms for maintaining visibility. For example, one of the interesting use-cases for measuring micro saccades is as a diagnostic test for ADHD [31], and there are other applications of such measurements [53].

However, high speed eye tracking is also very challenging on a wearable device. Commercial high-speed eye trackers achieve several hundred hertz tracking rates (e.g. the Eyelink high-speed eye tracker samples at 500Hz, and the ASL H7-HS can sample at rates up to 360Hz). Of course, these eye trackers are also bulky, tethered for power and connected to a computer for data storage and processing. One interesting question is, how fast CIDER can operate if it is not duty-cycled and is allowed to perform pupil estimation as fast as possible?

To evaluate the maximum speed achievable by CIDER, we run it continuously on our eyeglass without duty-cycling. We measure the rate at which it generates pupil center measures, and find that CIDER achieves frame rates of 250–350 Hz (depending on whether a medium-sized or small ANN is used). These speeds are comparable to the rates achieved by high-speed eye trackers. One caveat is that CIDER is not uniformly sampling since it occasionally uses the ANN. However, the irregularity during the use of ANN can be mitigated by using a smaller ANN model. The power consumption at this frame rate is several tens of milliwatts since the system is operating in always-ON mode. Therefore, many of the optimizations that we used earlier no longer work. However, we don't anticipate that the high-speed mode will be used continuously; rather, this mode may be triggered when appropriate. Overall, we think that the ability to sample at high speed has substantial implications for a wide range of health and cognition applications.

4.6.6 Accuracy of Labeling

To evaluate the accuracy of the labeling scheme described in §4.4.3, we hand-labeled 100 eye images from one subject's data. For each image, we selected an elliptical region that visually seemed to best fit the pupil area. We then compared the pupil center and size estimate with those provided by the automatic labeling system for the same frames. The results are given in Table 4.3. Note that for both measures, the hand-labeling and automatic labeling techniques yield very similar results. The pupil size is slightly higher, but this is most likely due to the fact that the low-resolution images do no provide as sharp of an edge as would be expected with a higher-resolution camera. Thus, the pupil edge appears spread over a one- to two-pixel area, and distinguishing the exact pupil boundary within that region is difficult for a human to do visually.

4.7 Discussion

To close, we discuss in this section some avenues of future work that we have not addressed in this paper.

4.7.1 Adaptation to dynamics

While our zero-effort pupil-labeling approach tackles the initial calibration problem, one question that we have not addressed is how to deal with dynamics in eyeglass positioning or the environment. For example, shifts in the position of the glasses relative to the user's eye would increase error in ANN predictions. Similarly, prediction error would also increase if the ambient infrared in outdoor settings is significantly different from the data used in training the ANN model. To address such dynamics, we need approaches to detect in real-time that the existing model is performing less accurately than expected, and dynamically adjust the model, perhaps by leveraging our zero-effort training procedures described in this paper. We are exploring techniques for dynamic re-calibration in ongoing work.

4.7.2 Form-factor

The form-factor of the eyeglass is a particularly important problem to tackle for more widespread use of such devices. The biggest challenge in the design is finding an unobtrusive placement of the cameras while simultaneously achieving good coverage of the eye to enable robust estimation of eye parameters. Even with our current prototype, there were instances where we could not obtain complete coverage of the eye due to differences in face shape and the limited field-of-view of the camera. This problem is exacerbated when the camera needs to be embedded in the eyeglass frame since the positioning of the camera is closer to the eye and more sensitive to changes in placement. One of the questions that we are currently exploring is how to embed the cameras in the eyeglass frame so as to reduce form-factor without sacrificing accuracy.

4.8 Conclusions

In summary, this paper describes a new method, CIDER, for estimating eye parameters on a computational eyeglass using a staged architecture that trades off power for robustness. Our architecture uses an optimized detector for the "common case" involving a user being indoors and in limited-noise settings, and tremendously reduces the overall power consumption down to numbers that are within the range of typical wearable devices. CIDER deals with more noise and variable illumination settings by using more computational and sensing heft to filter out noise and deal with variability. Finally, we achieve very high frame rates, which gives us the ability to sense fine-grained eye parameters. Most surprisingly, we enable all of this functionally while operating on the same small ARM Cortex M3 micro controller as used in the original iShadow work.

Model	**Pupil Size Error (pixels)**
Neural Network	0.50
CIDER	0.85

Table 4.2: Pupil radius estimation accuracy of CIDER

Feature	**Mean Difference (pixels)**
Pupil Center	0.853
Pupil Size	1.52

Table 4.3: Automatic labeling vs hand labeling of pupil

CHAPTER 5

ENGINEERING CONSIDERATIONS FOR PRACTICAL APPLICATIONS

The previous two chapters of this thesis have discussed the technical innovations required to realize our goal of a lightweight wearable eye tracking device. At the conclusion of the previous chapter, we had developed the hardware and algorithm to the point that the fundamental performance requirements had been met. However, one crucial element of the design remains unaddressed — specifically, the overall visible form factor of the device itself. The primary motivating goal for this thesis is to design a device that could be used for in-the-wild studies. This goal motivates all of the prior technical innovations discussed, but if subjects are unwilling to use the device or it interferes with their daily behaviors, then the technical achievements will have been hamstrung.

The problem of making the glasses less obtrusive and therefore more usable, however, is not simply a matter of creating a more attractive glasses frame design. There is a fundamental design tension regarding the placement of the eye-facing camera, which is that a favorable camera position for eye tracking is very obtrusive for the subject. Finding a middle point that trades off well against these two considerations is a significant challenge, and in this chapter we present several possibilities that we have explored.

We first present a few initial prototype ideas, followed by a working design solution that appears to optimally trade off user comfort and convenience and against the technical requirements of the system, thus fully realizing the wearable eye tracker design goals we have laid out in chapter 1. It has two slightly different implementations with varying properties, with preference likely decided by the needs of the specific application. We present a comparison of eye tracking results between this design and the CIDER iteration of the platform.

To conclude, we discuss a design for an even more advanced version of the system that uses flexible PCB material to integrate the cameras completely with the frame. This design has several desirable properties but proved to be a more serious engineering challenge to implement than the benefits merited. We give an overview of it here and describe the remaining open design challenges for integrating it into the iShadow platform.

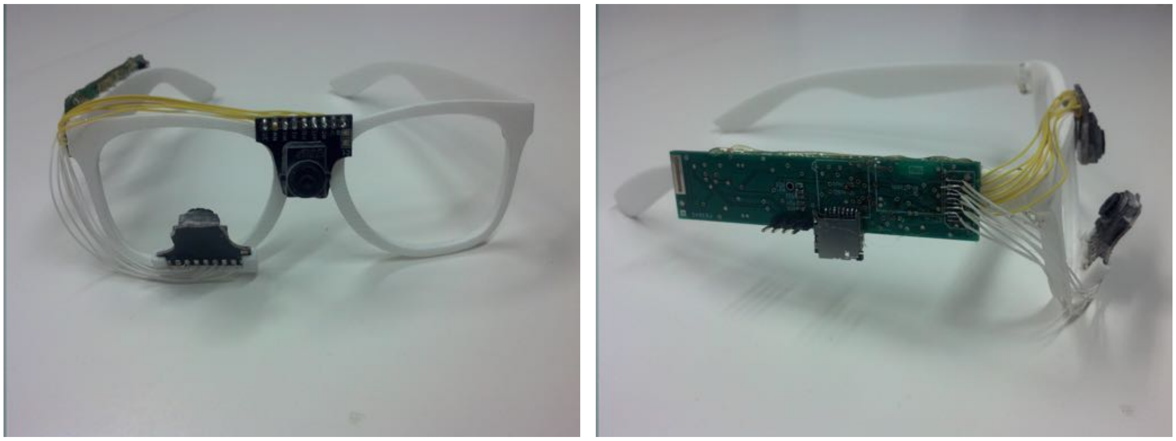
5.1 Background and Motivation

The final contribution of this thesis is the last step towards making the iShadow platform feasible as a device for in-the-wild studies. As was stated in chapter 1, this platform must be something that subjects can wear over the course of their daily life with little to no intrusion. This is a critical design goal because the target applications for this platform — mental health metrics, social context awareness, cognitive state and fatigue monitoring, to name a few — require that the subject be wearing the iShadow glasses for long periods in conjunction with their regular activities of living.

Recall the design criteria for this, from chapter 1:

1. Can be worn for long periods without physical discomfort
2. Will not inflict mental burden on the user — i.e., does not require regular attention to operate
3. Does not obstruct the user's vision or in any way impair their ability to do their daily tasks
4. Appears "normal" and will not draw attention from other people or become an eyesore to the wearer

These criteria are critical because without them, either (a) the subject will stop using the device due to inconvenience, irritation, or social pressure, or (b) they will use it but the



(a) Front View

(b) Side View

Figure 5.1: Positioning of the camera in the CIDER frames. Note that it clearly obstructs a portion of the eye's view.

data will become polluted due to bias introduced by the presence of the device itself. Of all of these, criteria 3 and 4 are the only outstanding issues at the conclusion of the previous chapter. Criteria 1 is addressed basically automatically, since most people are comfortable wearing glasses for long periods of time. The second criteria has been discussed as part of the previous chapters on the development of the algorithm — there is no need for the wearer to do anything extra once calibration is completed.

The last two criteria jointly point to issues primarily related to the placement of the cameras within the glasses frame. The cameras are the only component that needs to be near the eye and therefore may obstruct the wearer's vision, as in the CIDER iteration of the hardware (figure 5.1a). In the same vein, they are the only component that may require changes to the shape and structure of the glasses in order to be positioned properly. It is true that there are additional components, specifically the PCB and battery, which have to be accommodated, but devices such as Google Glass and JINS MEME [39] have demonstrated that it is feasible to wrap these inside of the "arms" of a glasses frame in a very subtle way.

Thus, one primary design goal is for the cameras to be placed in such a way as to be minimally noticeable or obtrusive. However, it is equally important that the cameras be placed in such a way as to have good visibility of the eye region across all possible wearers. As discussed in chapter 1, being able to see the entire eye is obviously crucial for maintaining good eye tracking fidelity. There is a great deal of variation in the shape of the eye region and nose between subjects, which can dramatically change the position and angle of the eye relative to the camera (see figure 3.6), therefore it is important that the eye-facing camera be placed in such a way as to have a wide view covering the face. As you can see in figure 5.1b, the way we accomplished this in CIDER was to set the camera far back and slightly below the eye facing up.

