ELECTRONIC IMAGE DETECTABILITY UNDER VARYING ILLUMINATION CONDITIONS

THESIS

Jeremy J. Miller, Major, USAF

AFIT-ENV-MS-20-M-228

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT-ENV-MS-20-M-228

ELECTRONIC IMAGE DETECTABILITY UNDER VARYING ILLUMINATION CONDITIONS

THESIS

Presented to the Faculty

Department of Systems Engineering and Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Engineering Management

Jeremy J. Miller, BS

Major, USAF

March 2020

AFIT-ENV-MS-20-M-228

ELECTRONIC IMAGE DETECTABILITY UNDER VARYING ILLUMINATION CONDITIONS

Jeremy J. Miller, BS

Major, USAF

Committee Membership:

Dr. Michael E. Miller
Chair

Dr. Michael Murdoch
Member

Maj Steven J. Schuldt, PhD
Member

AFIT-ENV-MS-20-M-228

Abstract

Light in the built environment plays an essential role in the vision and the health of humans through non-visual receptors in the eyes. Unfortunately, image analysts and other Air Force personnel who engage in the detection of objects on softcopy displays are often required to work in very dimly-lit or dark environments as higher illumination reduces the contrast of displayed information. Literature has shown that increases in light exposure improves circadian rhythm entrainment and reduces the negative health consequences of insufficient lighting. This research examines the effects of indoor lighting to determine if increases in ambient illumination or changes to the orientation of light fixtures improves or degrades alertness and visual performance. Positive findings of this study would lead to changes in the environment of image analysts, increasing productivity and long-term health. In a controlled environment, twelve participants were exposed to D65 ambient light at levels of 0, 21, 32, and 43 lux, through overhead lighting, wall-washing, and under-desk illumination orientations. During exposures, participants were asked to visually identify variations in low-contrast Gabor patches on a display. The data was used to calculate the contrast threshold of detectability and response times of participants, thus indicating visual performance. Surveys also measured the subjective alertness and discomfort of participants. Results showed that the orientation of light fixtures significantly affects participant sleepiness, weariness, and discomfort. Additionally, the results indicate that visual detection performance is improved with higher ambient illumination levels employed within this research.

Acknowledgments

I would like to express my sincere gratitude to my faculty advisor, Dr. Michael Miller, for his guidance and support throughout the course of this thesis effort. His insight, experience, and willingness to answer my numerous questions was certainly appreciated. I would also like to thank my committee members, Dr. Michael Murdoch and Maj Steven Schuldt, for their valuable contributions. Additionally, I would like to thank all the members of the 480th Intelligence, Surveillance and Reconnaissance Wing as my sponsor, for both the support and latitude provided to me in this endeavor. I would also like to thank all the volunteers who participated in the experiments. Finally, I would like to thank my family; without their support and encouragement, this would not have been possible.

Jeremy J. Miller

Table of Contents

List of Figures

Page

List of Tables

Page

ELECTRONIC IMAGE DETECTABILITY UNDER VARYING ILLUMINATION CONDITIONS

I. Introduction

General Issue

Many career fields in the Air Force and across the Department of Defense, including intelligence, air traffic control, radiology, and cyber operations, require members to closely analyze details in images or information on electronic displays. These personnel often work in dimly lit or dark work environments. For perspective on the amount of light in these types of working conditions, outdoor illumination on a clear day is approximately 10,000 lux and indoor illumination in typical office environments is 500 lux [1], [2]. However, the National Geospatial-Intelligence Agency (NGA) lighting standards mandate a maximum illumination in softcopy exploitation (digital analysis) facilities of 21 lux, or about 4% of a normal office environment [3]. In fact, many of those facilities have found that the easiest way to meet the strict standards is to operate without any illumination beyond the glow of the electronic displays, leaving the analysts to function almost completely in the dark.

The dark conditions benefit the analysts by removing any glare and improving the contrast on the displays. However, it produces an environment that is not ideal for employee health. Without being exposed to sufficient light humans cannot produce an adequate amount of vitamin D, which is a crucial component of good health. Additionally, our circadian rhythm–or sleep-wake cycle–can be disrupted, a consequence that is only exacerbated by the fact that many of these analysts perform shift work to maintain 24/7 operations. Research has shown that vitamin D deficiency and circadian rhythm disruption are correlated with degraded neurobehavioral performance, cardiovascular problems, immune dysfunction, an increased risk of cancer, as well as increased problems with depression, digestion, alertness and overall well-being [4], [5].

Reducing the glare on the display that could impair the operators' ability to perform image or data analysis is the basis behind the NGA standards for softcopy exploitation. However, those standards were originally developed 16 years ago when cathode-ray tube (CRT) displays were still being used [3]. Today's displays are substantially less reflective, which drastically diminishes the glare produced compared to previous electronic display technology, thus the amount of glare today does not impact the analysis the same as it once did. This calls into question the need for such stringent lighting standards.

Problem Statement

It is critical for intelligence image analysts, air traffic controllers working in transit centers, remotely piloted aircraft pilots, and radiologists analyzing images to accurately view their display so that they can recognize the specific key indicators of their profession, which can often be subtle. At the same time, they need to be able to operate in an environment that does not create any unnecessary health risks and permits them to remain alert during their entire shift.

The current standards only address lighting to focus on mission accomplishment without regard for the effects on human health, sending the message that analysts in these settings should prioritize mission accomplishment over their own health and well-being. They perform their responsibilities without regard to the negative health consequences associated with vitamin D deficiency and circadian rhythm disruption. Without the intelligence community addressing these possible adverse health effects, it increases the potential for operators to suffer from symptoms associated with circadian disruption and alertness at work. Left unchecked, these matters can have many harmful effects on the workplace. For example, employee depression–from a lack of vitamin D or a circadian disruption–leads to three times higher rates of absenteeism from work [6]. Also, lack of alertness or sleepiness decreases attentiveness and makes employees more prone to making errors by as much as 30% to 50% [7]. So even when the operators' performance is prioritized over the negative health effects, there is potential that the overall performance of the intelligence community may not actually be improved due to problems with depression, alertness, or other negative health issues. This thesis posits that the standard operating procedures have not evolved at the same rate as technology has improved, leaving room for improvements regarding the health and performance of image analysts.

Research Objectives and Hypotheses

The basis of this research is the ability to detect subtle image characteristics while operating in a variety of lighting conditions. Since the amount of ambient light in the room can impact the visual capabilities of the occupants when viewing a display, the following hypotheses will focus the objectives of this research.

Using modern technology, such as diffused light-emitting diode (LED) light fixtures and low-reflectance liquid-crystal displays (LCDs), this research hypothesizes that moderate increases in ambient light will increase detectability over predominantly dark environments. In the past, too much light produced excessive amounts of glare that impeded the average operator’s ability to accurately perform his or her image analysis responsibilities. Today, LCDs are designed to virtually eliminate glare and LED lighting provides the ability to reliably and accurately control the desired output color of light and total illuminance level with fine-tuned precision, radically more than incandescent or fluorescent lighting. Also, light sources that are very diffuse will be much less likely to produce glare than light sources that are not diffused well.

Additionally, it is hypothesized that the orientation of light plays an important role in visual detection performance by manipulating the amount of light shining on the display and directly in the operators’ eyes. The orientation and distribution of the light contributes to the amount of physiological adjustment of the eye through the geometric relationship between the light source, the display, and the operators’ field of vision. If a light source is oriented so that it cannot be reflected back to the operator, then it will explicitly impact his or her performance during visual tasks.

Finally, this research hypothesizes that increases in total ambient light in an environment increases analyst alertness and reduces discomfort associated with the eye’s accommodation between the bright display and the dark surroundings. With the benefits of modern technological advances, it is possible to employ moderate sources of ambient illumination to reduce the potential for negative health consequences without impacting the operators’ visual performance. Modifying the total light in the setting may achieve a better balance between performance and health.

Research Focus

To address the hypotheses in this research, this study will reference literature from previous investigations to form the basis for determining moderate levels of illumination that are appropriate. Particular areas of literature review include the influence of light on health, the human circadian rhythm, light in built environments, and visual performance in the field of radiology. The aim is to generate a variety of realistic lighting conditions that reduce glare through diffused light sources and different orientations that can be tested in a controlled environment. This research will also monitor short-term effects of light on health, such as headaches and eye discomfort, as a potential indicator of the impact on more serious and long-term health consequences.

Investigative Questions

Q1. Using diffused four-color LED lighting tuned to a correlated color temperature of 6500K and a NEC Multisync P212 LCD display, does increasing horizontal illuminance levels decrease visual detection performance?

In a dark environment, electronic displays are much brighter than the rest of the surroundings, which can cause eye strain and limit softcopy analysis results in critical discovery situations. This study expects to find that as the horizontal illuminance in the room approaches brightness levels closer to that of the display, the visual performance will increase. As the ambient light continues to increase there will be an inflection point relative to the characteristics of the display (i.e. reflectance) such that further brightness will start to diminish visual detection performance.

Q2. With equal total illumination, does shifting overhead lighting levels to more wall washing and under-desk lighting increase operators' visual performance?

Raising the ambient light solely from overhead lighting is expected to negatively impact the visual performance of operators using an electronic display at a much lower illuminance level than wall washing and under-desk lighting. In other words, it is expected that increasing the amount of ambient light through a combination of wall washing and under-desk floor illuminance will allow much higher illuminance levels before negatively impacting the visual performance of operators.

Q3. Do moderate increases in ambient lighting effectively enhance analysts' alertness and decrease symptoms of discomfort?

Human beings are active during the day because our internal psychological and physiological responses make it so that we are programmed to sleep when it gets dark. As such, working in the dark naturally encourages sleepiness. This research expects that moderate increases in illumination will increase the alertness of occupants in the environment. Additionally, it is expected that indications of discomfort will decrease as the ambient light level rises to more closely match the brightness of the display. Further increases in illumination significantly beyond that of the display will begin to reestablish signs of discomfort.

Methodology

This research will be conducted in a laboratory setting that emulates a generic image analyst's workspace. This controlled environment will use four-color LED lighting tuned to a correlated color temperature of 6500K, which is a neutral white color that is specified for use in facilities designed for image analysis. The LED light fixtures will be diffused to remove any point sources and possible causes of glare. Additionally, the LEDs will be individually addressable in three different zones: overhead, across the entire wall behind the operator's display, and under the desk directed toward the floor. Using LEDs in this way will enable researchers to manipulate the lighting conditions according to output levels and orientation of the light source. Experimenting under those varying lighting conditions, a sample of volunteers will be asked to identify specifically designed test patterns projected on the display, with increasingly subtle differences in contrast. The volunteers will also be surveyed to determine their subjective sleepiness level and symptoms of discomfort experienced.

The data from this experiment will be used to determine the minimum contrast level at which participants are able to correctly identify the patterns presented on the display. By comparing the minimum contrast threshold of the various lighting conditions, this research will be able to model how operator visual performance changes in relation to ambient lighting and orientation conditions. In addition, by analyzing the reported subjective sleepiness level and symptoms of discomfort, the researchers will be able to discover any correlations between ambient lighting and alertness. It will also broaden the understanding of the influence on short-term discomfort effects, which may be an indication of longer-term health concerns.

Assumptions and Limitations

This study is limited to a simulated environment of a generic image analyst's workstation with one computer and one display. The hypotheses are not being tested with an experiment conducted in an actual operating environment, where multiple workstations would be arranged together. This research assumes that the results from the simulated workstation will realistically represent what will happen in a standard operating environment with the accumulation of additional workstations. It assumes that if using equivalent displays and LED lighting with the same output illumination and configuration, there will be negligible differences in the real-world softcopy exploitation operations.

Another limitation of this research is that the experiment is of a reduced duration that is much shorter than an operator would experience over the course of a single work shift. The associated assumption of the study is that any effects observed in the experiment would be magnified when measured across multiple workdays.

Additionally, this study was time constrained, which prevented in-depth investigation into the effects the lighting conditions in this experiment have on the circadian rhythm or other long-term health concerns associated with vitamin D deficiency. It assumes that any lighting condition that reduces short-term discomfort is less likely to carry the potential for serious negative health consequences.

Implications

The results of this study can be used as justification to update or validate existing lighting standards for softcopy exploitation. If the research goes well it will reveal possible lighting conditions that are conducive to creating a better balance of decreasing the image analysts' exposure to negative health effects while maximizing their potential visual recognition performance in critical discovery situations. It will provide an improved operational environment that does not carry such a significant risk of depression and lack of alertness for the operators. It will mean that the Air Force and the Department of Defense will be able to leverage technologies in an innovative way to provide working conditions that do not jeopardize performance, while addressing the health concerns of the nation's service men and women.

II. Literature Review

Chapter Overview

The purpose of this chapter is to cover much of the scientific research performed that has provided an understanding of how we as humans are affected by the light around us and how that light can be harnessed within our built environment to maximize the potential benefits. This chapter first explores previous scientific research from a holistic perspective to better appreciate the complexity and nuances of the seemingly simple topic of light and its effects on the human body. Armed with this knowledge, this chapter will conclude with a discussion of the research gaps and the focus for this thesis.

Background

This section covers certain foundational topics that need to be to understood to provide a better appreciation for more recent research, such as the circadian rhythm, the melatonin hormone, and the internal workings of the human eye.

Circadian Rhythm

The human body runs on an internal biological clock that follows the daily pattern of light and darkness experienced across the globe. This core cycle is called the circadian rhythm, which repeats every 24 hours [8]. It is the reason we are alert during the day and drowsy at night. This natural, built-in rhythm is the internal process that determines our sleep-wake cycle and has a major impact on our performance, core body temperature, and production of hormones [9].

Melatonin

One of the hormones influenced by our circadian rhythm is melatonin. Since its first discovery in the late 1950s, extensive research has been performed regarding the role melatonin plays in the human body [10], [11]. Research has shown that melatonin follows a strikingly similar internal oscillation pattern to that of the circadian rhythm [12]. The distinct daily pattern of melatonin concentration in the blood peaks at night and is nearly absent during the day [10], [13]. The hormone is linked with sleep and plays an essential role in the regulation of circadian and seasonal rhythms [10], [11], [13].

The Human Eye

The human eye is sensitive to a narrow subset of the electromagnetic spectrum (i.e., 380 to 730 nanometers) called the visible spectrum [14]. The energy received by our eyes at the wavelengths of the visible spectrum are what we know as light. When light reaches our eyes, it is captured by photoreceptors, e.g., rods and cones, stimulating the body's visual system [14]. This allows us to visualize our surroundings and perceive color. Sandwiched between the shorter wavelengths, such as x-rays and ultraviolet, and the longer wavelengths, such as infrared and microwaves, this narrow band allows us to perceive images; in other words, it enables us to see.

The relationship of light and our eyes goes well beyond just affecting image perception. Considerable research efforts have investigated the non-image-forming effects of light on the body, or the processes other than vision that occur when light enters the eye. That research has shown than the human eye plays a larger role in our health and well-being, including impacting our alertness, working memory, pupil reflex, sleep quality and heart rate [15], [16].

Photoreceptors

The primary vision photoreceptors are the rods and cones. These photoreceptors are connected to retina ganglion cells within the retina. Each of the retinal ganglion cells collect signals from large numbers of rods and cones, creating signals that permit us to perceive relative light level as well as blue/yellow and red/green color signals, which are transmitted through the optical nerve to the portions of the brain associated with image perception.