5.1.1 Contributions

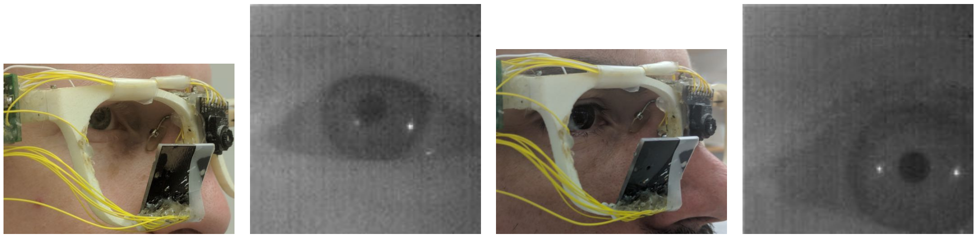
Both of these goals, unobtrusive camera placement and consistently good view of the eye, are critical for the utility of this platform in terms of in-situ studies on daily living. The focus of this chapter is a hardware redesign that addresses both of these major design concerns simultaneously. The overall contribution is a realization of the iShadow / CIDER system as a wearable device feasible for use in daily life environments. This includes concrete hardware components as well as more general methodologies which we used to guide the design, which can also be used for integrating the system into other glasses frame designs so as not to limit it to this particular incarnation. We break the contribution into multiple components:

- Design involving two cameras placed within the glasses frames so as to ensure good capture of the eye across virtually all subjects
- Glasses frame shape that confirms to a standard design so as to appear natural and be comfortable to wear
- Two different types of redesigned camera breakouts, each providing a separate control strategy for the eye cameras depending on application needs

- Redesigned lens housings which facilitate smaller and more modular camera hardware while providing a sufficiently wide viewing angle
- Strategy for placing the cameras and LEDs within the glasses frame so as to achieve a usable image across a very wide range of face and eye shapes

While none of these contributions are primarily advancing the fundamental research questions, they are nonetheless a critical step towards validating the effectiveness of the prior work. The thesis statement of this work is entirely focused on making eye tracking usable and practical under the design constraints of a wearable device. Therefore, until the final application has actually been tested in the field, there is no way of knowing whether those constraints have truly been met or if the work just represents a stepping stone in that direction. This chapter demonstrates that, while the last steps towards the goal are not fundamentally innovative, they are also not nontrivially challenging and require serious design and implementation effort.

In addition to the above contributions, we demonstrate the versatility and effectiveness of the sparse-sampling neural network model discussed in chapter 3. We used it for testing and evaluating the effectiveness of every camera configuration we attempted, and discovered that its performance is surprisingly consistent across camera placements so long as the majority of the eye is visible in the field of view. This was surprising to us because many of our configurations distorted the eye image in some way. For example, in many configurations the eye was not in clear focus, and in others the camera was set further away so that the eye took up a relatively small proportion of the image. However, in our single-subject tests we found that these factors had no significant impact on the neural network's performance. We discuss these results and the implications about the neural network's performance.



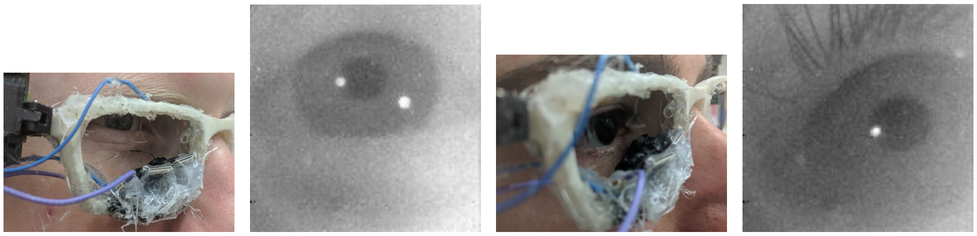
(a) Subject 1, CIDER - Position (b) Subject 1, CIDER - Image (c) Subject 2, CIDER - Position (d) Subject 2, CIDER - Image

Figure 5.2: Comparison of two subjects wearing the CIDER frames, looking at position relative to the eye and the corresponding image. For both subjects, the results are good.

5.2 Challenge: Competing Design Goals

Probably the most serious challenge for this work, and the reason that it is still something of an open question in the wearable eye tracking community, is that the two design goals we have laid out are in direct competition with each other. The intuition for this is simple: the easiest way to make the device design unobtrusive is to move all of the components directly into the existing frame, which means placing the camera component(s) in particular inside the frame near the eye. However, the easiest solution for achieving a consistently clear view of the eye region is to set the camera further back from the eye so as to maximize the area covered by the lens' angle of view, as done in the CIDER platform. Moving the camera into the frames places it much closer to the eye and at a much steeper angle of view, meaning that it is much more sensitive to position (which varies depending on the wearer's face shape) and that distortions of the eye view are much more likely.

We give a practical example of this issue using our hardware in figures 5.2 and 5.3, which shows images from both the CIDER hardware and the a later prototype that we developed with the camera mounted to the inside edge of the frames. We consider the case of two reference individuals whose face shapes lead to extremely different camera positions relative to the eye: on one subject the frame sits lower and further back, on the other, it sits closer and further forward, as shown. We show each camera setup (CIDER and an in-frame prototype) on both subjects, and a sample image from each. You can see that the CIDER



(a) Subject 1, revised - Position (b) Subject 1, revised - Image (c) Subject 2, revised - Position (d) Subject 2, revised - Image

Figure 5.3: Same comparison using a newer frame with a camera mounted inside. For subject 1 the eye is clearly visible, however, for subject 2 the eye is not at all visible and the lens is uncomfortably close to the eyeball.

camera positioning, while unappealing and visually distracting, achieves a good eye image for both subjects; the in-frame camera, however, cannot see one subject's eye at all while the other is in a relatively optimal position.

This example effectively illustrates the tension between the two major design goals: an "attractive" arrangement of the cameras dramatically reduces eye tracking utility. Finding a balanced tradeoff point between these two goals is a core design challenge of all wearable eye tracking devices, and our search for that optimal point is the primary discussion of this chapter.

As a brief aside, there are two other minor design constraints that are not of primary concern but worth mentioning nonetheless. First, we note that the camera must always be mounted on the lower part of the frame because the upper eyelid moves as the eye rotates vertically and often occludes the eyeball when viewed from above. Second, having the cameras mounted on the outside part of the frame means that prescription or sunglasses lenses cannot be used in the frames as they would likely introduce distortions to the cameras. Again, neither of these are primary motivators for this problem, but they do inform its solution somewhat.

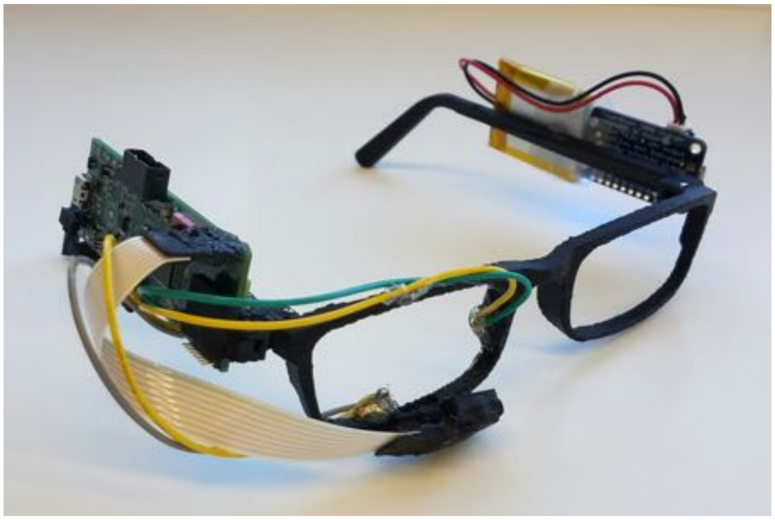


Figure 5.4: Refined version of CIDER design

5.3 Design Iteration 1: Refinements on CIDER Design

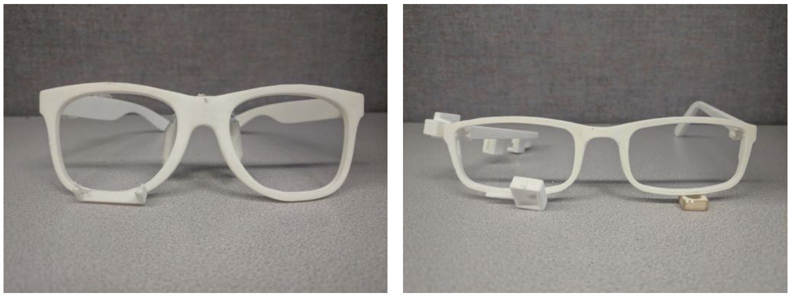
Our first step towards a more usable version of the frame developed for the CIDER work was not a full attempt at a solution to the above challenges; rather, it was an experimental attempt to integrate several new ideas which would later be essential to the finished work. Figure 5.4 shows the final product of this design pass. We discuss the following design changes which were explored with this iteration:

1. Change the frame shape to align more closely with standard glasses frames
2. Use a miniaturized camera breakout PCB
3. Integrate a smaller lens with a wider viewing angle

The frame that was developed as a result of this pass was used in a separate research effort to measure blink rates as an indicator of fatigue, primarily for driving safety [62].

5.3.1 Altered Frame Design

Starting with the CIDER work, all of the frames used for the iShadow glasses have been 3D printed in order to both make construction of the devices repeatable and to facilitate experimentation with different layouts. The CIDER frames, absent any electronic



(a) CIDER (b) New Design

Figure 5.5: Comparison of 3D printed frames for CIDER and the refined version

components, and the frames produced in this iteration and henceforth are compared in figure 5.5. The major difference is the shape of the eyepieces - during the work for CIDER the emphasis was on clean data collection, so the frames were deliberately designed to facilitate the placement of a set back several centimeters from the eye at a slight angle below it. This was also accounting for the specific size of the breakout boards used for the Stonyman up until that point. Obviously these frames are very distinctive and dissimilar from standard eyewear.

The new frames are based on standard-issue military glasses frames (model R-5A), for the sake of having a standard design to work from which was in actual use and resembled normal commercial eyeglasses more closely. The immediate downside is that camera placement becomes much more difficult, since the lowest possible mounting point is quite close to the eye. However, for this iteration we opted to build an extension onto the 3D printed design which is highlighted in the figure, which serves a unified enclosure for the new breakout board and lens. This extension places the camera as close as possible to that of the original CIDER frames, although the camera is necessarily closer to the eye. This created the need for a wider-angle camera lens as discussed in the following sections.

As the camera layout changed we moved away from the specific board enclosure seen in these frames. However, the overall design and shape stayed the same throughout all following design passes as we deemed it an acceptable approximation for regular eyewear.

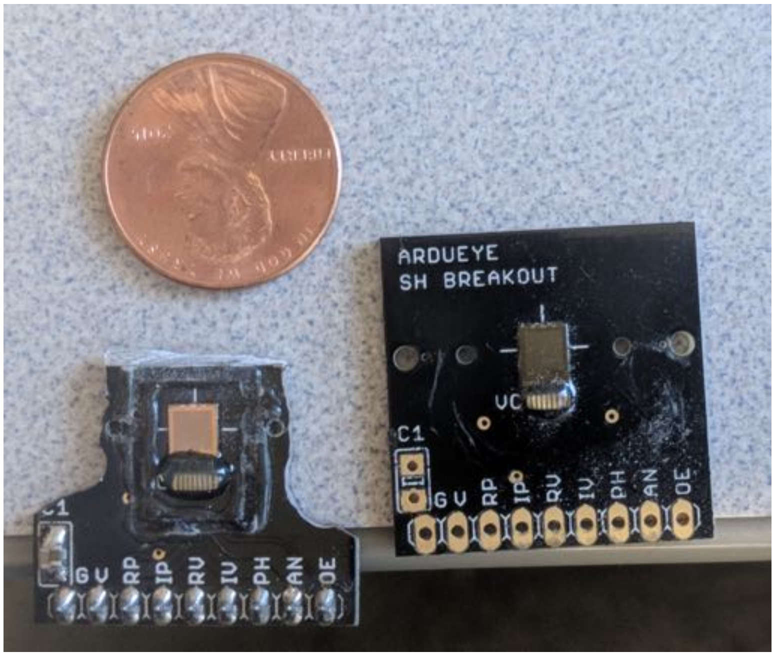


Figure 5.6: Full-size Stonyman breakout board compared against version with excess board material removed

5.3.2 Miniaturized Stonyman Breakout PCB

Examining previous versions of the hardware, the physical size of the camera boards themselves is obviously a major constraint. As shown in figure 5.6, only a small fraction of the real estate of the PCB is dedicated to the housing of the actual imager silicon. The vast majority is space for the lens mount (discussed in the next section), and for the physical pin headers used to connect the camera to the control board. In past iterations we used tools to physically remove excess board material from around the lens mounting where it was not needed (also shown in figure 5.6), which mitigated some of the size problems. However, to move forward, there was a clear need for a major reduction in the size of the board.