In two independent studies in 2001, researchers discovered that a small portion of the retinal ganglion cells additionally function as another kind of photoreceptor that promotes non-image-forming effects [17], [18]. The purpose of these cells, named the "intrinsically photosensitive Retinal Ganglion cells" (ipRGC), is to capture light and create signals that are transmitted to activate the system that controls the circadian rhythm [9]. The discovery of ipRGCs and their role in circadian entrainment produced the recognition that the function of the eyes goes beyond enabling vision, as shown in Figure 1 [19]. The combination of all photoreceptors grants the human eye a dual purpose of enabling vision through image-forming processes and maintaining well-being through non-image-forming effects [20].

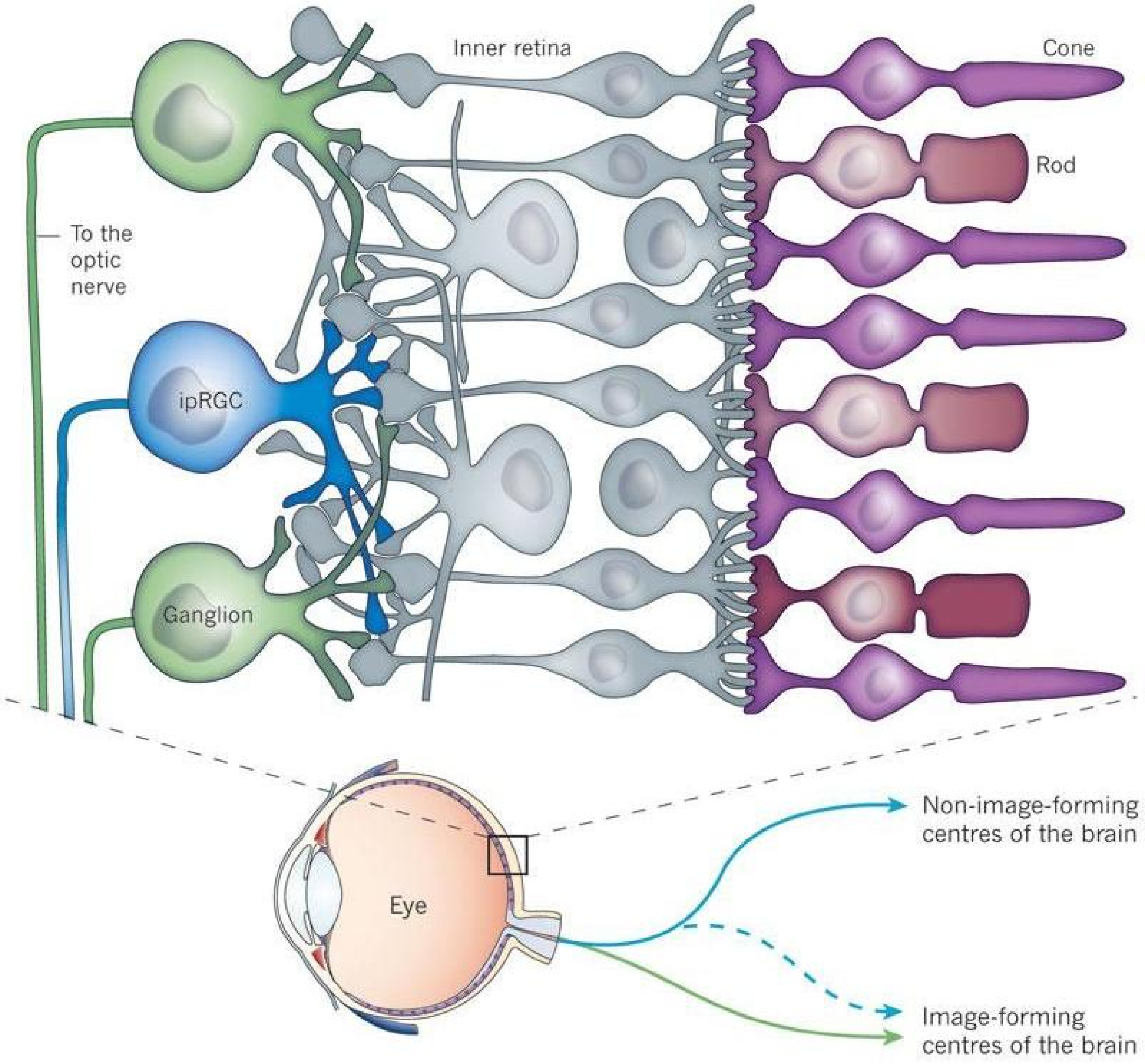
Inner retina
Cone
Rod
To the optic nerve
ipRGC
Ganglion
Eye
Non-image-forming centres of the brain
Image-forming centres of the brain

Figure 1. Close-up Depiction of the Photoreceptors in the Human Eye [21]

One of the most recognizable non-image-forming effects is physiological alertness. As stated above, the hormone melatonin is linked to sleep, and the amount in our body is higher at night than during the day. The regulation of this hormone provides a potential explanation for why we are alert during the day and drowsy at night [10], [11], [13]. It is widely known that light can suppress the total amount of melatonin that is released into our blood stream by the pineal gland, thus controlling alertness [20], [22]. Studies have shown that the ipRGCs are the primary system for light-stimulated melatonin suppression [17], [18]. Moreover, by testing various sources of light and measuring the melatonin response through saliva and blood samples, researchers have found that the ipRGCs have a peak sensitivity to wavelengths around 450 to 480 nanometers [1], [9], [17], [23]. An additional finding concluded that the intensity or magnitude of the light greatly influenced the outcome, with brighter lights leading to faster melatonin suppression, and thus more alertness [1], [4]. Consequently, melatonin suppression is most responsive to bright, short-wavelength (blue) light [1], [9], [16]–[18].

Light Orientation

Not only are the ipRGCs sensitive to the magnitude and spectral wavelength of light entering the eye, but they are also sensitive to the direction the light comes from, or the orientation of the light. The direction of light infiltrating the human eye is significant with regard to the scale of the non-image-forming effects [20].

In a 2003 study, participants were exposed to varying light orientations to investigate the effects of exposure on the inferior (lower) and superior (upper) halves of the retina exclusively. Results of the study showed that inferior retinal exposure led to 25% to 35% more melatonin suppression than the superior retinal exposure [20]. Additional studies have also clearly demonstrated the same sensitivity for stronger melatonin suppression in the lower part of the retina than in the upper part of the retina [1], [24]. This data implies a higher concentration of the ipRGCs in the inferior human retina, meaning that light focused on the lower half of the retina has a greater impact on the non-image-forming effects in the body, such as melatonin suppression [25].

Similarly, in 2005 scientists conducted a study to further investigate the distribution of ipRGCs within the retina [26]. In this study researchers specifically targeted the nasal (toward the nose) and temporal (toward the temple) halves of the human eye. Despite equal exposure characteristics, the data showed that nasal illumination of the retina suppressed the melatonin response more effectively than the temporal illumination [26]. These results have also been demonstrated in other studies as well. Manipulating the melatonin rhythm has significantly greater outcomes when the nasal part of the retina is exposed to light [27]. The results of these studies imply that the nasal half of the retina holds a higher concentration of ipRGCs compared to the temporal side.

The direction of light substantially influences the non-image-forming effects, but has not been given considerable attention within the literature. However, the studies that have explored these conditions, have found a significant relationship. The two areas in the human eye that are more effective than other areas regarding melatonin suppression are the inferior and nasal portions of the retina [20], [26]. Therefore, the highest density of ipRGCs, and thus a greater influence on non-image-forming effects, occurs in the inferior nasal region of the retina [26].

Relevant Research

Light is Essential to Our Health

The amount of light we are exposed to has consequences on our health. Skin cancer is often the only consideration with regard to the effects of light on human health. However, while skin cancer is a risk, the impact of light goes well beyond the surface of our skin, deep into our core. Generally, sunlight exposure increases a feeling of well-being [5]. In fact, through the ipRGCs light is able to affect our physical, physiological, and psychological behaviors [9]. The pathological response of those behaviors from the non-image-forming effects of light are just as important, if not more important, than the visual, image-forming effects, as light is one of our basic needs [9], [20].

An Indoor Society

Thanks to the advent of artificial light in 1879 and the industrial boom of the late nineteenth century, human beings have forsaken outdoor environments and now spend considerably more time indoors, particularly in modern, developed countries. In Western society, a large majority of people spend 80% to 90% of their lifetime indoors [1], [4], [20], [28]. An extreme example is in a 1994 study when scholars examined the quantity and duration of exposure to light of middle-aged adults living in San Diego [29]. Having moderate outdoor temperatures and being in the top 20% of all cities in the United States in terms of the amount of sunshine, San Diego has a culture known for spending a lot time outdoors and having a lot of experience with sunlight [29]. Surprisingly, the study concluded that the average middle-aged adult living in San Diego only spends 4% of each day, or just under an hour, in bright light equivalent to daylight (greater than 1,000 lux). Additionally, 50% of their time is spent in dim light (between 0.1 to 100 lux) and nearly 39% of their time is spent in light less than 0.1 lux, such as sleeping or driving at night [29].

The San Diego example shows that reducing the amount of our life spent outdoors to less than an hour each day, greatly impedes the total quantity of vitamin D that our bodies can produce from harnessing the sun’s rays, leading to the vitamin D deficiency seen throughout much of the world. In addition to San Diego, vitamin D deficiency has been documented in Northern Europe, the Arabian peninsula, the Mediterranean region, and Australia [5], [30], [31]. Research conducted in Italy in 2014 showed that a high number of adolescents had insufficient vitamin D levels, with 82% of those classified with the more serious designation of vitamin D deficiency [31]. Similarly, a 2016 study reported a high prevalence of vitamin D deficiency that required attention in 14 European populations, including Ireland, Germany, UK, Norway, Iceland, and Finland [31]. Additionally, various studies from 2013 through 2017 revealed a predominance of vitamin D deficiency covering the entire population of Saudi Arabia, and that 64% of participants in Qatar exhibited vitamin D deficiency, with similar results in the United Arab Emirates and Kuwait.

Vitamin D

As humans, we have the ability to generate vitamin D and historically most of the vitamin D in our body came as a result of contact with the light from the sun [30]. Similar to a plant's photosynthesis, our bodies rely on wavelengths from sunlight (i.e., 290 to 315 nanometers) to convert certain forms of cholesterol in our skin into vitamin D, a crucial ingredient for our overall health [5].

In addition to society's lifestyle that has developed to functioning more indoors, the physical characteristics of our environment can restrict the vitamin D production in our body. Characteristics such as significant air pollution, higher latitude, or light passing through glass and plastic all diminish our exposure to shortwavelength energy and thus the formation of vitamin D [5].

The industrial revolution of the 1600s and 1700s brought about a transformation to local communities as it swept across Northern Europe, where buildings were built in closer proximity and an increase in coal burning blanketed large cities in a cloud of air pollution [5]. Out of that gloom emerged a bone disease, called rickets, that caused skeletal deformities and stunted the growth of children [5], [31]. A study conducted in 1822 concluded that the higher incidence of rickets in urban children compared to rural children was because of their lack of exposure to the sun, exacerbated by the heavy air pollution [5]. By the end of the 19th century, rickets had grown to affect upwards of 80% to 90% of children in Northern Europe and in the Northeastern United States [5].

In addition to air pollution, our exposure to sunlight is also affected by the angle of the sun in relation to our environmental location. The risk of vitamin D deficiency increases in geographical areas with little sunlight for most of the year, such as locations at high latitude [5], [30]. During the winter months at latitudes greater than 33 degrees above or below the equator, sunlight travels farther through the ozone layer of the atmosphere, and consequently much of that sunlight is absorbed before reaching the earth's surface [30], [32]. The fact that rays from the sun travel farther through the atmosphere at higher latitudes, in combination with the fact that the duration of daylight hours is reduced in winter, decreases direct sunlight that reaches our skin, affording minimal, if any, vitamin D production [30], [32]. Even in sunny Australia, scientists showed that vitamin D levels in men older than 70 years of age were lowest during the winter [30].

Regardless of whether the source of the vitamin D deficiency is air pollution, geographical latitude, sunscreen, or spending too much time indoors, the impact on human health can be monumental. Studies have linked vitamin D deficiency with rickets, as already discussed, and other skeletal conditions such as osteoporosis [30], [31]. Also, chronically low levels of vitamin D leads to an increased risk of hypertension and cardiovascular disease [5], [30], [31]. In addtion to the skeletal and circulatory system impacts, vitamin D deficiency has shown to influence our mental and neurologic health with an greater propensity for seasonal affective disorder, depression, neurocognitive dysfunction, declined cognitive abilities, Alzheimer disease, dementia, schizophrenia, and epilepsy [5], [30], [31]. Moreover, it can cause metabolic problems including predisposed glucose intolerance, cardiometabolic syndrome, and type 2 diabetes [5], [30], [31]. Insufficent vitamin D in the body also is a risk factor for immune disorders such as some cancers, type I diabetes, and multiple sclerosis [5], [30], [33], [34]. Several studies support the hypothesis that cases of influenza infections drop in the summer and near the equator because of the higher levels of vitamin D [5]. Increased concentrations of the vitamin lower the risk of infectious disease, tooth loss, and can also decrease cancer mortality [5].

Response to Artificial Light

The amount of natural sunlight we experience can have profound influence on our life, but it is not the only light that affects our health. The artificial light sources we use indoors have been shown to trigger physiological responses [35]–[37]. Our average internal core body temperature rises and falls with measured regularity, corresponding to the day-night cycle. A study conducted in 1991 exposed participants to bright lights (5,000 lux) and dim lights (50 lux) at night, while monitoring their core body temperature [35]. Researchers found that while the dim lights had no effect on the participants, the bright light administered at night significantly increased the participant's average core body temperature, preventing the typical temperature decrease that humans experience throughout the course of a night [35].

The daily oscillations of body temperature in humans also closely resembles the day-to-day fluctuations of our resting heart rate [36]. Scientists demonstrated the effects of light on our heart rate by having participants lie down in the dark for 20 minutes and then changed the lighting levels to 150 lux [36]. The light exposure increased the resting heart rate of the participants, especially when performed early in the morning [36], [37].

Beyond the changes to our core body temperature and resting heart rate, artificial light affects our alertness. Recent studies have concluded that long-wavelength (red) light can enhance brain activity both during the day and at night [38]. In 2000, researchers exposed participants to light early in the night, after having spent several hours in the dark [39]. As a result, participants reported a reduction in sleepiness, which was positively correlated with the amount of melatonin suppression from the light [39]. The findings indicate that even modest ambient room illumination (90-100 lux) has considerable effect on alertness [39].

Photoentrainment

Both natural and artificial light in our environment controls our circadian rhythm, or the daily alert-drowsy cycle, through a process called photoentrainment [16]. The circadian rhythm, or biological clock, is photosensitive and is synchronized to the 24-hour day by the perpetual alternation of light and darkness every day [19], [22]. These non-image-forming responses of synchronizing the circadian rhythm, the increase in heart rate, and the elevation of core body temperature happen when light interacts with the ipRGCs [16]. Since the ipRGCs are the primary system for light-stimulated melatonin suppression, as discussed, melatonin and its rhythmic fluctuations play a leading part in synchronizing the circadian rhythm [1], [17], [18].

Occasionally we can disrupt our circadian rhythm by upsetting the harmony of our biological clock and the day-night cycle with irregular timing of light exposure. This can occur through things like rapidly crossing multiple time zones as is the case with long-distance air travel [9], [18]. The natural circadian rhythm can also be disrupted by night shift work, or by being exposed to light when it is not expected by the body, such as while sleeping [21] [18]. Additionally, sleep disorders such as insomnia and restless leg syndrome can cause disturbances to the circadian rhythm [9].

When the body's circadian rhythm is out of balance, the consequences can degrade our individual health. Studies have linked disruptions to the photoentrainment of the circadian rhythm to our neurological health by increasing irritability and depression, and degrading our neurobehavioral performance through impaired learning, memory disruption, and cognitive confusion [1], [4], [9], [20], [21]. In addition to neurological problems, disrupting our circadian rhythm can lead to problems with digestion, gastro-intestinal distress, and some metabolic problems, such as type II diabetes [1], [4], [9]. Moreover, it can increase the risk for certain heart diseases and cardiovascular problems, as well as immune system deficiencies to include breast cancer [1], [4], [15], [20]. Circadian disruption can also decrease our alertness, increase our sleepiness during the day, and cause fatigue, insomnia and sleep deprivation [1], [4], [9], [15], [20]. Directly related to our alertness, disruptions can lead to a decline in performance and an increase in error rates [1], [20]. Without the proper entrainment for our circadian rhythm our body can experience weight gain and an overall impaired well-being [4], [20]. [15]

Light Therapy

Either to combat the effects of vitamin D deficiency or disruptions to circadian rhythm, light therapy, or clinical exposure to high doses of direct light, has been used for over 100 years. One of the first uses was in the early 1900s when doctors recommended sun exposure and sunny environments to treat and prevent tuberculosis and upper respiratory tract infections [5]. Later on in the 1920s, various treatment methods were tried to counteract the spread of rickets through the Northeastern United States. One such method involved exposing patients to direct sunlight on the roof of a hospital in New York City, which resulted in significant improvement in those children diagnosed with rickets [5]. Subsequently, those treatments evolved to using a mercury arc lamp instead, with the same dramatic skeletal improvement over several months [5]. By the end of the 1930s, sensible sunlight exposure combined with vitamin D fortified milk had become widespread enough to essentially eradicate rickets altogether [5]. The efficacy of short-wavelength light to prevent or mitigate the effects of vitamin D deficiency is well documented [30]. The past 30 years of biomedical research literature confirmed that light influences virtually all human physiological and behavioral processes, either directly or indirectly, and promotes good health and overall well-being [4].

Light therapy can also be effectively used to treat individuals suffering from circadian disruptions. Studies have shown that the wide range of physiological responses associated with exposure to short wavelengths includes the functioning to regulate and adapt the biological clock by resetting the timing associated with the circadian rhythm [9]. The results of these studies indicate that exposure to light is an effective way to reset the circadian timing to treat the circadian misalignment issues related with jetlag, adaptation to shift work, and space missions [18], [19], [28], [40]. Circadian entrainment with light therapy has also been used to treat seasonal affective disorder, depression, and certain types of sleep disorders [18], [19], [28], [40]. It can also be used to improve performance, increase alertness, and reduce accidents on night shift tasks [18], [28], [35]. Since this regulation is based on exposure to short-wavelength light, specific lighting solutions (i.e. LEDs) can be utilized to achieve appropriate circadian adjustment to realize the benefits of light therapy [40].

Sick Building Syndrome

In addition to impacting our health, light can have a substantial influence in the workplace. Because we spend the vast majority of our time indoors, the light we receive within the built environment has become a principal concern for researchers, and many studies have explored the topic of lighting in a healthy environment [1], [28]. Utilizing lighting strategies that are designed to maximize the health benefits in building specifications should have the upmost importance. Focusing on lighting in the built environment has been proven to lead to healthy working environments [1], [28].

One example of not designing to utilize effective lighting strategies is the phenomenon known as sick building syndrome. Recognized by the World Health Organization, sick building syndrome is characterized by nonspecific feelings of malaise people experience while occupying and perhaps a few hours after leaving a building, but that generally clear up within a few hours [41]. More specific symptoms experienced include irritation of the eyes, nose, throat, or skin, runny nose and eyes, asthma-like symptoms, unpleasant odor or taste sensitivities, mental fatigue, reduced memory, lethargy, headache, dizziness, nausea, and tiredness [41]. A 2006 study regarding the comfort and health within buildings in nine European countries statistically demonstrated that there were more unacceptable buildings than there were acceptable buildings [1].

While many of the symptoms listed may seem minor, with no lasting effects after leaving the affected building, the influence on employees is not minor, including reduced voluntary overtime and increased staff turnover [41]. The impact of even slight symptoms is magnified by prolonged shift lengths, such as in health-care facilities and 24-hour service-based organizations [1]. Additionally, it increases employers' liability to lawsuits from workers who have become ill as a result of their workplace [41]. Sick building syndrome has also shown significant association with increased lack of mental well-being, depression, and burnout [28]. The symptoms experiened by employees are cumulatively enough to increase their sick time use and therefore absenteeism [41].

Even when employees are at work, organizational performance suffers as a result of sick building syndrome. In addition to the distress of workers feeling the effects of the symptoms, it causes widespread productivity loss from the time spent filing complaints and addressing those complaints [41]. Individual task performance is also directly affected and reports show that many sufferers have noticed a 20% reduction in productivity levels [20], [41].

While there are many potential causes of sick building syndrome suggested in literature, one relationship is that individuals working in areas with a view of exterior windows are less likely to experience symptoms [41]. Additionally, there is a connection between tinted or glazed windows and increased frequency of reported symptoms [41]. Those two relationships may seem to contradict one another, but it implies that the window glazing does not allow enough short-wavelength light through to provide any noticable benefit. As far as the non-image-forming effects are concerned, window glazing effectively removes the window from consideration. Among other factors, ambient light clearly plays a part in designing healthy building environments. One reason for that is because lighting influences not only physiological health, as discussed, but also an occupant's visual comfort.

An individual's subjective pyschological satisfaction with their visual environment is defined as their visual comfort [42]. The amount of visual comfort or discomfort affects task performance, workplace safety, and individual mood [41]. Sustained work under insufficient lighting conditions causes visual discomfort, which triggers eye irritation, eyestrain, and fatigue, and can easily cause sick building syndrome [41]. Visual discomfort is driven by the amount of ambient light compared with required levels for visual task functions, and the quality of the lighting, such as glare or non-uniformly-lit workspaces [42]. Continual adaptation to substantial variations in brightness of nearby parts of the field of view will cause frequent dilation and contraction of the eyes and produce eye irritation and visual discomfort [41]. However, poor indoor illumination is not the only thing that affects our visual discomfort and thus our performance. It is also related to the equipment we use because computer screens and other electronic displays introduce a more dynamic lighting scenario [28].

Displays

The CRT was the primary display used with computers when the computer was introduced in the 1950s. At the time, the CRT technology was still developing and was fraught with issues that caused visual discomfort. One particularly troublesome feature was the amount of glare. A study conducted in 1974 found that the glossy surface of the screen caused excessive reflections from windows and overhead lights resulting in visual discomfort in the form of sore eyes and headaches [43]. In many instances the glare was measured at intensities that exceeded the brightness of the text on the screen, hindering productivity [43]. Another aspect that lead to visual discomfort of CRTs was screen flicker due to the low refresh rate of the display [44]. The extent of discomfort varied among subjects depending on individual physiological differences, but some studies noted that up to 35% of users were bothered by screen flicker [44].

Shortly after the start of the 21st century, new technologies such as LCDs had made CRTs virtually obsolete. The modern LCDs eliminated the screen flicker issue and used non-glossy screen surfaces to decrease the reflectivity and thereby reduce visual discomfort and increase performance [45]. Many studies helped usher in the widespread adoption of the LCD technology. For example, in 1999 CRTs and LCDs were scrutinized for their suitability in accomplishing visual tasks [46]. In this analysis, visual search tasks were carried out on each display under various lighting conditions. The results showed that the subjective ratings of the participants scored LCD twice as high as CRT [46]. Objectively, the LCD reduced the error frequency in the tasks, with 34% less errors than the CRT displays [46]. A similar study using visual search tasks on LCDs and CRTs found error frequency to be 22% lower with LCDs [45]. Overall, these studies demonstrated the higher effectiveness of using the LCD for visual tasks over the older CRT technology [45]–[47]. In a more specific application, LCDs were compared to CRTs under clinical conditions for the purpose of detecting certain lung markings on radiological images [48]. Researchers found that LCDs performed significantly better for detecting certain lung markings than CRTs [48].

One reason that LCDs perform better under various circumstances is the contrast ratio of the display. Due to the technology used in the CRT, when the brightness of the display was increased it caused a pixel spreading, blurring adjacent pixels together and making text more difficult to read [45]. With LCDs, the luminance can be raised dramatically without pixel spreading, effectively increasing the inner contrast between text characters [45]. A study in 2004 noted that higher contrast ratios lead to significant improvements in visual recognition and higher subjective display preference in favor of LCDs [49]. Additionally, studies have found that during experiments using visual search tasks with low contrast and small character sizes, LCDs had shorter search times and less fixations per search than the CRT [50]. Another reason for the improved performance is that LCDs also have lower reflectivity compared to CRTs, which produces less glare to impede productivity [45].

Research Gaps

The research summarized below includes areas of contradiction or areas that have not been fully explored. Focusing on further research in these areas could improve our understanding of the optimal uses of light in the built environment for specific applications.

Light in the Surrounding Field

In the field of radiology, diagnostic radiological images, such as x-rays, moved away from traditional film to soft-copy digital images in the late 1990s [51]. Since then, the radiology image reading rooms were kept at very low ambient light levels to preserve contrast ratio in the dark regions of images, and increase diagnostic performance [52]. An unintended consequence of low ambient light is that it causes multiple variations in the adaptation of the analyst’s eyes when shifting focus from the dark surroundings to the bright display [52]. The continuous adaptation of the eyes can cause eye strain [41].

A study in 2007 examined the performance of participants to detect patterns on a display in two light conditions (1 lux and 50 lux) and various backgrounds of differing reflectance to vary the brightness of the area surrounding the display [52]. It found that with ambient room light of approximately 75-150 lux with a background that has reflectance similar to that of a white board or bright wallpaper, it is feasible to minimize eye adaptation differences and reduce visual fatigue [52]. Additional researchers have also demonstrated that increasing the light on the areas surrounding the display significantly reduced visual fatigue [53].

Increased ambient light of the surroundings also has the potential to influence visual performance. Some researchers suggest that “moderate” ambient lighting, as compared to high or low values, will improve performance [52]. However, the literature provides contrary positions regarding the effects on performance. In 2009, participants were asked to screen radiological images under low (1 lux) and elevated (50 lux) ambient lighting conditions [54]. The results concluded that there was no statistically significant differences in participant performance with regard to ambient lighting [54]. A similar study in 2012 found the same results when testing detection performance at low (1 lux) and elevated (50 lux) ambient lighting conditions [55]. Under each lighting condition, researchers could not demonstrate any statistically significant performance differences among the participants [55].

However, other researchers were able to find significant performance increases under different lighting conditions. One pilot study found that participants increased their probability of detecting subtle bar patterns on a display by 7% after elevating the ambient light from 1 lux to 50 lux, provided that the ambient light matched the eye's adaptation value while viewing the display [52]. Another study noted that when there are large variations between the brightness of the display and the surroundings, the adaptation of the eye causes additional time to glean information from a display [56]. They found that reaction times where significantly lower when the ambient light of the surroundings more closely resembled the brightness of the display [56]. Additionally, the potential benefits have been recognized by the American College of Radiology when they recommended a moderate increase of ambient lighting in reading rooms [54].

Ambient Light

Various researchers have yet to agree on what constitutes a moderate increase in ambient light. In 2009, a study tested the visual performance of participants across a wide range of ambient light levels (50, 500, 1500, 3000, 6000, and 9000 lux) and found that the values at the extremes (50 and 9,000 lux) had significant detrimental effects on visual performance [57]. The researchers did note that the detrimental effects were only seen in individuals over 60 years of age and the ambient light changes had no effect on participants 20-30 years of age [57]. Another study found that reaction times during visual search tasks were significantly faster under 50 lux ambient light than 550 lux [46]. These two studies have a large gap of untested ambient light levels between 50 and 500 lux. This indicates that perhaps 500 lux is more than a moderate increase and too much ambient light to notice any benefit. In 2004, researchers tested participants' visual recognition by identifying patterns on a display under three lighting conditions, 200, 450, and 700 lux [49]. The research concluded that the participants' identification of the patterns was slightly improved for 200 lux versus 450 or 700 lux [49]. Those results seem to contradict an earlier study where participants where reading various colored letters on multiple background display colors under different ambient light conditions [47]. The earlier study established that the participants visually performed better with 450 lux of ambient light than with 200 lux [47]. Additionally, in 2007 scientists observed that when reading radiological images participants performed better, as measured by less false-positive and false-negative findings, at 25 and 40 lux ambient light than at 7, 100, and 480 lux [58].

While these studies differ according to the threshold where performance diminishes with increasing ambient light, they do agree that raising the light levels away from total darkness does improve performance. When the only light in the room comes from the display, performance results are similar to rooms with excess lighting levels [58]. Therefore, sensible ambient lighting design and control is required in our built environment to maximize visual accuracy [58].

Ultraviolet (UV) Light in the Built Environment

Given the widespread vitamin D deficiency seen throughout much of the world and the broad list of negative health consequences associated with it, care needs to be taken to observe the role our artificial environments have in this process. Indoor illumination is focused entirely on the visual aspects of light, but research as proven that light impacts humans in many ways. To address the imbalance between the non-image-forming effects and vision of indoor lighting systems, more attention needs to be given to possibly including additional short-wavelength light sources. Studies have suggested using UV-enhanced lighting for those that are bedridden and have no access to unfiltered sunlight, such as elderly or ill patients [59]. Light therapy such as this has long been in use, but has been limited mostly to clinical environments. The use of UV for general lighting purposes has not been widely investigated. For example, implementing UV-enhanced lighting for homebound patients raises issues for in-home caregivers by increasing their risk of sunburn and skin cancer [59]. While increasing light levels can positively impact human health, performance, and alertness, more research is needed to understand the impacts of UV lighting within general purpose buildings. Areas of potential further study include exposure concentrations and proper application of short-wavelength light to balance the positive effects with the downside of over exposure.

Application Environment

Much of the literature reviewed in this chapter highlights human responses to light, including consequences of insufficient light exposure and how light can be applied to modify individual performance and alertness in the built environment. One striking application of the need to address quality lighting design is softcopy-image analysis in the intelligence community. Due to the nature of the work, analysts operate in windowless buildings and are often required to carry out their duties during extended night shifts.