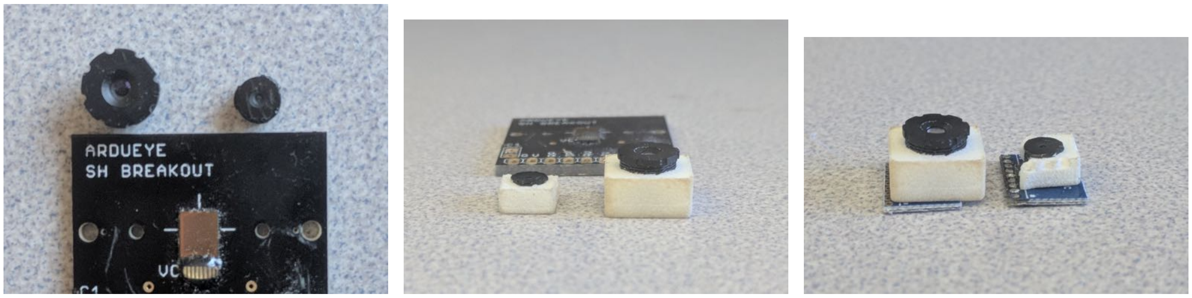
We commissioned a re-designed version of the breakout board with a much smaller pin connector, reducing the physical size requirement dramatically. In addition, the board does not feature any excess space around the imager for supporting a lens mount so as to avoid wasted space. This did create some design challenges for attaching the lenses to the boards without compromising image quality.

These smaller boards proved extremely effective for our purposes as we moved beyond proof-of-concept implementations to designs that were actually practical for realistic use cases. Very surprisingly to us, they did present a number of challenges at the beginning in terms of maintaining the same quality of images as with the larger breakout boards we were accustomed to using. As it turned out, all of these issues stemmed from problems related to the design of the lens enclosures. Once these had been addressed, the miniaturized PCBs became a staple of our work and are used in the final version of the design presented in this chapter.

5.3.3 Wide-Angle Lenses and Enclosures

Having migrated to the smaller camera breakout boards, the final (and largely unanticipated) challenge was integration of the lenses. As discussed previously, this iteration of the frames had a much smaller eye-piece design, meaning that even the furthest possible placement of the cameras would be much closer to the face and eye region than in the CIDER implementation. This implies that there is much more likely to be a problem with generalizability across the population, since, with the same viewing angle at a closer distance, the camera will see a smaller area of the face. The straightforward solution is to change to using a lens with a wider viewing angle to mitigate the problem.

The reason we had not used this approach before is because using a wider viewing angle does have some disadvantages. The biggest is that the amount of usable data is reduced - i.e., if a camera in the same position in space takes an image using a wider viewing angle, the area of the image corresponding to the eye will obviously be reduced. Thus, since all of the visible features have become smaller and harder to separate from each other, the locations of specific eye regions cannot be identified with as much specificity. Thus, eye tracking quality will degrade. In experiments in later iterations we discovered that, so long as the camera is placed relatively close to the eye, this effect has negligible impact on the performance of the sparse sampling neural network. However, with the camera position



(a) Lenses (b) Mount Size (c) Mounts on Small Boards

Figure 5.7: Size comparison of lenses and lens mounts, with a size comparison of mounts against smaller Stonyman breakouts

sufficiently far back as in the CIDER case, there was already sufficient viewing angle and there was no need to risk the possible performance degradation.

	Original	Wide-Angle
Size	1\3”	1\6”
EFL	4.19	2.15
F/NO	2.4	2.7
Field of View	72°	70°

Table 5.1: Specifications for lenses used

5.3.3.1 Lens Comparison, New vs Old

Figure 5.7a shows the new wider-angle lens we chose compared to the original, narrow-viewing-angle lens; the Stonyman imager chip is included as a size reference. Exact specifications for both lenses are given in table 5.1. Note that what we have been referring to as the “wide-angle” lens actually has a similar viewing angle to the “standard” lens, however, the standard lens is wider than the actual imager chip so large portions of it are cut off. The wide-angle lens is significantly smaller than the original, so the entire FOV is utilized, and also smaller than the Stonyman IC. Being smaller than the imager though, is also not ideal as it means that the image will project onto only a subset of the area and the pixels in the corners will not be used, thus losing valuable visual data. However, as there are very

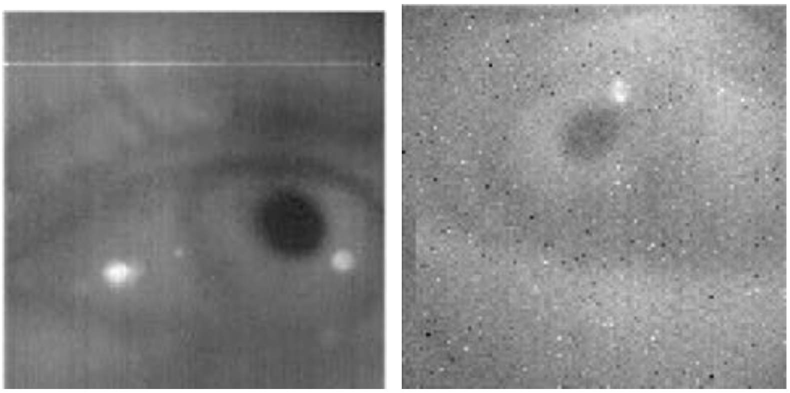

(a) Original Lens (b) Wide-Angle Lens

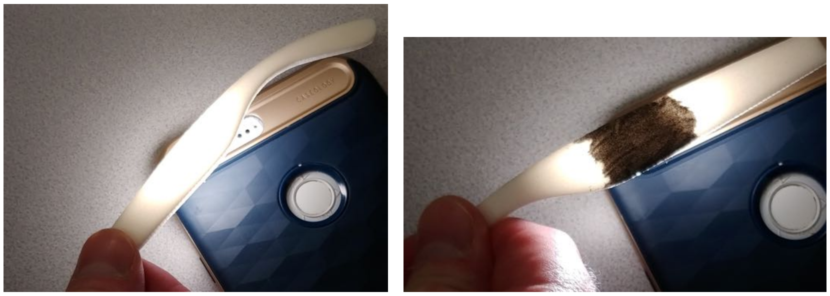
Figure 5.8: Comparison of image contrast between original and wide-angle lens setups

few vendors of lenses this size (most are purchased directly for integration into a camera mechanism for devices such as tablets and smartphones), we were unable to find a lens with a better fit that still had a wide viewing angle.

A major (also unexpected) benefit to the new lens, however, is that the physical size of the enclosure for mounting the lens to the board is much smaller - see figure 5.7b. This beneft stems from two attributes - first, because the actual dimensions of the lens are smaller, the PCB's width and height (X and Y dimensions) are much smaller and do not extend far beyond the edge of the PCB, thus not increasing the size requirements in those dimensions. To emphasize this point, see figure 5.7c, which shows examples of the miniaturized board with mounts attached for each lens type. As you can see, the larger lens mounts hang over the edge of the PCB by a wide margin. In addition, the depth of the breakout is significantly lesser because the focal length of this lens is much loser - thus, it needs to be mounted much closer to the imager compared to the original lens in order to focus on an object at the same distance.

5.3.3.2 Engineering Challenge - Light Leakage

A subtle and very difficult issue arose in this process regarding light leakage - it is critical that no light be able to reach imager other than through the lens. As an example, if



(a) Original (b) Painted

Figure 5.9: One arm of the glasses is held in front of a light source. A significant amount of light passes through the 3D printed material unless we cover it with a layer of acrylic paint.

the overhanging gap between the lens mount and the PCB shown in figure 5.7c were to be left open, the light leakage would overwhelm the light through the lens portal and reduce the contrast dramatically. We assumed this issue to be the culprit when we first mounted the wide-angle lenses to the new miniaturized boards and found that the images had very low contrast — see the comparison in figure 5.8. As described in section 4.4.2, lower contrast means that the cross-model portion of CIDER fails completely and even the neural network model has to work harder to accomplish reasonable eye tracking fidelity.

We assumed that the imager ICs were functioning properly as this issue was present in all of them. We went through several design iterations of the lens mounts, attempting to minimize the air gaps between the printed mounts and the board. The final design shown previously in figure 5.4 completely enclosed the camera and integrated a lens mount directly into the enclosure, however, the issue was still present. We eventually determined that the 3D printed material itself is actually translucent (figure 5.9) — while it seems to the naked eye to be completely opaque, it is sufficiently transmissive that a significant amount of light can pass through and reduce contrast dramatically, which was the cause of the issue shown in figure 5.8. By applying layers of acrylic paint to the outside of the enclosure, we were able to block out the light leakage and match the contrast of the original Stonyman board configuration.

With the lens mount issue solved, we concluded this iteration of the hardware redesign.

5.4 Design Iteration 2: Modular Camera Breakout PCBs

The next pass of the hardware revisions was the first attempt at a full solution to the issue of minimizing the visible impact of the camera hardware on the frames. As previously discussed, the best solution to this problem is simply to move the camera to the inside of the frame eyepiece so as not to change the profile of the frames at all. However, we determined that even with the wide-angle lenses, the camera would not be able to get a full picture of the eye region for all subjects simply due to the extremely short distance between the eye and the camera.

Therefore, we proposed a redesign of the camera hardware that would place multiple cameras at key points within the frame, each of which would have a slightly different view of the face. Our initial design called for 2 - 4 cameras, with empirical exploration to determine their exact number and positions. We created a modular multiple-camera architecture with separate PCB components that could be interconnected using traditional wire or ribbon connectors.

In the course of this design iteration we found a working implementation that achieved both of our original design goals: unobtrusive camera placement and consistent view of the entire eye across subjects. In addition, since the final design only required the use of two cameras, there are two possible ways to implement the system depending on design needs. Since the MCU has two ADCs, it is possible to connect both cameras completely independently of each other and run them in complete parallel. However, this blocks the use of a third camera for recording the world. There are many applications for which this may not be necessary, in which case the system can leverage higher speed and ease of control on the two eye cameras. If, however, a world-facing camera is needed, we provide an architecture for chaining the two eye cameras and driving them together on a single ADC and control bus.