Digital image analysis in the intelligence community is governed by the narrow NGA standards. According to those criteria the lighting levels in softcopy exploitation facilities cannot exceed 21 lux vertical illumination measured normal to the display, at a correlated color temperature of 6500K. A further requirement is to limit horizontal illumination to less than 21 lux measured normal to the keyboard. Depending on the orientation of the light fixtures, meeting the horizontal illumination constraint often results in much lower vertical illumination measurements; an example of the two measurements used in this research is given in Table 7. Consequently, many intelligence image analysis facilities do not utilize lighting fixtures, but work in the dark, relying solely on the glow from the display. For analysts this magnifies the potential for the negative health consequences discussed in this chapter. It also provides a focus area with considerable room for improvement, where more research can be conducted to challenge the long-held lighting designs that are prevalent in the built environment.

The complexity of workplace lighting hinders the research in this thesis from investigating all factors in depth. For example, the time constraints limit the ability of this study to examine the long-term effects of UV light in an office environment and its impact on vitamin D deficiency. Additionally, the effects of workplace illumination on the photoentrainment of the human circadian rhythm is indirectly analyzed through subjective surveys of sleepiness rather than blood or saliva samples to measure melatonin concentration. Ultimately, the focus of the experiment in this thesis is the influence of total ambient illumination and light orientation on visual performance, alertness, and occupant discomfort.

Summary

This chapter covered much of the scientific body of knowledge gained throughout the years from research performed regarding our understanding of humans are affected by the light around us. It also served to highlight a few areas where additional information can be learned about how light can be harnessed within our built environment to maximize the potential benefits.

III. Methodology

Chapter Overview

The purpose of this chapter is to describe the specifics of the experiment that was conducted. This chapter covers test conditions, equipment, calibration, test participants, and procedures, and how I applied them in order to conduct a streamlined and controlled experiment that could produce reliable and replicable results.

This chapter discusses both luminance and illuminance. Because of the similarity of the two words and their uses, it is necessary to understand the difference between the two. When turning on the light switch in an office, for example, the overhead lights produce a certain amount of light, called luminance. Luminance is measured as the quantity of light given off by an object in a specific direction, as if seen by human eyes, whether the light source is a candle, a lamp, or an electronic display. As the light in the office falls on a desk, for example, some of the light is absorbed and some is reflected off the desk into our eyes, allowing us to see the desk. The amount of light reflected off an object is also called luminance. So luminance is the light a lamp produces, but also the light that reflects off objects. In other words, luminance can be thought of as any light that is leaving a surface, whether produced or reflected. Illuminance on the other hand, is measured as the quantity of light that falls on a surface, before any light is absorbed or reflected. Returning to the desk example, illuminance is the amount of light that impacts an object, but luminance is only the portion of that light that is reflected by the object.

When referring to the quantity of illuminance, this chapter primarily uses the SI unit of lux (lx) consistent with literature, which is the amount of light impinging upon a surface for a given area. For luminance, this report uses the SI unit of candelas per square meter (cd/m^2).

Environment

The experiment performed in this research was accomplished in a traditional office cubicle approximately 2.13 m high by 1.83 m wide by 1.83 m deep. The cubicle was modified to enclose the top with a solid ceiling. Mounted to the entire ceiling and the wall behind the electronic display were full-cubicle width LED ribbon lights, spaced at 7.6 cm apart. The ribbon lights included four colors of LEDs–red, green, blue, and white (RGBW)–in an alternating pattern and each able to be operated independently, as shown in Figure 2.

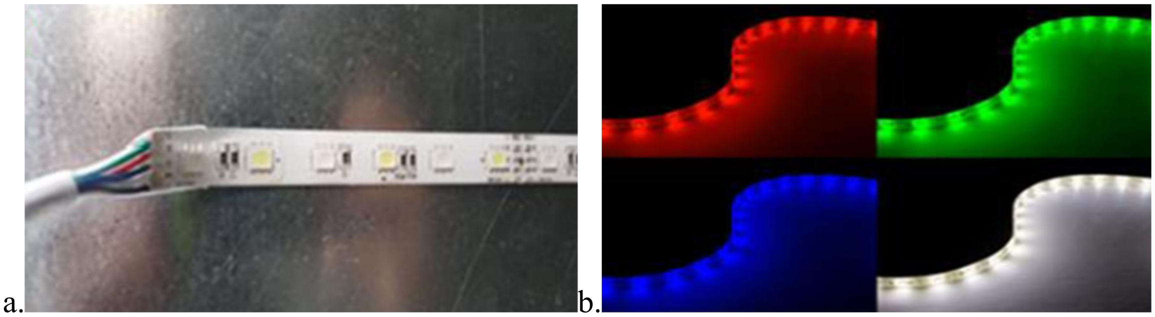

Figure 2. (a) Close up of the LEDs used in the Experiment (b) Demonstration of Each LED Color in Use

A translucent film was placed 7.6 cm in front of the lights, between the LED lights and the participants, to diffuse the light emanating from the LEDs and remove any single point sources. The chromaticity coordinates of each of the four LED colors used in the experiment as well as the D65 target white point is in Table 1 along with the corresponding chromaticity diagram in Figure 3. The doorway to the cubicle was enclosed with a non-glossy curtain, in order to block out any external light and control the lighting in the environment.

Table 1. Chromaticity Coordinates of Each LED and Target White Point

LED Color	x Coordinate	y Coordinate
Red (R)	0.6858	0.3048
Green (G)	0.1550	0.6891
Blue (B)	0.1482	0.0576
White (W)	0.2516	0.2166
Target White Point (D65)	0.3127	0.3290

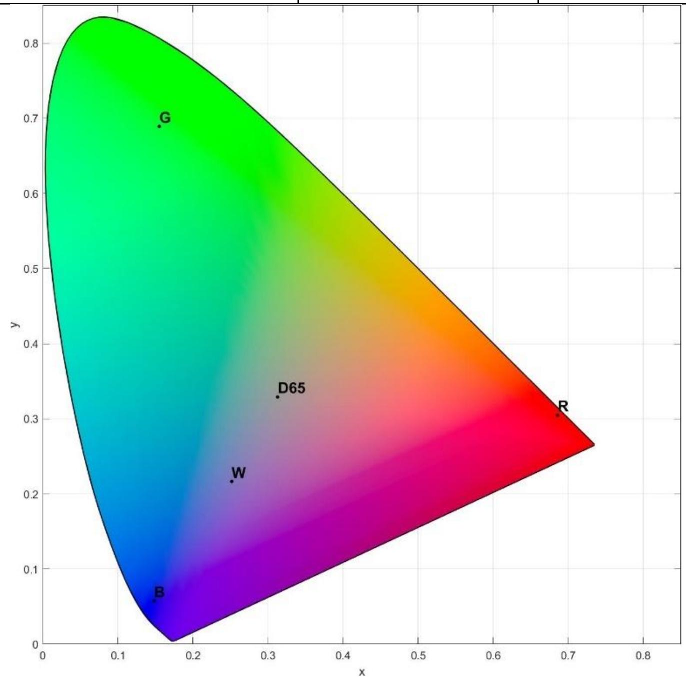

Figure 3. Chromaticity Diagram of Each of the Four LEDs Used Along with the D65 Target White Point

The cubicle contained only a basic cubicle-style desk, an adjustable height office chair, the electronic display, a keyboard, and a chin rest, as shown in Figure 4. The dimensions of the desk were approximately 0.76 m high by 1.52 m wide by 0.61 m deep. The electronic display was a NEC Multisync P212 LCD display.

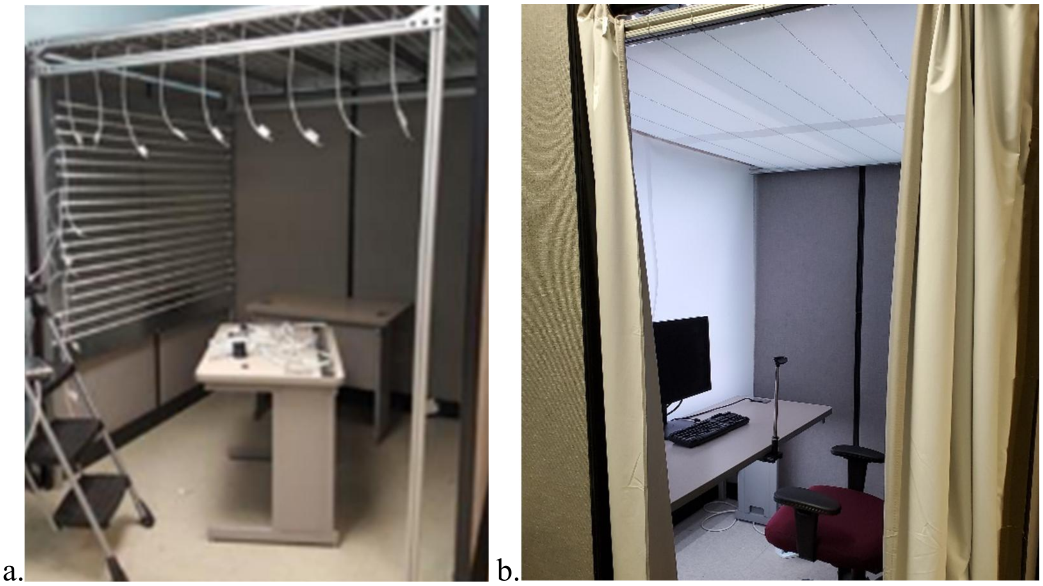

Figure 4. Photo of the Modified Cubicle Used in the Experiment (a) Without and (b) With the Diffusers and Curtain in Place

Table 2. Chromaticity Coordinates of the Display Primary Colors and Target White Point

LED Color	x Coordinate	y Coordinate
Red (R)	0.6414	0.3309
Green (G)	0.3008	0.6003
Blue (B)	0.1509	0.0689
Target White Point (D65)	0.3127	0.3290

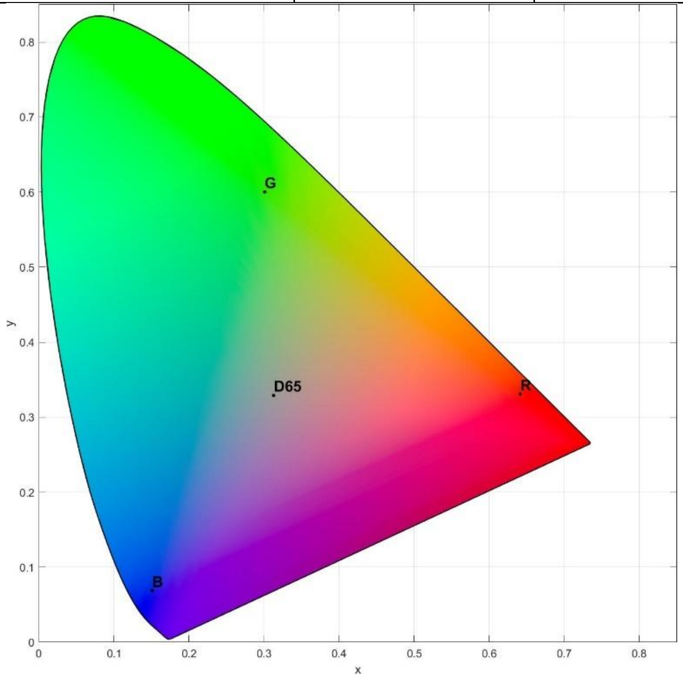

Figure 5. Chromaticity Diagram of the Display's Primary Colors Along with the D65 Target White Point

The chromaticity coordinates of the primary colors in the display used in the experiment as well as the D65 target white point is in Table 2 along with the corresponding chromaticity diagram in Figure 5. The chin rest controlled the participants' proximity to the display at a fixed distance of 0.53 m, and a height between 10 and 15 degrees above the center of the display (depending on the size of the participant's facial features). A diagram of the layout of the environment inside the modified cubicle is shown in Figure 6.

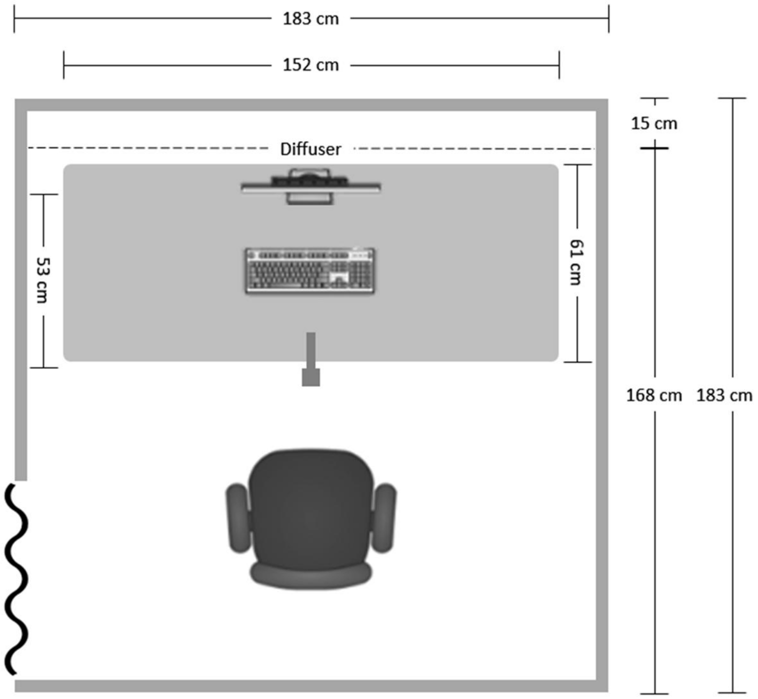

Figure 6. Layout of the Modified Cubicle Used in the Experiment

Conditions

As mentioned previously, the study sought to examine whether the orientation and distribution of light can influence visual detection performance, through a combination of overhead, wall washing, and under-desk illuminance lighting conditions. While these conditions often include more than one source of light, for the purposes of this report they will be referenced by their differentiating aspect and shall be subsequently referred to as the ceiling, wall, and desk conditions. Further clarification on the study conditions can be found in Table 3 and in the paragraphs below.

The ceiling condition was the most straightforward and exhibited the usual condition found in an office environment, consisting of entirely overhead lighting. Different from most office settings, however, is that the lights were diffused LED lighting with relatively even luminance across the entire ceiling area. The LEDs were operated to produce various illumination levels from 0 to 43 lux of horizontal illumination measured at the keyboard.

The wall condition captured the lighting technique of wall washing in that it used diffused LED lighting across the entire wall behind the display, from below the top of the desk to the ceiling. In this condition, the luminance from the wall wash LEDs were set to a constant 40 cd/m^2. This amounted to less than the 21.53 lux of horizontal illumination measured at the keyboard. The overhead LEDs provided the remainder of the ambient light to meet each illumination condition within the experiment.

Building upon that concept, the desk condition also used a constant 40 cd/m^2 wall luminance, but added a constant 40 cd/m^2 luminance from LEDs mounted to the underside of the desktop, shining on the floor. Again, this created a horizontal illumination at the keyboard less than the 21.53 lux condition. The overhead LEDs provided the rest of the ambient light to meet each illumination condition within the experiment.

Combinations of light from these three areas were adjusted to provide four different illumination levels, including a dark condition as well as illumination conditions, which provided the maximum permissible illuminance at the keyboard of 21.53 lux in accordance with the NGA standards, 1.5 times the permissible illuminance, and 2 times the permissible luminance [3]. These illumination levels and the three orientations and distributions of illuminance resulted in 10 experimental conditions, as shown in Table 3.

Table 3. Study Conditions of Light Orientation and Illumination

		Illumination Conditions			
		0.00 lux	**21.53 lux**	**32.29 lux**	**43.06 lux**
Orientation Conditions	**Overhead LEDs only** *(ceiling)*	All lights off			
	Overhead & wall wash LEDs *(wall)*				
	Overhead, wall wash, & under-desk *(desk)*				

Calibration

LED Calibration

A key aspect to the hypotheses of the research is the use of highly tunable LED lighting. In order to sufficiently constrain the LEDs to provide appropriate control to the study, the LEDs had to be properly calibrated. This involved blending each of the four LED colors, red, green, blue, and white (RGBW), to create D65 light. This color temperature was chosen to be equivalent to the color temperature of the display as specified in the NGA standards [3], [60].

The first step to calibrating the LEDs to D65 at the desired illuminance levels for each orientation condition was to gather data on the performance of the LEDs. I used the Photo Research Inc. PR-680L SpectraDuo (a combined spectroradiometer and PMT based photometer) to measure the luminance values of a Spectralon wavelength calibration standard and the Konica Minolta T-10 illuminance meter to measure the illuminance of the workstation. The Spectralon standard was positioned vertically on a tripod directly in front of the center of the display, and the illuminance meter was positioned on the desk in front of the display where the keyboard might normally reside. I then recorded the luminance, illuminance, and CIE chromaticity coordinates for each RGBW LED across their entire range from the minimal programmable code value through full intensity. This set of measurements was repeated for each of the LED locations, the ceiling, wall, and desk LEDs.

The luminance and illuminance data followed specific exponential curves described by the power function in equation 1, where *x* is the luminance or illuminance, *a* and *c* are constants, and *b* is the nonlinear response unique to each LED. By using Matlab's fit function on the measured data points in conjunction with equation 1, I was able to determine the coefficients *a*, *b*, and *c* for the luminance and illuminance of the output curves of all the LEDs. An example of the coefficients of the luminance curves is shown in Table 4. See Appendix A for the Matlab source code that calculates the output curves of all the LEDs.

$$x = a * code\ value^{b} - c \qquad (1)$$

Table 4. LED Luminance Curve Coefficients

	Ceiling Luminance			**Wall Luminance**			**Desk Luminance**		
LED	***a***	***b***	***c***	***a***	***b***	***c***	***a***	***b***	***c***
Red	0.00014	2.80371	0.17401	0.00012	2.08093	0.02062	4.16E-09	4.26331	0.00169
Green	0.00034	2.78984	0.30932	0.00030	2.11571	0.05244	1.06E-08	4.23154	0.00171
Blue	0.00164	1.99735	0.19704	0.00001	2.49504	0.00495	6.13E-09	3.90348	0.00205
White	0.00445	2.30739	1.46480	0.00040	2.22481	0.10217	6.64E-08	3.97038	-0.00242

The other important piece of data collected was the chromaticity coordinates of each of the four LED colors. This information was needed to calculate the ratios of the colors required to produce D65 light. According to Hunt's Center of Gravity Law of Color Mixture, the perceived color of light can be found by computing the weighted average of the x and y coordinates of the input colors [61]. Using Matlab and by working backward from the desired D65 perceived color in conjunction with the x and y coordinates of the input colors, I was able to find the necessary weighted average or ratio of the input colors. I first found the ratios of the red, green, and blue LEDs that would produce a color that is perceived as D65 (R_1, G_1, B_1). I then found the ratios of the red, green, and white LEDs that would produce the D65 color (R_2, G_2, W_2). In order to find the output matrix that uses all four of the LED colors (R_o, G_o, B_o, W_o), I simply combined equal parts of each of the first and second ratios found, as shown in equation 2.

$$[R_o,\ G_o,\ B_o,\ W_o] = [\tfrac{1}{2}R_1 + \tfrac{1}{2}R_2,\ \ \tfrac{1}{2}G_1 + \tfrac{1}{2}G_2,\ \ \tfrac{1}{2}B_1,\ \ \tfrac{1}{2}W_2] \qquad (2)$$

By knowing the exponential curves for luminance and illuminance, along with the ratios of each LED color I was able to determine the code values to program to each LED for the desired illuminance levels for each lighting condition at the proper calibration. For example, the ceiling condition values were calculated using Matlab's solve function on equation 1 by setting *x* equal to the desired illumination condition (21.53 lux) and using the exponential curve coefficients of white illuminance for *a*, *b*, and *c*. This solves for the code value that will produce 21.53 lux of horizontal illuminance. That code value can then be used again on equation 1 with the coefficients of the white luminance curve, providing the output luminance needed to reach 21.53 lux of horizontal illuminance. That output luminance was then split among each of the four RGBW colors using the ratios from Hunt's Center of Gravity Law of Color Mixture and equation 2. Finally, the RGBW ratios were then used in equation 1 in place of *x* and applying the respective color coefficients with Matlab’s solve function to find the code value each LED color necessary to produce a calibrated D65 light. After programming the values, the output was measured using the photometer and illuminance meter. Programming values were adjusted as necessary until proper calibration requirements were confirmed. This process was repeated for the other illumination conditions of 32.29 lux and 43.06 lux. The CIE 1931 2-degree tristimulus values for the ceiling condition at each ambient illumination level is shown in Table **5**, and the corresponding spectral power distributions are included in Figure 7, as an example of the calibration results.

Table 5. Tristimulus Values for the Ceiling Condition

	43.06 lux	**32.29 lux**	**21.06 lux**
X	13.73	10.93	7.65
Y	14.38	11.40	7.60
Z	15.85	13.05	8.73

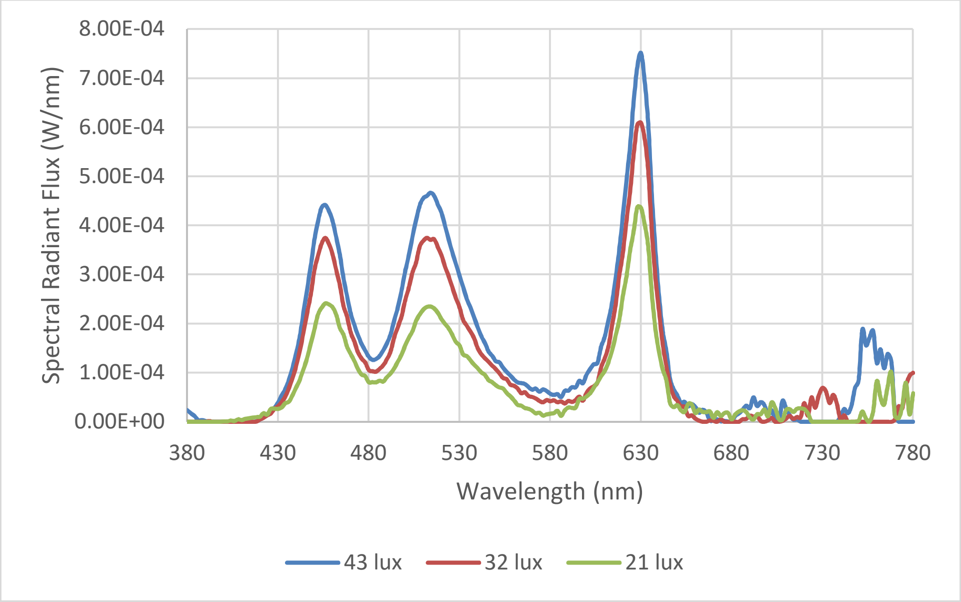

Figure 7. Spectral Power Distributions for the LEDs Under the Ceiling Condition

The wall condition used a similar process to find the calibrated programming values. The difference was that instead of first finding the values for 21.53 lux horizontal illuminance at the keyboard, the target was 40 cd/m^2 luminance of the wall. I then programmed the wall with those values and confirmed that the wall alone was producing D65 luminance of 40 cd/m^2. Using the illuminance meter, I determined that the 40 cd/m^2 of luminance at the wall translated to 17.69 lux of horizontal illuminance measured at the keyboard. In the wall condition, the difference between 21.528 lux and the measured horizontal illuminance, 17.69 lux in this case, comes from the ceiling LEDs. This mathematical difference became the target illuminance for the Matlab calculations to find the ceiling LED programming code values based on equation 1 and equation 2. After programming the values, the output was measured using the photometer and illuminance meter. Programming values were adjusted as necessary until proper calibration requirements were confirmed. This process was repeated for the other illumination conditions of 32.29 lux and 43.06 lux.

Correspondingly, the desk condition followed this same pattern of setting a constant output from the wall and desk LEDs and then measuring the horizontal illumination and adding the overhead LEDs to reach the variable illumination conditions. The values to program the wall LEDs to 40 cd/m^2 were already calculated when calibrating the LEDs for the wall condition. The desk condition then added an additional step of finding the code values for the desk LEDs to output 40 cd/m^2. This was accomplished in Matlab using equation 1 and equation 2. The desk LED output was confirmed with the photometer not at the lighting surface but by placing the perfect diffusing Spectralon standard on the floor directly under the desk LEDs and measuring the luminance at the diffusing standard. Using the illuminance meter, I determined that the 40 cd/m^2 of luminance at the wall plus the 40 cd/m^2 of luminance at the diffusing standard under the desk translated to 18.55 lux of horizontal illuminance measured at the keyboard. In the desk condition, the difference between the first illumination condition, 21.528 lux, and the recently measured horizontal illuminance, 18.55 lux, came from the ceiling LEDs. This mathematical difference became the target illuminance for the Matlab calculations to find the ceiling LED programming code values based on equation 1 and equation 2. After programming the values, the output was measured using the photometer and illuminance meter. Programming values were adjusted as necessary until proper calibration requirements were confirmed. This process was repeated for the other illumination conditions of 32.29 lux and 43.06 lux. See Appendix B for the Matlab source code that calculates the calibration of the LEDs for each illumination condition.

Display Calibration

The second key aspect to the hypotheses of the research is the use of modern displays. Thus, ensuring proper calibration of the display was necessary to provide appropriate control to the study. This involved adjusting the settings of the display to form an applicable framework for the display's calibration within the context of the research. The NEC Multisync P212 display had multiple built-in picture modes to choose from, so I adjusted the display to use the ITU-R Recommendation BT.709 picture mode, which is the industry standard for high-definition displays, published by the International Telecommunications Union [62]. This pre-set picture mode situated the white point at D65 and fixed the gamma correction at 2.2. I also adjusted the luminance of the display to 180 cd/m^2 consistent with displays used in the intelligence community.

To verify these display settings, I had to determine the display's gamma correction, or the exponent of the power function that relates to the non-linear manner in which humans perceive light and color. This was accomplished by projecting a series of

2.54 cm squares positioned at the center of the display, with a black background, and measuring the luminance of the squares with the photometer. The measurements were taken for 16 discrete RGB code values of the squares from a white square through multiple grays and ending at black (R=G=B=255, 200, 150, 100, 80, 60, 50, 40, 35, 30, 25, 20, 15, 10, 5, 0). Each time the pattern was repeated by reducing the shade and measuring the new code value with the photometer. By plotting the code values of the square to the luminance value measured with the photometer, I was able to visualize the power function of the gamma correction. The power function is in the form of equation 1, where b is the gamma value, and a and c are constants. Using Matlab's fit function on these data points with equation 1 found that the display's gamma is 2.1912, which is within 0.5% of the 2.2 that it should be for the ITU-R Recommendation BT.709 display setting used, which is the standard for high-definition displays [62]. See Appendix A for the Matlab source code that calculates the gamma correction of the display.

The luminance data from measuring the multiple squares also validated additional calibration settings of the display. The data verified that each square, and thus the white point of the display, had CIE 1931 x and y chromaticity coordinates of 0.313 and 0.329 respectively, which appropriately corresponded to a color temperature of D65. The CIE 1976 $\Delta E_{ab}^{*} = 1.0455$ for the target neutral patches. With the photometer data I also confirmed the desired setting of 180 cd/m^2 luminance of the display.

In addition to finding the color temperature, gamma correction and display luminance, proper calibration of the display also included determining the reflectance of the electronic display used in the experiment to guarantee the study included a low-reflectance display. This was accomplished with the display turned off and all of the LED lights within the modified cubicle turned on to full intensity. Using the photometer, I measured the luminance of the Spectralon standard immediately in front of the display, as well as the luminance of the display while it was turned off. Comparing the luminance values for each LED color, I calculated the percentage of luminance reflected by the display compared to the Spectralon standard. Using the ratios of the LED colors to create D65 light found in the lighting calibration steps above, I was able to determine the weighted average reflectance value. As shown in Table 6, under the conditions used in this experiment the display had a 1.83 percent reflectance.

Table 6. Display Reflectance for D65 Light

LED Color	Reflectance	D65 Ratio	Reflectance x Ratio
Red	1.913%	18.365%	0.351%
Green	1.319%	23.675%	0.312%
Blue	1.994%	55.203%	1.101%
White	2.374%	2.757%	0.065%
		Weighted Average	1.830%

Test Participants

The performed experiment employed 12 research participants between the ages of 23 and 41 years of age (mean: 30.1 years, S.D.: ±5.5). The participants consisted of 11 males and one female. All participants worked for the United States Department of the Air Force; 11 were active duty Air Force officers and one was a General Schedule (GS) civilian employee. All were right-handed and were not smokers. Six participants reported regular caffeine consumption. Participants were required to adhere to their habitual caffeine consumption (mean: 64.2 mg, S.D.: ±82.1) and sleep schedules (mean: 7.4 hours per night, S.D.: ±0.94).

As this research is focused on the visual detection ability, it relied on the eyes of the participants. For that reason, all potential participants were screened for color deficiency and visual acuity. To be eligible, they had to pass the Ishihara Color Vision Test. Additionally, all participants also needed to demonstrate at least 20/25 near field vision by using a Snellen eye chart, with or without corrective lenses. Three of the eligible participants wore corrective lenses and another two had previously undergone eye surgery to improve their visual acuity. Of the 12 participants, five had blue eyes, three had hazel colored eyes, two had green eyes, and two had brown colored eyes. Due to university policies, all participants volunteered their time and received no compensation, which limited the number of participants used in the study. Despite the limitations, the number of participants was consistent with literature on other vision studies [55], [56], [71], [63]–[70].

Test Procedure

The basics of the test required the participants to be seated in the experiment environment with their chin on the chin rest. The display in front of them had a uniform gray color. When the test began, a fixation cross, or “plus sign”, appeared in the center of the screen to direct the participants’ attention. After a random interval between one and 1.5 seconds, the fixation cross disappeared and the display was again a solid color. The display remained devoid of stimulus for a random interval between 1.5 and two seconds, at which time a Gabor patch was presented in the center of the display where the fixation cross used to be, using the Psychophysics Toolbox Version 3 in Matlab. The Gabor patch was approximately 1.9 cm in diameter, and had 16 cycles of high and low intensity. The Gabor patch was presented with a random orientation tilted either 45 degrees to the left or right of vertical, as in Figure 8. The participant was instructed to identify which way the Gabor patch was tilted (i.e., was the top of the pattern pointing towards the left or the right) and to correspondingly press the left or right arrow key on the keyboard. Once a keypress was recognized, the pattern would repeat by alternating between the fixation cross and the Gabor patch.