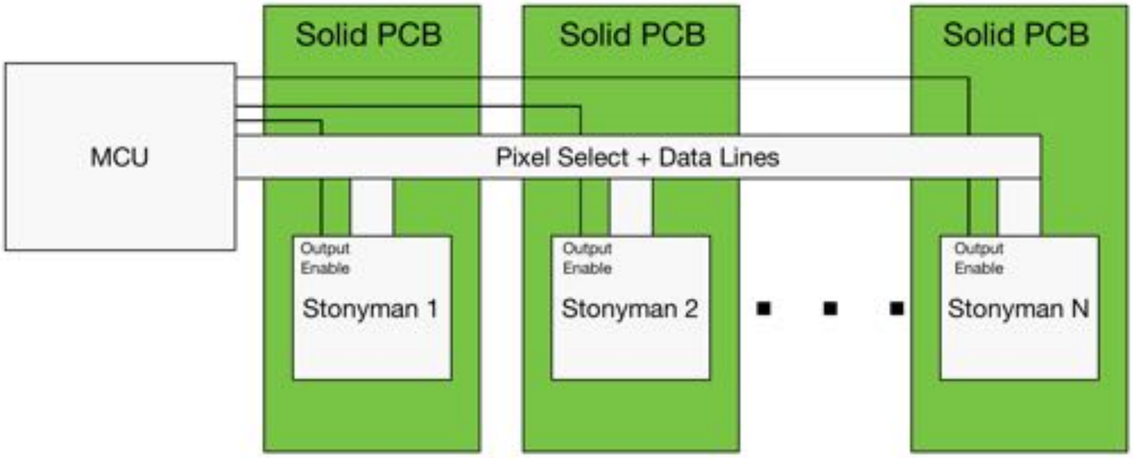


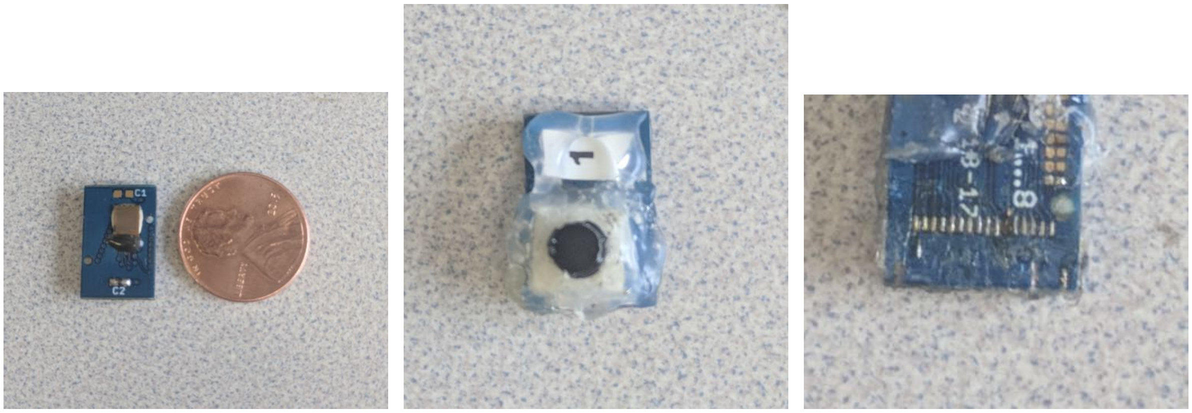
Figure 5.10: Block diagram of distributed-breakout architecture

We here describe the design steps and data collection iterations to arrive at the final design scheme. The design itself also is described in detail over the course of this section.

5.4.1 Architecture Changes and First-Draft Implementation

The first step was to develop a high-level system design for determining how to connect and control the separate camera boards. Our design is given in block diagram form in figure 5.10. The two major design constraints were (a) physical space of the PCB, so as to not be dramatically larger than the edge of the frame, and (b) the number of ADCs available on the Cortex M3 MCU. Each Stonyman has a 9 control lines, and running a completely separate set to each imager would require a significant number of physical wires, which would make the individual PCBs significantly larger than needed due to restrictions on board-to-board connector size. Having separate control lines for each imager would exceed this limit by an order of magnitude. In addition, the MCU we have chosen only has two separate ADC units, meaning that only two cameras could be read from simultaneously. Thus, the system would not be able to operate more than two separate sets of control lines at a time.

The solution to this was straightforward, because one of the Stonyman control lines is an output disable. When pulled low, the IC will block its output line and not read out the



(a) Bare Board (b) Lens Mounted (c) Solder Pad Torn Out

Figure 5.11: Images of distributed breakout boards

value of the selected pixel. We designed the system to share the other control lines across all cameras, and only the output disable was duplicated for each so that one camera could be enabled at a time for pixel sampling. This reduced the board size to within an acceptable range.

Since all of the control lines, including pixel select, are shared, the same pixel access pattern must be shared across all cameras. Indexing a specific pixel will select that same pixel index on all cameras in the system. There are more intelligent schemes that could be used to leverage the fact that the MCU has two separate ADCs in order to more efficiently make use of space, however, for first-run testing we opted for maximum control. We designed the PCBs to be designed with eight enable lines so as to be completely sure that we would not be bottlenecked in that regard.

5.4.1.1 Assembly and Physical Strain

Figures 5.11a and 5.11b give images of the layout of the physical camera boards and examples of lenses mounted to them. In the course of mounting these boards to the frames we again encountered unexpected and potentially catastrophic design challenges. Ribbon connectors used are out of necessity relatively stiff and inflexible, this is to prevent the internal wires from being torn. However, connecting them to the control board required bending

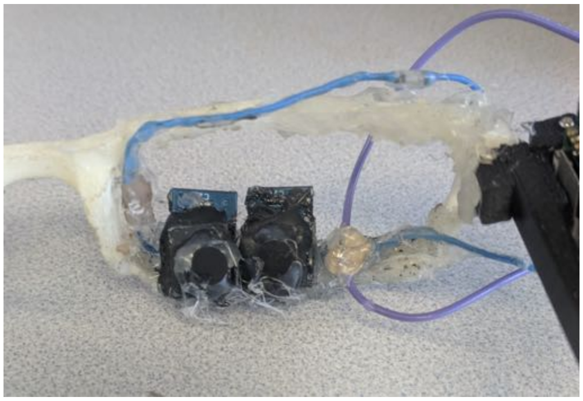


Figure 5.12: Relative size of breakouts to frame

them at extreme angles, which placed significant physical strain on the corresponding board connectors. This resulted in several connectors being physically torn from the boards and taking the solder contacts with them — an example of this is shown in figure 5.11c.

We were eventually able to surpass this issue by changing some elements of the design and adding daughterboards with reinforced connectors to take stress off of the actual camera breakouts. While this is obviously not a finalized solution, it enabled easy experimentation with positions and configurations of the cameras.

5.4.1.2 Board and Lens Size Considerations

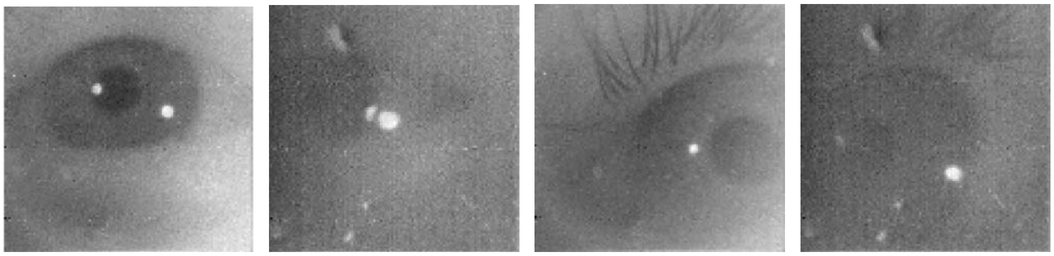
Another design challenge we quickly discovered is the size of the camera breakouts. As with the original two Stonyman breakouts we experimented with (section 5.3.2), the primary contributor to board size is the physical connectors. Figure 5.12 shows just two of the boards placed against the inside of one of the eyepieces with ribbon cables between them. As you can see, the goal is for the cameras to be as close to the nose as possible in order to obtain a straight-on view of the eye, however, there is not a simple practical way to accomplish that given the dimensions of the boards. This problem could be mitigated somewhat with a slight redesign of the dimensions to make it more square, however, it seems that even so no more than two or three cameras can fit into the allotted space.

Thus, we did our initial experiments with only two cameras placed approximately as seen in the figure. To begin, we used the larger, narrower-angle camera lenses from the iShadow / CIDER work (see again figure 5.12 for reference). Our primary reason was the intuition that, as previously mentioned, a narrower field of view will provide better data of the same subject (so long as it is visible) and therefore be more useful tracking. The obvious problem, as discussed in section 5.3.3, is that the narrower field of view is much more sensitive to face shape. However, by fusing the views of multiple cameras we hoped to be able to get a sufficiently wide joint view of the eye region so as to be able to see the eye for all subjects.

5.4.2 Empirical Testing and Final Design

In the end, two problems arose with this design that forced us to make one last change. First and perhaps unsurprisingly, two cameras with narrow lenses were not at all sufficient to achieve good coverage for all subjects — refer back to figure 5.3 for an example with this iteration of the platform. It only shows the output from a single camera, but it illustrates the significance of the problem. We did not proceed further with more cameras, however, because of another problem, also illustrated previously in figure 5.3c. Simply put, since the narrow-angle lenses had to be held fairly far from the imagers to achieve proper focus, on some individuals the lenses would actually touch the face or the lower eyelashes, creating physical irritation for the wearer. Between of these two issues, we decided to migrate to the wide-angle lenses again and examine their performance.

To mitigate both of these issues — overly narrow field of view and lenses touching the face — we changed over to the wide-angle lenses described in section 5.3.3 previously. We experimented with one and two cameras with the wide lenses and found that two cameras provided sufficient visual coverage for all face shapes we attempted. See figure 5.13, which shows the output of the two cameras for the same two subjects on opposite ends of the spectrum in terms of face shape and camera positioning. In addition, the lens mounts were



(a) Subject 1, Camera 1 (b) Subject 1, Camera 2 (c) Subject 2, Camera 1 (d) Subject 2, Camera 2

Figure 5.13: Image pairs from subjects shown in figure 5.3

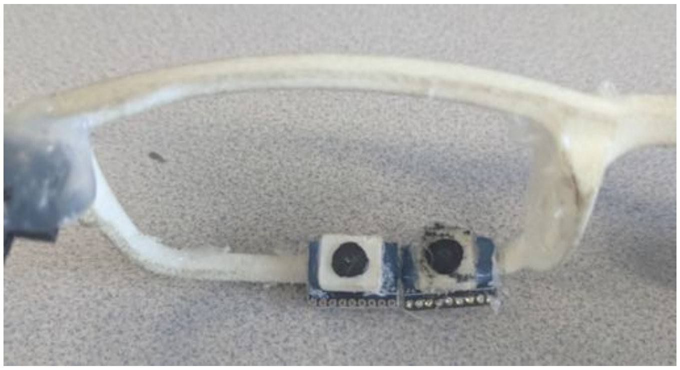


Figure 5.14: Simpler breakouts mounted to frame

sufficiently shallow so as to not be touching the face or eyelashes. Our concern with this change was that the condensed visual field would mean less usable information about the eye, however, we present experimental results in the next section which demonstrate that not to be the case.