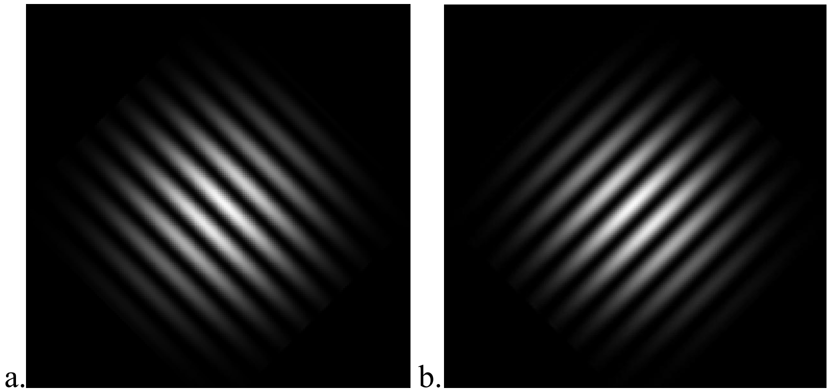

Figure 8. Example Gabor Patch Tilted (a) Left (b) Right

After every Gabor patch, the experiment was adapted based on the participant's responses by using the QUEST function in the Psychophysics Toolbox. The Bayesian adaptive psychometric method of the QUEST function increased or decreased the relative contrast of subsequent Gabor patches presented based on the results of whether or not the participant correctly identified the orientation of the preceding Gabor patch [72]. The QUEST function methodically caused the Gabor patch to be increasingly more difficult to detect by decreasing the contrast for every correct response. Increasing responses caused the QUEST function to increase the contrast value to make the Gabor patch easier to detect. This method allowed the contrast to approach the threshold at which point the participant was no longer able to accurately identify the orientation. Each participant was required to complete 40 evaluations at each illumination condition. By recording the relative contrast of the Gabor patch, the accuracy of the participant's responses, and the vertical illumination for each condition, I was able to determine the threshold of detectability for each illumination condition. Table 7 shows the vertical illumination, as measured normal to the center of the display, for each lighting condition. See Appendix C for the Python source code to control the LEDs when changing illumination conditions.

Table 7. Vertical Illuminance Measured at the Display for Each Lighting Condition

		Horizontal Illumination Conditions			
		0 lux	**21.53 lux**	**32.29 lux**	**43.06 lux**
Orientation Conditions	**Ceiling**	0 lux	12.24 lux	22.11 lux	28.90 lux
	Wall	0 lux	9.75 lux	17.21 lux	23.93 lux
	Desk	0 lux	9.58 lux	17.18 lux	23.67 lux

In the first step after determining the participant’s eligibility with the visual tests, the participants were introduced to the experiment environment and all lights were turned off. While the participants waited in the dark for three minutes to allow their eyes time to dark adapt, they were given a practice scenario of five Gabor patches. See Appendix D for the Matlab source code of the practice scenario. During the practice scenario they were provided feedback on whether they were able to correctly identify the orientation of the Gabor patch. Upon completion of the dark adaptation timer, and armed with realistic hands-on experience, the participants began the experiment for step one. The various sequences of illumination conditions in steps two through 10 was systematically created based on the possible permutations of experiment conditions and then randomly assigned to the participants, as shown in Table 8. Each of the six permutations of the orientation conditions (ceiling, wall, and desk) was assigned to two participants. Once for increasing illumination conditions from 21.53 to 43.06 lux, and once for decreasing illumination conditions from 43.06 to 21.53 lux.

Table 8. Condition Sequencing

Step	1	2	3	4	5	6	7	8	9	10
Subject	**0 lux**	**21.53 lux**			**32.29 lux**			**43.06 lux**		
1	All lights off	wall	desk	ceiling	wall	desk	ceiling	wall	desk	ceiling
3		desk	ceiling	wall	desk	ceiling	wall	desk	ceiling	wall
4		ceiling	wall	desk	ceiling	wall	desk	ceiling	wall	desk
5		desk	wall	ceiling	desk	wall	ceiling	desk	wall	ceiling
8		ceiling	desk	wall	ceiling	desk	wall	ceiling	desk	wall
11		wall	ceiling	desk	wall	ceiling	desk	wall	ceiling	desk
Subject	**0 lux**	**43.06 lux**			**32.29 lux**			**21.53 lux**		
2	All lights off	ceiling	wall	desk	ceiling	wall	desk	ceiling	wall	desk
6		wall	desk	ceiling	wall	desk	ceiling	wall	desk	ceiling
7		desk	wall	ceiling	desk	wall	ceiling	desk	wall	ceiling
9		wall	ceiling	desk	wall	ceiling	desk	wall	ceiling	desk
10		desk	ceiling	wall	desk	ceiling	wall	desk	ceiling	wall
12		ceiling	desk	wall	ceiling	desk	wall	ceiling	desk	wall

At the beginning of steps one, two, five, and eight, participants were required to wait for three minutes for their eyes to adapt to the new lighting condition. During this waiting period at the beginning of steps two, five, and eight, they were given a questionnaire consisting of the Stanford Sleepiness Scale shown in Figure 9 and the discomfort survey shown in Figure 10. The questionnaire was given one additional time following the conclusion of step 10. At the beginning of each of the other steps (3, 4, 6, 7, 9, 10) a shorter one-minute adjustment timer was used. The shorter time was used because despite the difference in orientation condition, those steps have the same total horizontal illuminance at the keyboard as the previous step, thus the participant’s eyes did not need as much time to adapt to the new illumination conditions.

Degree of Sleepiness	Scale Rating (circle one)
Feeling active, vital, alert, or wide awake	1
Functioning at high levels, but not at peak; able to concentrate	2
Awake, but relaxed; responsive but not fully alert	3
Somewhat foggy, let down	4
Foggy; losing interest in remaining awake; slowed down	5
Sleepy, woozy, fighting sleep; prefer to lie down	6
No longer fighting sleep, sleep onset soon; having dream-like thoughts	7
Asleep	X

Figure 9. Stanford Sleepiness Scale

Please indicate to what level you are currently experiencing any of the following:	
Weariness or general fatigue	None Low Medium High Very High
Headache	None Low Medium High Very High
Pain above eyes	None Low Medium High Very High
Dryness in eyes	None Low Medium High Very High
Irritation in eyelids	None Low Medium High Very High
Sensitivity to light	None Low Medium High Very High
Hard to see sharply	None Low Medium High Very High
Blink frequently	None Low Medium High Very High
Eyes watering	None Low Medium High Very High
Dull, aching feeling from eyeballs	None Low Medium High Very High

Figure 10. Discomfort Survey [73]

All steps followed the basic procedure as described by presenting a randomly oriented Gabor patch and waiting for a keyboard press response. For each step, 20 Gabor patches were presented on a dark gray background of 6.3 cd/m^2. Then within the same step, the QUEST function was reset and an additional 20 Gabor patches were presented on a lighter gray background of 72 cd/m^2. Rendering Gabor patches on the dark and then lighter background colors was accomplished for each step shown in Table 8. If any Gabor patch was portrayed on the display for longer than seven seconds without a keypress, a warning tone would sound to remind the participants that a key press was needed. This kept the experiment moving when participants may have been struggling to detect the Gabor patch. See Appendix E for the Matlab source code of the experiment.

Summary

The purpose of this chapter was to describe the details of the experiment that was conducted. This chapter covered test conditions, equipment, calibration, test participants, and procedures and how I applied them when I conducted the controlled experiment to produce reliable and replicable results.

IV. Analysis and Results

Chapter Overview

This chapter summarizes and discusses the results of the experiment on participant visual performance and alertness. The participants' responses were used to calculate the Michelson contrast threshold of visual detectability. Results are provided for both the illumination and orientation conditions. Participant alertness is also discussed by analyzing the response time of the participants and the Stanford Sleepiness Scale survey results. The results of the discomfort surveys are also discussed.

Contrast Threshold Overview

To begin analyzing the results of the experiment, the data points were converted to more useful information. The highest and lowest code values used in each Gabor patch presented was applied to equation 1, using the coefficients of the gamma correction, to calculate the luminance of the peak white and minimum black portions of the pattern. Since the light that enters the participants' eyes is not solely from the display, the reflected luminance, L_R, or the amount of ambient illumination from the LEDs that is reflected off the display, was calculated using equation 3 [74]. The reflected luminance is different for each vertical illuminance, which was measured in Table 7, and assumes a perfectly diffuse reflection from the surface of the display [74]. This is a valid assumption given the low spectral reflectance value of the display, as measured during the experiment calibration, and the large illumination sources.

$$L_R = \frac{R}{\pi} Y \tag{3}$$

where:
R = reflectance of the display
Y = vertical illuminance of each lighting condition measured at the display surface

With the peak white and minimum black luminance and the reflected luminance, I was able to calculate the Michelson contrast, according to equation 4 [14], [75]. This was an important step because contrast is one of the major parameters that affect visual performance and the contrast found using the Michelson contrast formula is the standard used in the field of vision science, and also referenced in the intelligence community's display standards [14], [60], [75]. The Michelson contrast can range from zero, which is neutral gray stripes on an equally neutral-gray background, up to one, which is black stripes on a pure-white background.

$$M = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} \tag{4}$$

where:
L_{max} = luminance of peak white code value plus L_R
L_{min} = luminance of minimum black code value plus L_R

In this experiment, the Michelson contrast gives a very accurate value of the display contrast for each pattern presented. However, because the participants were required to choose left or right each time, even if they could not identify the direction, they could guess correctly half the time. Therefore, the exact relationship between this relative contrast value and the participants' contrast threshold is not fully specified by the data. The QUEST function was able to provide approximate relative contrast information, but it only approached the participants' limit of detectability. To determine the precise contrast threshold, the data was evaluated using a survival analysis approach, also known as reliability analysis or failure rate analysis. This type of analysis involves modeling the point at which an event will happen, such as part failure in manufacturing. In this experiment it refers to the contrast point at which the pattern changes from being able to be detected to undetectable.

To counteract the inexact relationship between the relative contrast presentation conditions and the individuals' threshold, the data had to be assumed to be censored. Matlab only supports right censored data, so the calculation was performed using one minus the Michelson contrast with the correct responses corresponding to a one in the censoring data. This approach was taken as the QUEST algorithm began with very high values of relative contrast and worked toward the visual threshold, typically increasing contrast when an incorrect response was recorded. Therefore, the contrast for incorrect responses was typically closer to the threshold than many of the correct responses.

To find the value of the contrast at which point participants could no longer detect the Gabor pattern on the display, I aggregated the one-minus-Michelson contrast value from all participants and created data subsets for each lighting condition. A Weibull cumulative distribution function is often used in survival analysis to model the failure rate changes over time. Therefore, using Matlab's 'wblfit' function, a Weibull function was fit to each subset, and the one-minus-Michelson contrast was calculated that corresponded to a 75% probability of correctly identifying the pattern on the display. The 75% probability was chosen as there were only two possible responses, the threshold is located at the midpoint of the correct responses, or 50% of the upper 50% of responses. From here, the threshold was one minus the value of the Weibull function that corresponded to the 75% probability. This threshold is the target value on which the entire analysis is based. Similarly, using the α=0.05 confidence interval of each threshold, the upper and lower error bounds were also calculated.

The threshold denotes the Michelson contrast value at which participants were no longer able to detect the pattern on the display. Lower values correspond to a subtler difference between the brightest and the darkest portions of the pattern and represents cases near the lower limits of human perception. In other words, a lower threshold means there were more contrast levels able to be detected by the participants, translating to a higher visual performance for a wider variety of images. An example of the Michelson contrast value representing the threshold of detectability at which point participants were able to identify patterns with a 75% probability is shown in Figure 11. See Appendix F for the Matlab source code that calculates and plots the contrast thresholds.

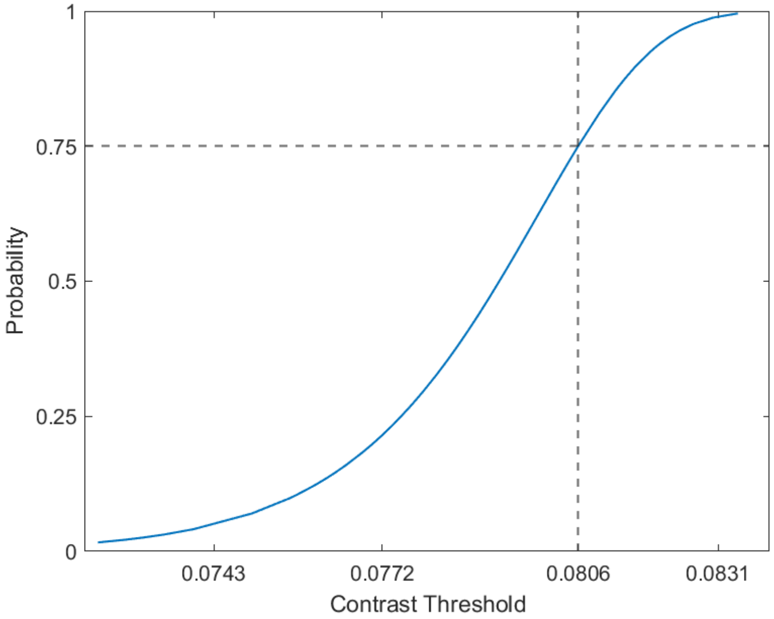

Figure 11. An Example Weibull Function Showing Contrast Threshold at 75% Correct Answers

Contrast Threshold Results

The contrast threshold directly pertains to the visual performance when viewing images on a display. Analyzing the threshold levels for each lighting condition was intended to help determine if performance positively or negatively changed in response to lighting exposure in order to answer the first and second research questions: (1) Does the ambient illumination level affect occupant visual performance? and (2) Does the orientation of light affect occupant visual performance?

To get a grasp of the Michelson contrast and its relationship to the luminance of the display, the vertical illuminance, and the reflected luminance values, the Michelson contrast value was graphed for each illuminance condition, orientation condition, and background scenario in Figure 12. The values used in this this figure are based on the minimum luminance differences for the most subtle pattern that the display was capable of presenting. These values are entirely based on the mathematical relationship between the variables and do not account for participant data. One thing to notice from this figure, and all subsequent figures, is that the point on the far left side is the 0 lux horizontal illumination condition where no LEDs are operating. We notice a trend for lower contrast thresholds with higher ambient illumination on the dark background scenarios. The same cannot be said for the light background scenarios as the differences are so small that they occur at the fifth decimal place.

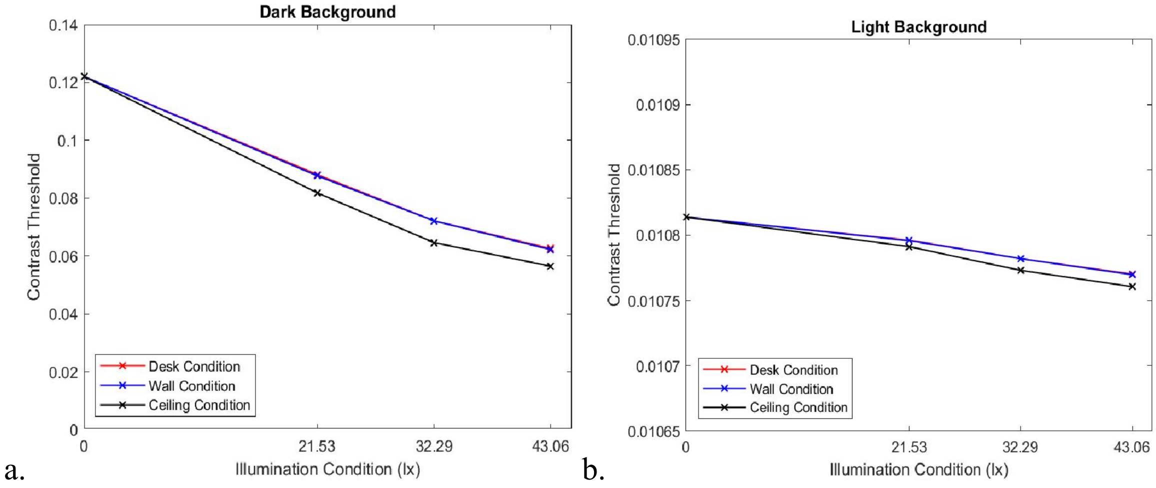

Figure 12. Ideal Contrast Thresholds with (a) Dark and (b) Light Backgrounds

From the participant data, thresholds were calculated for all patterns presented on the dark background (6.3 cd/m^2), comparing each condition of orientation and horizontal illumination. The plot in Figure 13 shows the Michelson contrast threshold of each lighting condition along with the corresponding shaded confidence interval. The figure demonstrates that participants performed their worst when they operated in the dark. It also shows that the lowest threshold occurred for the overhead lighting at the 32.29 lux horizontal illumination condition. This data point also falls outside the error bounds of the desk and wall conditions, indicating that it is unlikely that it happened by random chance.

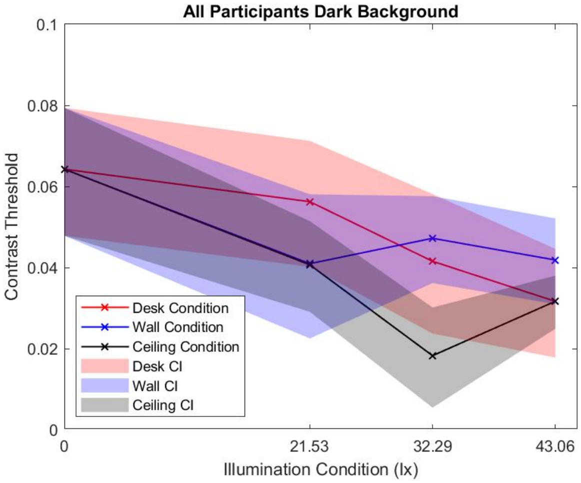

Figure 13. Contrast Thresholds with Shaded Confidence Intervals for All Participants with a Dark Background at Each Illuminance Condition

Using IBM's SPSS Statistics, the contrast threshold data was analyzed by performing a linear regression for illumination and orientation with the dark background scenarios. The results of that regression, portrayed in Table 9, show that when using the dark background, there is no statistically significant relationship between horizontal illumination and contrast threshold or between orientation and contrast threshold. Interestingly, the analysis did find a significant effect on contrast threshold from vertical illumination (p=0.032). The light measured at the display (vertical illuminance) has a mathematical association concerning the light measured at the keyboard (horizontal illuminance) and the orientation of the lighting. Finding a lack of correlation with horizontal illumination and orientation alone, while having a correlation of vertical illumination and contrast threshold indicates that contrast threshold is related more to the amount of light on the display than the total ambient light or direction it originates from. Thus as the light on the display increases, the contrast threshold decreases and participants perform better.

Table 9. Results of Linear Regression for Contrast Threshold on Dark Background Scenarios

Coefficients[a,b]					
	Unstandardized Coefficients		Standardized Coefficients		
Model	B	Std. Error	Beta	t	Sig.
Horizontal Illumination (lux)	0.021	0.011	1.698	1.885	0.108
Orientation Condition	-0.001	0.003	-0.096	-0.399	0.703
Vertical Illumination (lux)	-0.004	0.001	-2.396	-2.778	0.032
a. Dependent Variable: Contrast Threshold b. Selecting only cases for which Background Shade = Dark Gray					

The thresholds for all patterns presented on the light background (72 cd/m^2), evaluating each condition of orientation and horizontal illumination, are shown in Figure 14. It is important to recognize the considerable change in scale from the dark background in Figure 13 to the light background in Figure 14. This is expected due to the mathematical ratio relationship in the Michelson contrast formula in equation 4. While absolute differences in luminance from peak white to minimum black might be comparable between dark and light background patterns, their relative differences will not be similar. This is because a small luminance difference is such a small percentage of the total illuminance of the light background (72 cd/m^2) compared to a much larger percentage of the dark background (6.3 cd/m^2). It is also interesting to note that the threshold appears to trend downward with higher illumination levels, with the 0 lux condition performing the worst, and the 32.29 lux condition performing the best.

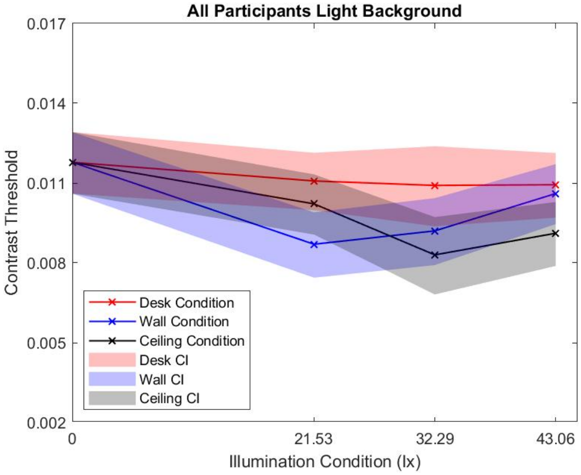

Figure 14. Contrast Thresholds with Shaded Confidence Intervals for All Participants with a Light Background at Each Illuminance Condition

The linear regression analysis of the contrast threshold with the light background scenario is in Table 10. The results are very similar to those found during the dark background scenario, with slightly less significance. The vertical illumination and contrast threshold was only significant at the 0.1 level (p=0.073). The reduced significance is likely due to the fact that more ambient light on the display is necessary to have an effect given the higher brightness of the display's background.

Table 10. Results of Linear Regression for Contrast Threshold on Light Background Scenarios

Coefficients[a,b]

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
Horizontal Illumination (lux)	0.003	0.001	2.354	1.827	0.118
Orientation Condition	0.0000	0.000	-0.039	-0.114	0.913
Vertical Illumination (lux)	0.000	0.000	-2.681	-2.173	0.073

a. Dependent Variable: Contrast Threshold

b. Selecting only cases for which Background Shade = Light Gray

Having examined all participants together, an additional analysis was performed to determine if the age of the participants had any effect on the contrast threshold. Analyzing the ages of the participants I noticed a gap where none of the participants were 29 years of age, and there happened to be an equal number of participants above and below 29 years of age. With the data subdivided into these two categories, I first analyzed the conditions with the dark background, and plotted the results in Figure 15. Under the 0 lux condition, the participants over 29 years of age had a slightly lower contrast threshold than the participants under 29 years of age, but it was within the confidence interval so the difference was not significant. In both age groups, the contrast threshold tended to trend downward with more ambient illumination, with the lowest point at the 32.29 lux condition with only overhead LEDs. In the under 29 years of age group, the 32.29 lux ceiling condition had a much lower threshold than the rest of the conditions and was outside the error bounds of the other orientation conditions, reducing the possibility it was due to random chance.

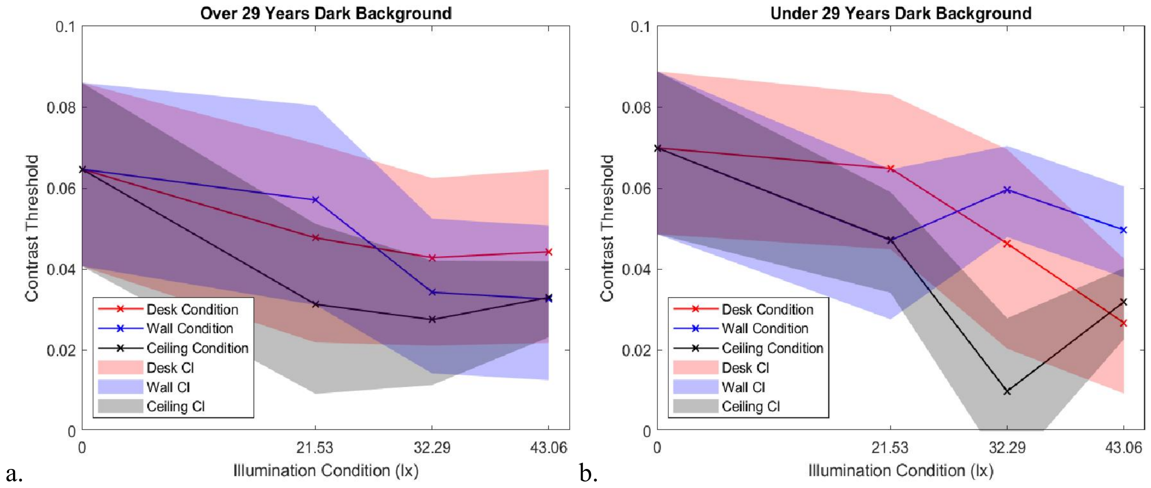

Figure 15. Comparison of Contrast Thresholds Related to Age with a Dark Background. (a) Over 29 and (b) Under Years of Age

The contrast threshold levels between the two age groups was then compared for the light background scenarios, as shown in Figure 16. In a divergence from the previous investigations, at 32.29 lux the wall condition performed better, but only for participants over 29 years of age. In each orientation condition, the under 29 years of age participants performed the best at 32.29 lux, with the ceiling condition leading the way. Regarding age, there was no statistical significance across background scenarios, orientation condition, or horizontal illumination condition.

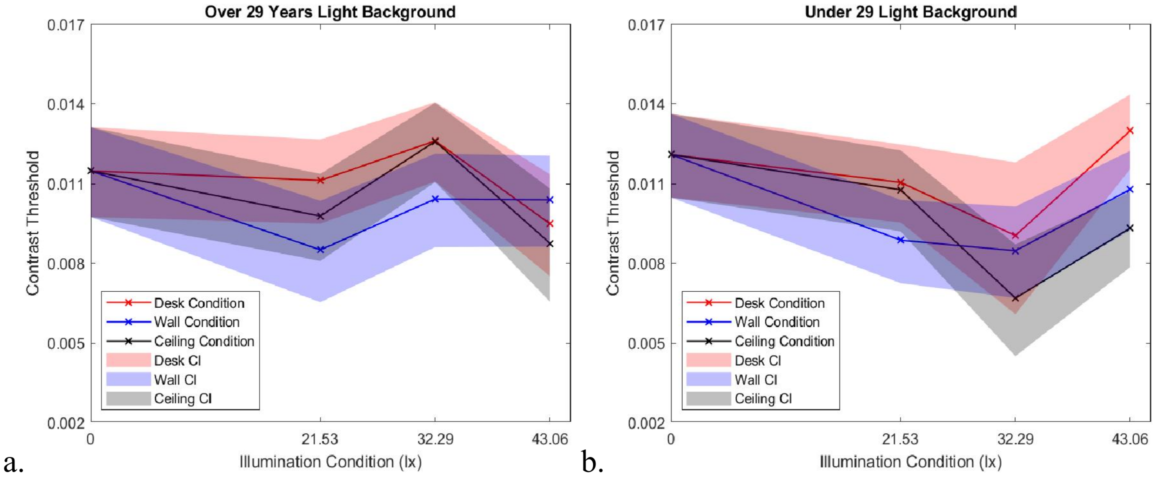

Figure 16. Comparison of Contrast Thresholds Related to Age with a Light Background. (a) Over 29 and (b) Under Years of Age

Response Time Results

In addition to the ability to see images displayed, as measured with contrast threshold, another large factor in visual performance is how quickly those image characteristics are identified. During the experiment, response times were recorded by measuring the time taken from when the pattern was first displayed to when the participant pressed a button on the keyboard. All participants' response times were averaged for each lighting condition and scenario. See Appendix G for the Matlab source code that calculates and plots the average response times. This analysis was intended to help determine if performance changed in response to lighting exposure in order to answer the first and second research questions: (1) Does the ambient illumination level affect occupant visual performance? and (2) Does the orientation of light affect occupant visual performance?

The plots of the results in Figure 17 show that across all participants, response times were slightly faster when presented with patterns on a light background as compared to the dark background scenarios. As far as the orientation of the light is concerned, on average the ceiling and desk conditions occupy all of the fastest response times. The wall condition had the slowest response times in most the of illumination conditions.

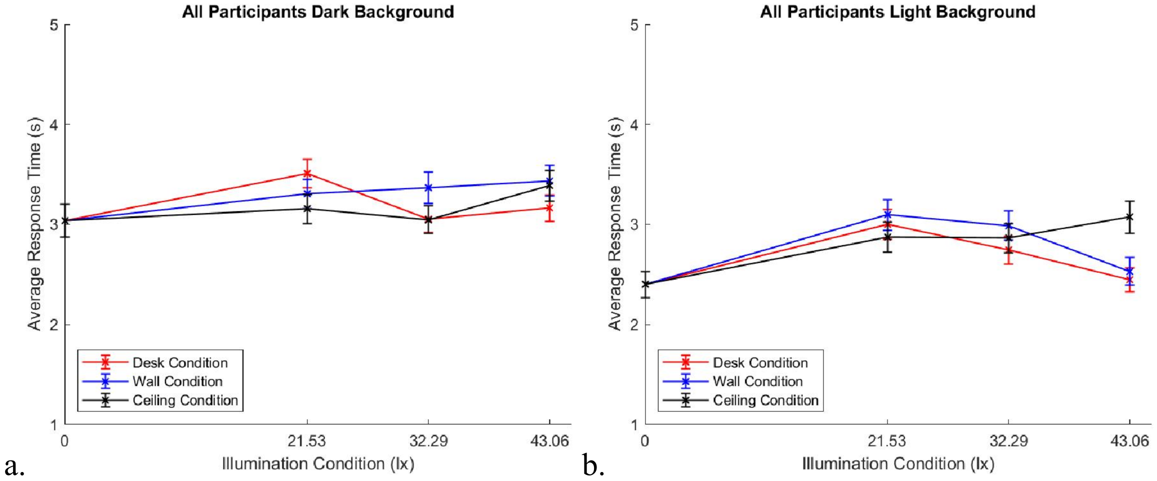

Figure 17. Comparison of Average Response Times for (a) Dark and (b) Light Backgrounds of All Participants

However, none of the orientations consistently perform differently than the others, especially when considering the uncertainty shown in the error bars. The large uncertainty is attributed to the number of participants in the experiment. Analyzing the total data of the response times showed that a couple of the participants are outliers compared with the rest of the group. The experiment includes both very decisive and very methodical participants. Examining Figure 18 also shows a large uncertainty within each participant’s response times. The mean response time of all participants is 3.02 with a standard deviation of 2.04. This experiment focused more on accuracy than on efficiency to allow for more precise calculation of the contrast threshold. Participants were not specifically told that their response times were recorded, which explains some of the large standard deviation within each participant. Understanding the patterns of the participants’ response times explains the sizeable error bars for the illumination conditions.