Thus, we have established this design as the solution for our challenging design tradeoff: “attractive,” unobtrusive camera placement vs. robust view of the view across subjects. We performed experimental validation of this, discussed in the next section, which indicates that this camera arrangement does not negatively impact the quality of the simpler neural-network-based gaze tracking algorithm in a significant way.

5.4.2.1 Design Space Options

Since only two cameras are needed, an additional design parameter becomes available: whether to use the two cameras on a shared bus or independently. As discussed at the beginning of this section, since there are two ADCs, it is possible to independently connect each camera to the MCU with separate control lines and outputs. The benefit of this is twofold. First is granularity of control, there is independent control of the pixel sampling for each camera as well as parallel sampling of the pixels on the two ADCs. The second benefit is that the smallest breakout boards can be used (as seen in figure 5.14), since the shared bus lines are not needed. The drawback of this configuration is that it is no longer possible to use a third, world-facing camera without resorting to another multiplexing scheme.

Which side of this tradeoff to use will generally be specified by the application needs. Also, it is as of yet unclear whether being able to sample from both cameras is beneficial or if it would be easiest to simply use only the camera that has the better view of the eye, effectively treating it as a single-eye-camera system. In that case, the shared-bus architecture is objectively superior since it always allows for the possibility of a world-facing camera. Current data suggests that the neural network does not benefit from the added data when using both cameras simultaneously, it seems that one camera with a sufficiently good view of the eye is enough. However, a larger user study would have to be done to confirm that this is truly the case across the population.

5.5 Neural Network Gaze Algorithm Performance

To conclude this chapter, we present some small benchmarks to compare the performance of the new camera-integrated frame design with the previous CIDER design. The goal is to demonstrate that performance of eye tracking has not suffered dramatically and that we have not sacrificed device performance to an undue extent in the service of making the frames more comfortable and usable.

For the evaluation, we used the performance of the original neural-network-only gaze tracking algorithm from chapter 3 for testing the effectiveness of a given camera configuration and compare between them. We used the neural network partly for fast iteration — data collection and evaluation was very straightforward, discussed momentarily — and also because we expected the effectiveness of the learning algorithm to be a rough proxy for the utility of the data generated from a particular camera configuration. I.e., if a certain placement of the camera yielded better gaze tracking accuracy, we would assume that placement provided more data about the eye in an information-theoretic sense.

5.5.1 Experimental Setup

For ease of changing the camera positions, we opted to do these experiments using two independently connected eye-facing cameras. This meant, however, that we had no world-facing camera for gaze labeling. What we decided instead to do was use the same moving-dot process as described in section 3.4, but generate the labels on the host machine based on the position of the dot at the time an eye frame was taken. To make this process consistent, we had to make two adjustments to the standard calibration process: first, we needed to ensure a consistent head position so that the angular error would be the same across experiments — we did this by creating a platform for subjects to rest their chin.

The second adjustment to the calibration process was to synchronize the iShadow MCU clock with the host PC. This was done in a very approximate manner by running starting the moving-dot script at the same time as the iShadow data collection begins. The starting times are therefore not exactly matched and drift will start to accumulate, however, since the experiment lasts less than five minutes we expect that the drift will not become significant. Empirical observations confirm this to be true, we find that the pattern of recorded eye movements matches the movements of the target consistently across the experiment. Above all, since the effects of drift will be the same across experiments, it will be an apples-to-apples comparison and we believe that this is an acceptable setup for the purposes of comparing various camera configurations and subjects.

5.5.2 Note on Resistance of Performance to Optimizations.

As a brief aside before discussing the final results, it is worth noting that we performed a large number of iterations of this experiment on a single user during the design of the two-camera configuration described above. Our goal was to explore the design space and determine whether there was any way to improve the baseline performance of the ANN model with certain configurations of the cameras. Across all of these experiments, however, we found that it was not possible to increase the performance of the ANN above that of the CIDER case. This was surprising and unintuitive to us, as we had expected some of these to naturally improve eye tracking performance. Configurations that we tried included:

- Giving both cameras a good view of the eye and providing data from both to the neural network. This did not improve performance over having a single camera with an adequate view of the eye, and the feature selection would tend to choose the majority of the pixels from one camera or the other when given the option of both.

- Providing two cameras each with a partial view of the eye. However, unless the iris was out-of-frame for a large portion of the time in both cameras (i.e., each camera's field of view was missing a significant amount of the eyeball region), the ANN model would choose whichever camera had the better view and predominantly select pixels from that camera. Even if it did choose to make use of both to some degree, the performance was not significantly better than using just the one camera in most cases.

- Within a given range, it seems that the size of the eye within the image (i.e., the zoom level) does not dramatically affect the performance. This effect can be observed within figure 5.3 above — even though the eye region occupies fewer pixels in the CIDER case, and therefore there is less available information, its performance is not dramatically affected. Although it is generally counterinituitive, it seems reasonable that having the eye occupy a smaller number of pixels is actually helpful in this case, since each pixel provides more information for the cost of accessing it. In fact, Tonsen et. al. recently observed that they were able to subsample down to 15x15 pixels and still achieve good performance (albeit across multiple cameras) [81].

- Perhaps unsurprisingly, the ANN's performance does not seem to suffer if the eye is out of focus. We experimented with lens positions with significantly incorrect focus, resulting in a very blurry image, and did not find a serious decrease in performance. Since the visual information is still present but simply smeared across multiple pixels, it is sensible that the ANN would be able to learn to extract the information.

All of these experiments lead us to believe that the ANN model's performance has likely saturated, since in a few of these experiments we are concretely adding and subtracting information about the eye and not observing a corresponding change in performance. It is fair to conclude, then, that the model is simply not complex enough to leverage the additional information in a significant way. This was an interesting discovery and an open avenue for future work. Exploring the tradeoff of turning model complexity up and down — namely, more performance vs longer execution time — would be worthwhile, especially as embedded processing hardware continues to improve in speed and power.

5.5.3 Results

We recruited four subjects for this evaluation, chosen based on differing eye positions relative to the camera. The goal is to show that performance is acceptable across the entire spectrum of eye positions, so we selected individuals to cover this spectrum. The two subjects discussed previously in the context of figure 5.3 are included, as well as two others with an eye position closer to what we have found to be the average by empirical observation. Each subject performed the standard moving-dot experiment as described above, once with the CIDER frames and once with the new camera-integrated frames.

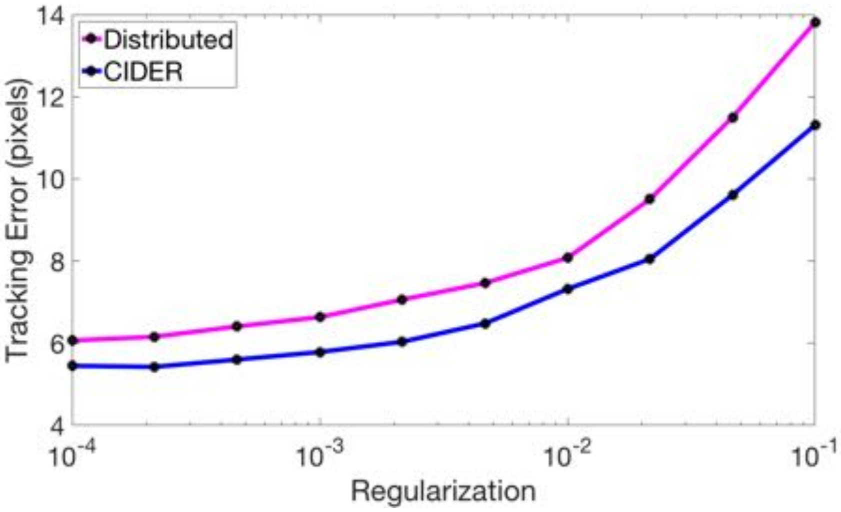


Figure 5.15: Comparison of gaze tracking performance between CIDER frames and camera-integrated frames. Error is given in pixels, and the x-axis is the range of regularization values used.

The results are given in figure 5.15. The performance is given across the standard range of regularization values used for evaluation in previous chapters. The performance of the camera-integrated frames is very close to that of the CIDER frames, averaging 15% higher overall. We consider this to be a very acceptable performance level, partly because the absolute error of the CIDER platform has been demonstrated to be very low, so a low-percentage increase in error is also small on an absolute scale.

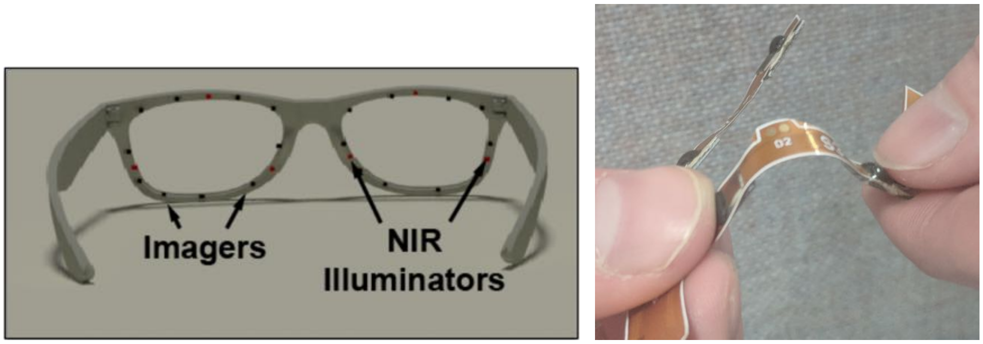
The more significant observation regarding the relative performance is that this is a first-pass attempt at optimizing the camera positions and angles for the distributed imager camera, and we do not contend that it has been completely optimized. There are a number of small factors which contribute to decreased performance in the distributed frames which are all correctable with further redesigns. First, as can be seen in figure , the most divergent subjects' eyes are mostly captured by the cameras, and it would take only slight adjustments for them to be fully captured. One option would be to commission lenses with a slightly wider viewing angle, which would be feasible at production scale. Another simple fix would be to redesign the printed frames so that the cameras have a berth and sit within the frame instead of being glued on top of it, as is the current configuration — this fix would allow the cameras to sit further back and cover a wider range.

There are other small design issues owing to the fact that this is a prototype design, and a more complete refactor of the platform would need to be undertaken to take fullest advantage of the new camera placement strategy. However, we feel that this result shows that there is in fact a point in the design space which effectively trades off the design goals discussed at the beginning of this chapter. With this design, we have demonstrated that it is possible to create a wearable eyeglass system that operates locally with effective performance levels without imposing extreme mental or social burden on the wearer. With minimal revisions to the design, this version of the platform could become the tool needed to perform true in-the-wild studies and open up new avenues for eye-tracking-based research.

5.6 Future Design: Flex-PCB Multi-Camera Configuration

The previous design has succeeded in the design goals we set out for, however, due to physical restrictions it is restricted to using two cameras. It has been shown in outside work that an array of small cameras lining the inside of the frames may yield even better results (potentially with image downsampling to reduce the amount of data per camera) [81]. We began exploring an architecture to facilitate this idea and did a first implementation pass. However, the implementation challenges to a device of this nature are even more extreme than the ones discussed thus far, and at this current time it is unclear that the benefits are worth the added design effort.

We present here the details of our design and the progress made on the first pass, as well as the challenges encountered. Completing this architecture and integrating it into a more appropriate algorithm is discussed in chapter 6 as future work.



(a) Concept

(b) Flex PCB

Figure 5.16: Concept for the distributed-imager frame, and the flexible PCB material used to realize it.

5.6.1 System Architecture

We proposed a redesign of the camera hardware that would place multiple cameras at key points within the frame, each of which would have a slightly different view of the face. Our initial design called for at least four cameras, with empirical exploration to determine the exact number and positions of the cameras. See figure 5.16a for a digital mockup of the idea.

Our implementation of this design was based on the idea of using PCBs with a flexible substrate. This material (see figure 5.16b) is a thin, flexible polymer that can accept circuit components and electrical traces and still be able to deform a small amount. It does not have the same shape requirements as traditional “hard” PCBs — i.e., does not have to be rectangular — and thus can be shaped to exactly fit the eyepiece. The overall idea was to create a flexible PCB in the shape of the glasses eyepiece which would contain all of the cameras and interconnect circuitry.

The first step was to develop a high-level system design for determining how to connect and control the camera bank. Our design is given in block diagram form in figure 5.17. It is very similar in concept to the architecture for the two-camera solid-PCB system described previously; the major change is that both the cameras and the interconnects are all included in a single flex board instead of being distributed.

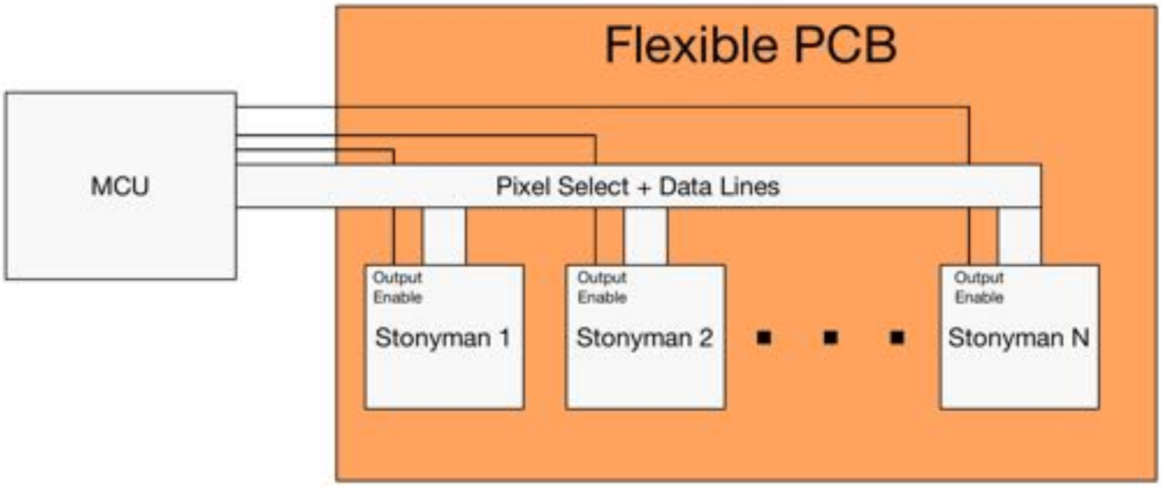
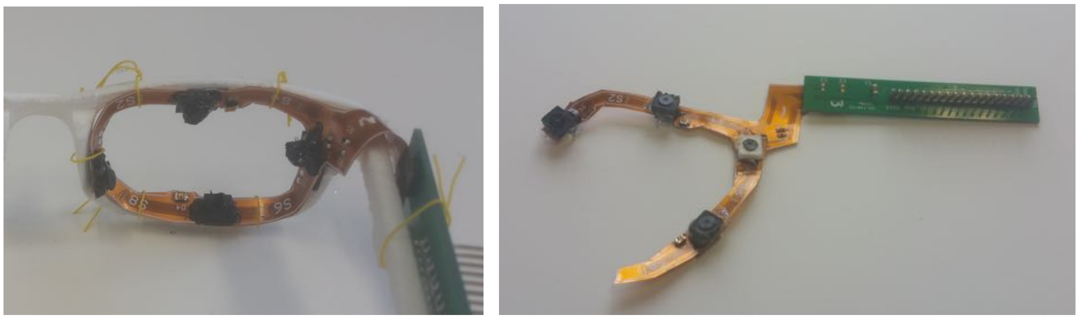


Figure 5.17: Block diagram of flex-PCB imager architecture



(a) Flex Board in Frames

(b) Flex Board Unwrapped

Figure 5.18: Flex boards and width comparison with frames

We again use the output-enable control scheme, for the same reasons as before — there is not enough physical space for separate control lines for all cameras. With the same number of control lines as used previously (eight), the width of the flex board is on the scale of the frames' edge. See figure 5.18a for an example. In addition, having this many control lines and the corresponding mount points for the cameras allows for the option of experimenting with a number of potential camera positions to empirically determine which are the most useful.

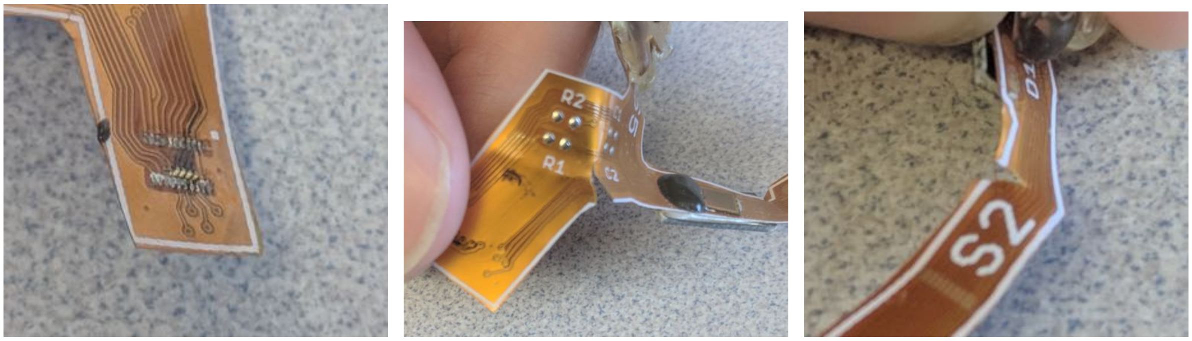


Figure 5.19: Examples of most common types of damage to flex boards. Typically the substrate would tear under stress (middle and right), occasionally the board-to-board connector pins would rip out of the substrate (left).

5.6.2 Flex Structure Design and Assembly

After designing the system architecture, we developed the physical layout of the flex PCB structure. As discussed previously, flex PCBs do not have the same shape requirements as traditional boards and so could be shaped to fit the inside edge of the eyepiece. To get a correct fit, we used a simple "paper doll" approach in which we used measurements of the eyepiece to create a 2D layout on paper. These layouts functioned as stand-ins for the flex boards so that we could ensure that the design correctly fit the eyepiece. Since this involves bending a 2D shape to fit a 3D structure, a few iterations were needed to ensure that the dimensions of the layout were correct against the eyepiece.

Once the layout was finished, we commissioned and received the flex boards, seen in figure 5.18b. The layout had eight potential positions for the imagers, split evenly between top and bottom. It was impractical to populate every position on each board, so we ordered them populated with four imagers apiece, alternating "odd" and "even" locations to ensure that we could test every position. The interface between the flex board and the controller was a simple board-to-board connector, as shown in the figure. For ease of experimentation and rapid prototyping we designed a simple hard PCB connector (also shown in the figure) to serve as a breakout for the flex board so that wired connections could be used.

5.6.2.1 Board Testing and Design Problems

At this point we were prepared to begin testing the boards. Over the course of several months of testing and redesign we realized a fatal flaw with the use of the flex PCBs: the substrate is extremely frail, meaning that even mild physical stress on the boards would very likely damage them irreparably. See figure 5.19 for examples of this. The primary cause of this was tears in the substrate that occurred either during the process of mounting to the frame or would develop shortly after mounting due to simple movements of the frames (e.g., setting them down on a table). The other physical stressor that caused damage was the connection point between the flex board and the solid breakout. The physical connection between the contact points and the underlying substrate is not as strong as on hard PCBs, meaning that it took very little stress on the board-to-board connector in order to rip out the solder contact points, thus rendering the board completely unusable. An example of this is also shown in the figure.

We made several attempts to mount boards to the frames that all resulted in the types of stress damage discussed to occur. Each attempt was complicated by the fact that mounting lenses to the cameras was extremely time-consuming, as aligning the lenses properly and anchoring them without causing damage to the substrate, the IC, or the connection between them required very careful use of tools. After a number of failed attempts to safely attach the flex PCBs to a glasses frame, we began altering the design slightly in order to reduce the stresses placed on the PCB — for example, we designed an additional 3D-printed part to mount to the glasses to reduce the physical stress placed on certain portions of the board. Unfortunately, instead of solving the problem this only increased the mean time to failure, as tears would eventually develop at the less intense stress points over time.

After a number of the PCBs had been destroyed by the mounting process, we concluded that integrating the flexible substrate material is a non-trivial task. It is required to redesign the 3D printed frames such that all stresses could be removed from the flex boards to use them safely, and any changes to the flex board would require a corresponding frame redesign. In addition, there are still open design questions such as: (a) how to safely bridge from the flex material to a more traditional connector for the master control board, since the board-to-board connectors proved quite fragile; and (b) how to mount camera lenses to the substrate material in an effective, repeatable manner without causing damage to the substrate or the imager IC.

With these challenges in mind, we concluded development on the flex PCB architecture for the conclusion of this thesis. In future work (chapter 6), we discuss the options for resolving the design of this platform and integrating with interesting new techniques from the eye tracking community.

CHAPTER 6

SUMMARY AND FUTURE WORK

6.1 Thesis Summary

This thesis presents the design and implementation of a first-of-its-kind lightweight wearable eye tracking device. By exploiting the physical and temporal structure inherent to the eye tracking problem to do adaptive sampling, we demonstrate that it is possible to implement an eye tracking scheme with power consumption orders of magnitude lower than that of existing commercial devices while keeping similar eye tracking fidelity. In addition, we discuss the nontrivial design challenges required at all stages to realize these benefits in a platform that is fully suitable for in-situ use with actual subjects.

The foundation of this platform is the technique of intelligently sampling pixels from the imager hardware based on features of the eye and then doing efficient local processing on them to extract the relevant eye parameters at low power cost. This could be viewed as both a resource-aware eye tracking algorithm and an application-aware information sensing and processing regime. The first realization of this was a sparse-sampling neural network algorithm that learned to identify the most salient image regions for the purpose of tracking the wearer's gaze location in space. We fused this algorithm with a custom imager that facilitates sampling specific pixels in order to dramatically decrease resource consumption. This first-of-its-kind platform, iShadow, was able to do accurate gaze tracking at a fraction of the power needs of commercial systems.

We then approached the problem again with a view towards dramatically improving the ratio of power spent to quality of eye tracking data, in order to make the platform feasible as a realistic wearable device. We constructed a new pupil-tracking algorithm based on lightweight computer vision features, which leverages the smoothness of the eye's motion to reduce even further the amount of camera sampling needed. To guard against very infrequent discontinuities resulting from blinks or reflections off the eye, we integrated this model with the previously-used one-shot neural network algorithm. The use of these two together formed the core of the improved CIDER platform, which demonstrated eye tracking performance comparable with industrial devices while achieving power draw of less than ten milliwatts.

Lastly, having achieved our core technical goals through these innovations, we engaged in the engineering work required to push the platform from being a lab prototype to a system that is practical for long-running in-situ studies on actual subjects. The core design challenges were (a) making the system unobtrusive to the wearer so as to be comfortable to use for long periods and not introduce bias into their behavior, and (b) to ensure that viable eye tracking could be collected across the population. Due to the competing nature of these objectives, we had to iterate a number of design concepts attempting to find an implementation that would balance them and also keep to the core restrictions of a glasses-mounted platform. We were able to achieve such a design, and at the conclusion of this thesis have constructed a practical, wearable eye tracking device suitable for longitudinal studies of behavior and cognition.

6.2 Future Work

Here we briefly discuss interesting extensions and improvements upon this work as potential future research endeavors.

6.2.1 Dynamic Resolution Scaling with Miniature Camera Bank

Yet another potential avenue for exploration is presented in the work of Tonsen et. al. [81], who performed experiments with varying levels of software subsampling to create low-resolution full images. They discovered that even for resolutions as low as 10x10, it is possible to get reasonable eye tracking performance with multiple cameras (in their case, three per eye). However, this could only be explored at the software level due to their use a standard camera architecture, whereas the Stonyman cameras offer an in-hardware resolution scaling feature called “binning,” which averages groups of pixels together in a grid pattern.

They explored the possibility of using multiple miniature cameras spread through the glasses, but were unable to build a practical prototype due to the fragility of the camera hardware. The results they did generate suggest that it would be beneficial to have a bank of small cameras distributed throughout the frame, similar to the flex PCB architecture we began developing. This design, in concert with the downsampling scheme, merits exploration since it could reduce sampling time (and associated power costs) even further than is possible with our current algorithm, and adding a high number of cameras would allow for good coverage of the eye region at an extremely low per-camera power and time cost. It is even possible that a scheme for dynamically changing resolutions (perhaps differently across cameras) would yield even better performance.

6.2.2 Sensor Fusion

Some other works have explored the possibility of doing eye tracking with a system that fuses data from video-oculography and other sensing modalities, such as a photosensor [61] or IMU [43]. However, these systems have all used traditional full-camera sampling techniques. We hypothesize that it may be possible to integrate these sensing modalities, or others such as EOG, into the eye tracking pipeline much closer to the physical layer. A straightforward conception of this would be to add the time series data as an input to the ANN model, or even to adaptively change the active subsampling strategy based on higher-level understanding of what’s happening based on other sensors. This is an appealing opportunity to leverage the strengths of less dense, cheaper-to-sample time series sensors to assist the more complex camera-based sensing models.

6.2.3 Improved Machine Learning Model

As observed in section 5.5, the subsampling neural network model formed at the beginning of this work seems to have reached its peak performance capacity with the optimizations described throughout this work. While its performance has been outstanding, it is unable to benefit from added information such as when data from two cameras is available. It seems reasonable to explore the possibility that there is more room to explore the tradeoff between simpler models with faster execution speed but low capacity, and slightly more advanced models that can leverage the newly available data in more recent iterations of the design. A fully unexplored avenue would be to have a tiered model that is able to take advantage of network resources such as smartphones or even the cloud to enhance performance by streaming select data to more complex models, executed remotely, that can glean deeper insights from the data.

BIBLIOGRAPHY

[1] A neural-based remote eye gaze tracker under natural head motion. *Computer Methods and Programs in Biomedicine 92*, 1 (2008), 66 – 78.

[2] *ETRA '16: Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications* (New York, NY, USA, 2016), ACM.

[3] Adhawk Microsystems. Adhawk Microsystems Website. http://www.adhawkmicrosystems.com/, 2018.

[4] Aerts, M B, Esselink, R A J, Abdo, W F, Meijer, F J A, Drost, G, Norgren, N, Janssen, M J R, Borm, G F, Bloem, B R, and Verbeek, M M. Ancillary investigations to diagnose parkinsonism: a prospective clinical study. *J. Neurol.* (Nov 2014).

[5] Ahmad, Rizwan. *Understanding the Language of the Eye: Detecting and Identifying Eye Events in Real Time via Electrooculography*. PhD thesis, UC San Diego, 2016.

[6] Applied Science Laboratories. NeXtGeneration Mobile Eye: Mobile Eye XG. `http://www.asleyetracking.com/Site/Portals/0/MobileEyeXGwireless.pdf`, 2013. Online; accessed April 7, 2013.

[7] Babcock, Jason S, and Pelz, Jeff B. Building a lightweight eyetracking headgear. In *Proceedings of the 2004 symposium on Eye tracking research & applications* (2004), ACM, pp. 109–114.

[8] Baluja, Shumeet, and Pomerleau, Dean. Non-intrusive gaze tracking using artificial neural networks. Tech. rep., Pittsburgh, PA, USA, 1994.

[9] Baumeister, RF, and Alquist, JESSICA L. Self-regulation as a limited resource: Strength model of control and depletion. *Psychology of self-regulation: Cognitive, affective, and motivational processes 11* (2009), 21–33.

[10] Baumeister, Roy F, and Heatherton, Todd F. Self-regulation failure: An overview. *Psychological inquiry 7*, 1 (1996), 1–15.

[11] Bekkering, H, Neggers, S F, Walker, R, Gleissner, B, Dittrich, W H, and Kennard, C. The preparation and execution of saccadic eye and goal-directed hand movements in patients with Parkinson's disease. *Neuropsychologia 39*, 2 (2001), 173–83.

[12] Bishop, Christopher M. *Neural networks for pattern recognition*. Oxford university press, 1995.

[13] blooloop.com. AdHawk creates game-changing motion-tracking chip for VR/AR. https://blooloop.com/link/adhawk-game-changing-vr-chip/, 2018.

[14] Bolding, Mark S, Lahti, Adrienne C, White, David, Moore, Claire, Gurler, Demet, Gawne, Timothy J, and Gamlin, Paul D. Vergence eye movements in patients with schizophrenia. *Vision Res. 102* (Sep 2014), 64–70.

[15] Bonnet, Cecilia, Rusz, Jan, Megrelishvili, Marika, Sieger, Tomáš, Matoušková, Olga, Okujava, Michael, Brožová, Hana, Nikolai, Tomáš, Hanuška, Jaromír, Kapianidze, Mariam, Mikeladze, Nina, Botchorishvili, Nazi, Khatiashvili, Irine, Janelidze, Marina, Serranová, Tereza, Fiala, Ondřej, Roth, Jan, Bergquist, Jonas, Jech, Robert, Rivaud-Péchoux, Sophie, Gaymard, Bertrand, and Růžička, Evžen. Eye movements in ephedrone-induced parkinsonism. *PLoS ONE 9*, 8 (2014), e104784.

[16] Borsato, Frank H, and Morimoto, Carlos H. Episcleral surface tracking: Challenges and possibilities for using mice sensors for wearable eye tracking. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications* (2016), ACM, pp. 39–46.

[17] Bulling, Andreas, Roggen, Daniel, and Tröster, Gerhard. Wearable eog goggles: Seamless sensing and context-awareness in everyday environments. *Journal of Ambient Intelligence and Smart Environments 1*, 2 (2009), 157–171.

[18] Carvalho, Nicolas, Noiret, Nicolas, Vandel, Pierre, Monnin, Julie, Chopard, Gilles, and Laurent, Eric. Saccadic eye movements in depressed elderly patients. *PLoS ONE 9*, 8 (2014), e105355.

[19] `http://www.centeye.com/%20products/current-centeye-vision-chips/`. ”Current Centeye Vision Chips”, Accessed: 2015-06-24.

[20] Cheng, Daniel, and Vertegaal, Roel. An eye for an eye: a performance evaluation comparison of the lc technologies and tobii eye trackers. In *Eye Tracking Research & Application: Proceedings of the 2004 symposium on Eye tracking research & applications* (2004), vol. 22, pp. 61–61.

[21] Christiansen, Paul, Cole, Jon C, and Field, Matt. Ego depletion increases ad-lib alcohol consumption: Investigating cognitive mediators and moderators. *Experimental and clinical psychopharmacology 20*, 2 (2012), 118.

[22] `http://www.arm.com/products/processors/cortex-m/cortex-m3.php`. ”Cortex-M3 Processor - ARM”, Accessed: 2015-06-24.

[23] Dadkhahi, Hamid, and Duarte, Marco F. Masking strategies for image manifolds. *IEEE Transactions on Image Processing 25*, 9 (2016), 4314–4328.

[24] Dadkhahi, Hamid, Duarte, Marco F, and Marlin, Benjamin. Isomap out-of-sample extension for noisy time series data. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on* (2015), IEEE, pp. 1–6.

[25] Danziger, Shai, Levav, Jonathan, and Avnaim-Pesso, Liora. Extraneous factors in judicial decisions. *Proceedings of the National Academy of Sciences 108*, 17 (2011), 6889–6892.

[26] Deshpande, Amol, Guestrin, Carlos, Madden, Samuel R, Hellerstein, Joseph M, and Hong, Wei. Model-driven data acquisition in sensor networks. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30* (2004), VLDB Endowment, pp. 588–599.

[27] Dhuliawala, Murtaza, Lee, Juyoung, Shimizu, Junichi, Bulling, Andreas, Kunze, Kai, Starner, Thad, and Woo, Woontack. Smooth eye movement interaction using eog glasses. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction* (2016), ACM, pp. 307–311.

[28] Duchowski, Andrew T. *Eye Tracking Methodology: Theory and Practice*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[29] Duchowski, Andrew T, Shivashankaraiah, Vinay, Rawls, Tim, Gramopadhye, Anand K, Melloy, Brian J, and Kanki, Barbara. Binocular eye tracking in virtual reality for inspection training. In *Proceedings of the 2000 symposium on Eye tracking research & applications* (2000), ACM, pp. 89–96.

[30] Flinn, Jason, and Satyanarayanan, Mahadev. Energy-aware adaptation for mobile applications. In *ACM SIGOPS Operating Systems Review* (1999), vol. 33, ACM, pp. 186–201.

[31] Fried, M., Tsitsiashvili, E., Bonneh, Y. S., Sterkin, A., Wygnanski-Jaffe, T., Epstein, T., and Polat, U. ADHD subjects fail to suppress eye blinks and microsaccades while anticipating visual stimuli but recover with medication. *Vision Res. 101* (Aug 2014), 62–72.

[32] Fried, Moshe, Tsitsiashvili, Eteri, Bonneh, Yoram S, Sterkin, Anna, Wygnanski-Jaffe, Tamara, Epstein, Tamir, and Polat, Uri. ADHD subjects fail to suppress eye blinks and microsaccades while anticipating visual stimuli but recover with medication. *Vision Res. 101* (Aug 2014), 62–72.

[33] Friedl, Karl E, Grate, Stephen J, Proctor, Susan P, Ness, James W, Lukey, Brian J, and Kane, Robert L. Army research needs for automated neuropsychological tests: monitoring soldier health and performance status. *Archives of Clinical Neuropsychology 22* (2007), 7–14.

[34] Hansen, Dan Witzner, and Ji, Qiang. In the eye of the beholder: A survey of models for eyes and gaze. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 32*, 3 (2010), 478–500.

[35] Hastie, Trevor, Tibshirani, Robert, Friedman, Jerome, and Franklin, James. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer 27*, 2 (2005), 83–85.

[36] Holmqvist, Kenneth, Nyström, Marcus, Andersson, Richard, Dewhurst, Richard, Jarodzka, Halszka, and Van de Weijer, Joost. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011.

[37] Invensense-9150. MPU-9150 Nine-Axis (Gyro + Accelerometer + Compass) MEMS MotionTracking Device. `http://www.invensense.com/mems/gyro/mpu9150.html`, 2013.

[38] Ishiguro, Yoshio, Mujibiya, Adiyan, Miyaki, Takashi, and Rekimoto, Jun. Aided eyes: eye activity sensing for daily life. In *Proceedings of the 1st Augmented Human International Conference* (2010), ACM, p. 25.

[39] JINS MEME. JINS MEME Eyewear. https://jins-meme.com/en/, 2018.

[40] Kahneman, Daniel. *Thinking, fast and slow*. Macmillan, 2011.

[41] Kassner, Moritz, Patera, William, and Bulling, Andreas. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication* (2014), ACM, pp. 1151–1160.

[42] Kim, Elizabeth S, Naples, Adam, Gearty, Giuliana Vaccarino, Wang, Quan, Wallace, Seth, Wall, Carla, Perlmutter, Michael, Kowitt, Jennifer, Friedlaender, Linda, Reichow, Brian, et al. Development of an untethered, mobile, low-cost head-mounted eye tracker. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (2014), ACM, pp. 247–250.

[43] Lanata, Antonio, Valenza, Gaetano, Greco, Alberto, and Scilingo, Enzo Pasquale. Robust head mounted wearable eye tracking system for dynamical calibration. *Journal of Eye Movement Research 8*, 5 (2015).

[44] Lander, Christian, Krüger, Antonio, et al. heyebrid: A hybrid approach for mobile calibration-free gaze estimation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1*, 4 (2018), 149.

[45] Lee, Eui Chul, Park, Kang Ryoung, Whang, Min Cheol, and Park, Junseok. Robust gaze tracking method for stereoscopic virtual reality systems. In *International Conference on Human-Computer Interaction* (2007), Springer, pp. 700–709.

[46] Li, Bin, Fu, Hong, Wen, Desheng, and LO, WaiLun. Etracker: A mobile gaze-tracking system with near-eye display based on a combined gaze-tracking algorithm. *Sensors 18*, 5 (2018), 1626.

[47] Li, Dongheng, Babcock, Jason, and Parkhurst, Derrick J. openeyes: a low-cost head-mounted eye-tracking solution. In *Proceedings of the 2006 symposium on Eye tracking research & applications* (2006), ACM, pp. 95–100.

[48] Li, Ming, Ganesan, Deepak, and Shenoy, Prashant. Presto: feedback-driven data management in sensor networks. *IEEE/ACM Transactions on Networking (TON) 17*, 4 (2009), 1256–1269.

[49] Li, Tianxing, Liu, Qiang, and Zhou, Xia. Ultra-low power gaze tracking for virtual reality. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems* (2017), ACM, p. 25.

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

[51] Linder, J. A., Doctor, J. N., Friedberg, M. W., Reyes Nieva, H., Birks, C., Meeker, D., and Fox, C. R. Time of Day and the Decision to Prescribe Antibiotics. *JAMA Intern Med* (Oct 2014).

[52] Mack, David J, Schönle, Philipp, Fateh, Schekeb, Burger, Thomas, Huang, Quiting, and Schwarz, Urs. An eog-based, head-mounted eye tracker with 1 khz sampling rate. In *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE* (2015), IEEE, pp. 1–4.

[53] Martinez-Conde, S., Macknik, S. L., Troncoso, X. G., and Hubel, D. H. Microsaccades: a neurophysiological analysis. *Trends Neurosci. 32*, 9 (Sep 2009), 463–475.

[54] Molitor, Robert J, Ko, Philip C, and Ally, Brandon A. Eye Movements in Alzheimer's Disease. *J. Alzheimers Dis.* (Sep 2014).

[55] Morgante, James D, Zolfaghari, Rahman, and Johnson, Scott P. A critical test of temporal and spatial accuracy of the tobii t60xl eye tracker. *Infancy 17*, 1 (2012), 9–32.

[56] Morimoto, C.H., and Mimica, M.R.M. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding 98*, 1 (2005), 4–24.

[57] Nocedal, Jorge, and Wright, Stephen J. *Numerical optimization*. Springer Science+ Business Media, 2006.

[58] Patney, Anjul, Salvi, Marco, Kim, Joohwan, Kaplanyan, Anton, Wyman, Chris, Benty, Nir, Luebke, David, and Lefohn, Aaron. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics (TOG) 35*, 6 (2016), 179.

[59] Pupil Labs. Pupil Labs Website. https://pupil-labs.com/, 2018.

[60] Rantanen, Ville, Vanhala, Toni, Tuisku, Outi, Niemenlehto, P, Verho, Jarmo, Surakka, Veikko, Juhola, Martti, and Lekkala, Jukka. A wearable, wireless gaze tracker with integrated selection command source for human-computer interaction. *Information Technology in Biomedicine, IEEE Transactions on 15*, 5 (2011), 795–801.

[61] Rigas, Ioannis, Raffle, Hayes, and Komogortsev, Oleg V. Hybrid ps-v technique: A novel sensor fusion approach for fast mobile eye-tracking with sensor-shift aware correction. *IEEE Sensors Journal 17*, 24 (2017), 8356–8366.

[62] Rostaminia, Soha, Mayberry, Addison, Marlin, Benjamin, Ganesan, Deepak, and Gummeson, Jeremy. ilid: Low-power sensing of fatigue and drowsiness measures on a computational eyeglass. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (2017), UbiComp '17.

[63] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature 323*, Oct (1986), 533–536+.

[64] Ryan, Wayne J, Duchowski, Andrew T, and Birchfield, Stan T. Limbus/pupil switching for wearable eye tracking under variable lighting conditions. In *Proceedings of the 2008 symposium on Eye tracking research & applications* (2008), ACM, pp. 61–64.

[65] Sarkar, Niladri. Eye-tracking system and method therefor, June 16 2016. US Patent App. 14/966,733.

[66] Schmitt, Lauren M, Cook, Edwin H, Sweeney, John A, and Mosconi, Matthew W. Saccadic eye movement abnormalities in autism spectrum disorder indicate dysfunctions in cerebellum and brainstem. *Mol Autism 5*, 1 (2014), 47.

[67] Seiple, William, Rosen, Richard B, and Garcia, Patricia M T. Abnormal fixation in individuals with age-related macular degeneration when viewing an image of a face. *Optom Vis Sci 90*, 1 (Jan 2013), 45–56.

[68] Sewell, Weston, and Komogortsev, Oleg. Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems* (2010), ACM, pp. 3739–3744.

[69] Srivastava, Anshul, Sharma, Ratna, Sood, Sanjay K, Shukla, Garima, Goyal, Vinay, and Behari, Madhuri. Saccadic eye movements in Parkinson's disease. *Indian J Ophthalmol 62*, 5 (May 2014), 538–44.

[70] `http://www.st.com/stm32`. "STM32 32-bit ARM Cortex MCUs - STMicroelectronics", Accessed: 2015-06-24.

[71] STM32. STM32 32-bit ARM Cortex MCUs. `http://www.st.com/web/en/catalog/mmc/FM141/SC1169`, 2013.

[72] Stonyman. Stonyman Vision Chip. `http://centeye.com/products/stonyman-vision-chip-breakout-board/`, 2013.

[73] Tan, Kar-Han, Kriegman, D., and Ahuja, N. Appearance-based eye gaze estimation. In *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision, 2002.* (2002), pp. 191–195.

[74] techcrunch.com. Google buys Eyefluence eye-tracking startup. https://techcrunch.com/2016/10/24/google-buys-eyefluence-eye-tracking-startup/, 2016.

[75] The New Yorker. Whats the Problem with Google Glass? https://www.newyorker.com/business/currency/whats-the-problem-with-google-glass, 2014.

[76] Tibshirani, Robert. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), 267–288.

[77] `http://www.tobiiglasses.com/`. ”Tobii Glasses: Mobile Eye Tracker for real world research,” Accessed: 2015-06-24.

[78] Tobii. Tobii EyeX Controller. http://www.tobii.com/eye-experience/, 2013.

[79] Tobii. Tobii Virtual Reality Products. https://www.tobii.com/tech/products/vr/, 2018.

[80] Tobii Technology. Tobii Glasses Eye Tracker. Online, 2013. Online; accessed April 7, 2013.

[81] Tonsen, Marc, Steil, Julian, Sugano, Yusuke, and Bulling, Andreas. Invisibleeye: Mobile eye tracking using multiple low-resolution cameras and learning-based gaze estimation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1*, 3 (2017), 106.

[82] uploadvr.com. GDC 2017: FOVE and AMD Take Aim At Improving VR Rendering. https://uploadvr.com/gdc-2017-fove-amd-take-aim-improving-vr-rendering/, 2017.

[83] uploadvr.com. Oculus Patented New Eye Tracking Device Days After Acquiring The Eye Tribe. https://uploadvr.com/oculus-patented-new-eye-tracking-device-days-acquiring-eye-tribe/, 2017.

[84] `https://www.tobiipro.com/learn-and-support/learn/eye-tracking-essentials/how-do-tobii-eye-trackers-work/`. ”How do Tobii Eye Trackers work?”, Accessed 2017-05-12.

[85] `http://www.tobii.com/Global/Analysis/Marketing/Brochures/ProductBrochures/Tobii_X1_Light_Eye_Tracker_Technical_Specifcation_Leaflet.pdf?epslanguage=en`. ”Tobii X1: Gaze Precision and Gaze Accuracy”, Accessed: 2015-06-24.

[86] Willett, Rebecca, Martin, Aline, and Nowak, Robert. Backcasting: adaptive sampling for sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks* (2004), ACM, pp. 124–133.

[87] Ye, Zhefan, Li, Yin, Fathi, Alireza, Han, Yi, Rozga, Agata, Abowd, Gregory D, and Rehg, James M. Detecting eye contact using wearable eye-tracking glasses. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (2012), ACM, pp. 699–704.

[88] Young, L.R., and Sheena, D. Survey of eye movement recording methods. *Behavior Research Methods 7*, 5 (1975), 397–429.

[89] Yuan, Ming, and Lin, Yi. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*, 1 (2006), 49–67.

[90] Zemblys, Raimondas, and Komogortsev, Oleg. Developing photo-sensor oculography (ps-og) system for virtual reality headsets. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (2018), ACM, p. 83.

[91] Zhang, Lan, Li, Xiang-Yang, Huang, Wenchao, Liu, Kebin, Zong, Shuwei, Jian, Xuesi, Feng, Puchun, Jung, Taeho, and Liu, Yunhao. It starts with igaze: Visual attention driven networking with smart glasses. In *Proceedings of the 20th annual international conference on Mobile computing and networking* (2014), ACM, pp. 91–102.