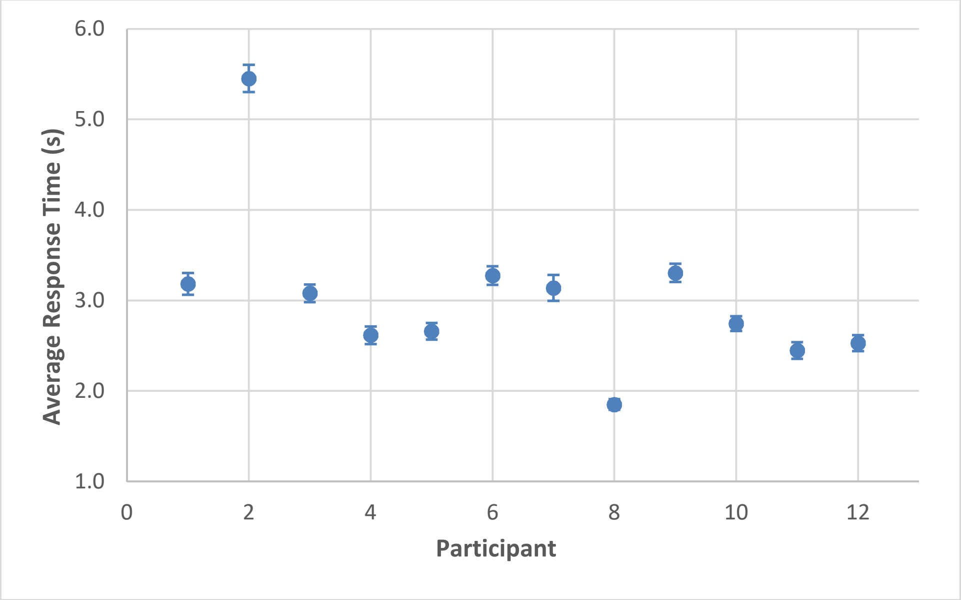

Figure 18. Average Response Time for Each Participant

The statistical analysis of response times versus the horizontal illumination and orientation conditions was performed in SPSS using a two-way repeated measures ANOVA test. This analysis was able to compare the same subjects under each of the different conditions. The results in Table 11 show that none of the conditions are statistically significant with respect to participant response times. I expect that encouraging participants to work at efficiently as possible would reduce the uncertainty in response times and may show more significant results regarding the various lighting conditions.

Table 11. Results of Two Way Repeated Measures ANOVA analysis of Response Times

Multivariate Tests[a]

Effect		Value	F	Hypothesis df	Sig.	Partial Eta Squared	Observed Power[c]
Horizontal Illumination	Wilks' Lambda	0.707	1.241[b]	3.0	0.351	0.293	0.231
Orientation	Wilks' Lambda	0.852	.871[b]	2.0	0.448	0.148	0.160
Horizontal Illumination * Orientation	Wilks' Lambda	0.643	.556[b]	6.0	0.753	0.357	0.120

a. Design: Intercept
 Within Subjects Design: Horizontal Illumination + Orientation + Horizontal Illumination * Orientation
b. Exact statistic
c. Computed using alpha = .05

I examined the data further by exploring the impact of age on response time using the same age groups discussed previously. As shown in Figure 19, participants under 29 years of age responded faster than the older participants when operating with the dark background. Generally speaking, this is not surprising and is to be expected since the reaction time in humans decreases as we get older.

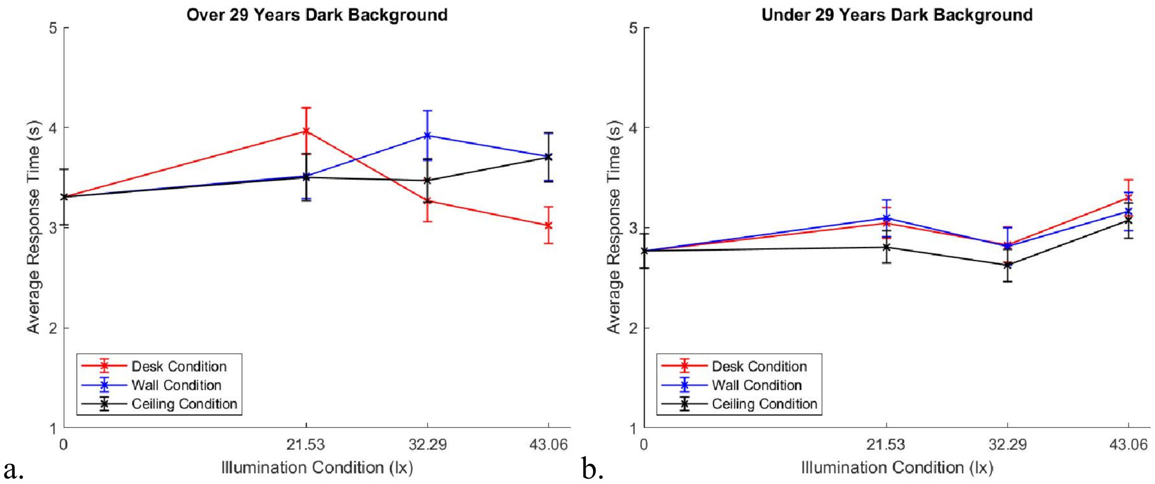

Figure 19. Comparison of Average Response Times Related to Age with a Dark Background. (a) Over 29 and (b) Under Years of Age

However, studying Figure 20 we see that it is not always the case that younger participants are faster. On light backgrounds at 0 lux, the participants over 29 years of age responded faster than their younger counterparts. Nevertheless, in almost every other condition the younger age group performed faster with smaller uncertainty than the older group.

Regardless of the background scenario, the participants over 29 years of age responded fastest with the desk condition at 43.06 lux. Whereas, the participants in the under 29 years of age group performed best when there were only overhead LEDs illuminated, specifically the 21.53 lux condition the fastest. Additionally, none of the data subsets showed any statistical significance between horizontal illumination and response time. Neither was there any relationship between orientation and response time. It is noteworthy that the best conditions for both participant groups were at least as good, if not better than the 0 lux condition.

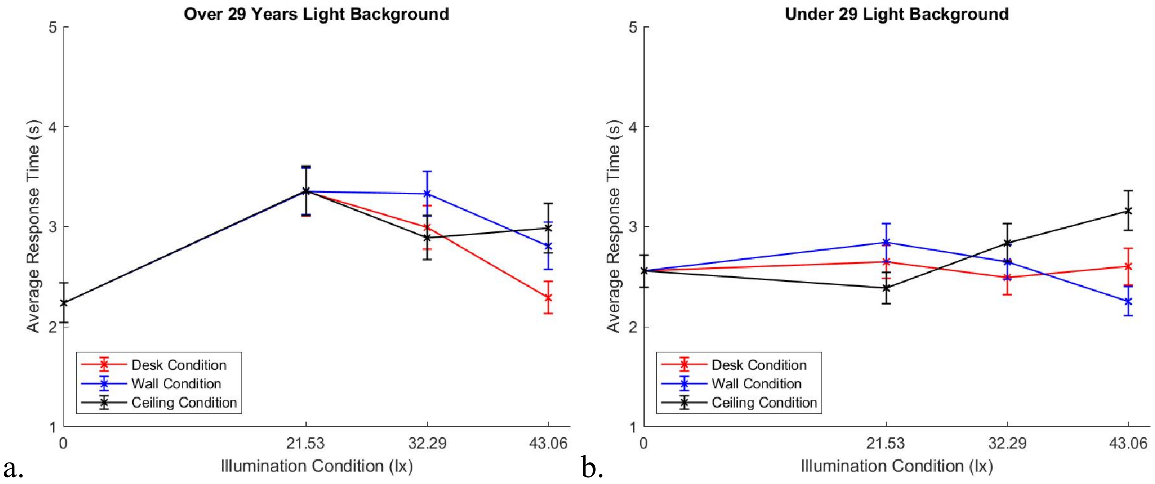

Figure 20. Comparison of Average Response Times Related to Age with a Light Background. (a) Over 29 and (b) Under Years of Age

However, some of the lowest response times did occur when the ambient illumination was 0 lux. One possible explanation for this comes from the condition sequencing information in Table 8. It shows that the order of the 21.53, 32.29, and 43.06 lux conditions varied, but every participant started their trial under the 0 lux illumination condition first. The large number of patterns presented to each participant, along with the tedious nature of the experiment had the potential to slow the response times of the participants during the experiment. Figure 21 shows the progression of the response times throughout the course of the experiment. In both the dark and the light backgrounds, the average response times do tend to slow down as the experiment continues.

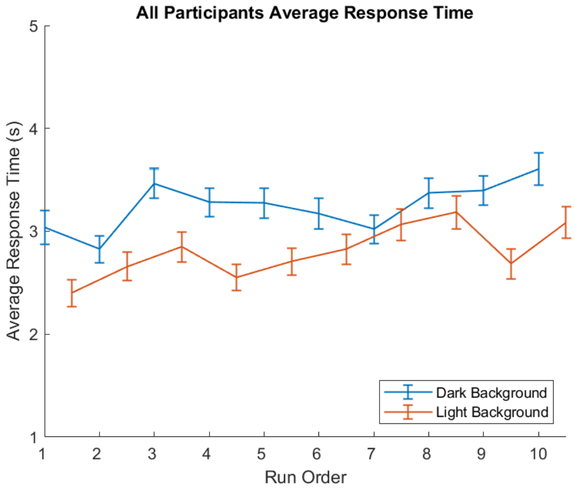

Figure 21. Average Response Time Trends Throughout the Experiment

Interestingly, the response times from the light background scenarios were faster than the response times from the dark background scenarios during every step in the experiment. The statistical test performed was the Kruskal-Wallis test, which is useful when the data violates some of the assumptions for using the ANOVA test, such as a small sample size and data that is not normally distributed. SPSS was able to statistically demonstrate the effects of the background scenario on the response times, H(3)=11.063, p=0.001, as shown in Table 12.

Table 12. Results of Kruskal-Wallis Test on Response Times Versus Background Shade

Test Statistics[a,b]

	Avg Response Time
Kruskal-Wallis H	11.063
df	1
Asymp. Sig.	0.001
Exact Sig.	0.000
Point Probability	0.000

a. Kruskal Wallis Test
b. Grouping Variable: Background Shade

Alertness Results

Not only does response time play a factor in performance, it also plays a factor in alertness as well. To further measure alertness, participants were given questionnaires periodically to gauge their subjective alertness using the Stanford Sleepiness Scale and also asked about their level of weariness or general feeling of fatigue. This was intended to help determine if alertness changed in response to lighting exposure to answer the third research question: Do moderate increases in ambient lighting effectively enhance analysts' alertness?

To be able to analyze the questionnaire data, the responses were transformed by assigning a numerical value representing each category. The Stanford Sleepiness Scale was already coded from 1-7, but the rest of the survey questions were each given a value from 1 (representing experiencing no symptoms), up to 5 (which represented very high experience of the listed symptoms). The average sleepiness and weariness data is shown in Figure 22 for each horizontal illumination condition. The lower values indicate a more wide-awake feeling and less weariness experienced. This shows that as the illumination increases, the participant's alertness decreases, as measured by sleepiness and weariness.

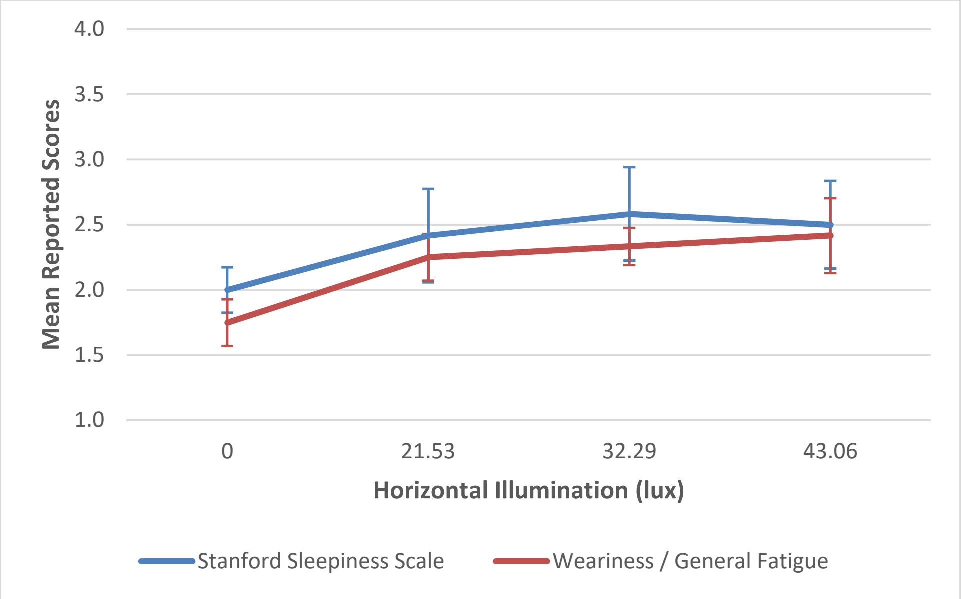

Figure 22. Average Reported Stanford Sleepiness Scale and Weariness/General Fatigue Versus Horizontal Illumination

A multivariate analysis was conducted to examine the statistical evidence of horizontal illumination and orientation on the sleepiness and weariness dependent variables. This allows simultaneous investigation into how the independent variables of horizontal illumination and orientation affect the dependent variables of sleepiness and weariness jointly and separately. Even though there appears to be a trend of decreasing alertness in the graph shown in the figure, it is not sufficient enough to produce statistical evidence of a direct correlation between horizontal illumination and either sleepiness or weariness. Table 13 shows that for horizontal illumination the two independent variables have p-values of 0.936 and 0.841 respectively.

Table 13. Results of Multivariate Analysis on Alertness Versus Horizontal Illumination and Orientation

Tests of Between-Subjects Effects								
Source	Dependent Variable	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Observed Power
Horizontal Illumination	Stanford Sleepiness Scale	.167	2	.083	.067	.936	.003	.059
	Weariness / Gen Fatigue	.167	2	.083	.174	.841	.009	.075
Orientation	Stanford Sleepiness Scale	4.667	2	2.333	1.867	.169	.089	.364
	Weariness / Gen Fatigue	3.167	2	1.583	3.297	.048	.148	.591
Horizontal Illumination * Orientation	Stanford Sleepiness Scale	.667	4	.167	.133	.969	.014	.075
	Weariness / Gen Fatigue	.667	4	.167	.347	.844	.035	.120

Additionally, the analysis in Table 13 shows no correlation between the combined effect of the horizontal illumination and orientation conditions on either measure of alertness. In short, there is no direct statistical relationship between horizontal illumination and alertness and there is no significant correlation to alertness when examining the interaction of horizontal illumination and orientation. Together, these results imply that alertness is not affected by the amount of horizontal illumination.

The most likely reason that there is insufficient statistical evidence is that the horizontal illumination is not the cause of the decreasing alertness. As was pointed out regarding Figure 21, the response times slowed down during the experiment regardless of the lighting conditions. Similarly, it is reasonable to expect that the alertness of the participants also decreased over the course of the experiment. Figure 23 reveals how the alertness decreased as the experiment progressed. While both Figure 21 and Figure 23 show decreasing alertness, the difference between the two figures is that Figure 23 preserves the order in which participants rated their subjective alertness. Thus we see that regardless of the horizontal illumination and orientation conditions, every measure of alertness used in the experiment–response time, sleepiness, and weariness–decreased as a result of the lengthy and tedious tasks that the participants were asked to perform.

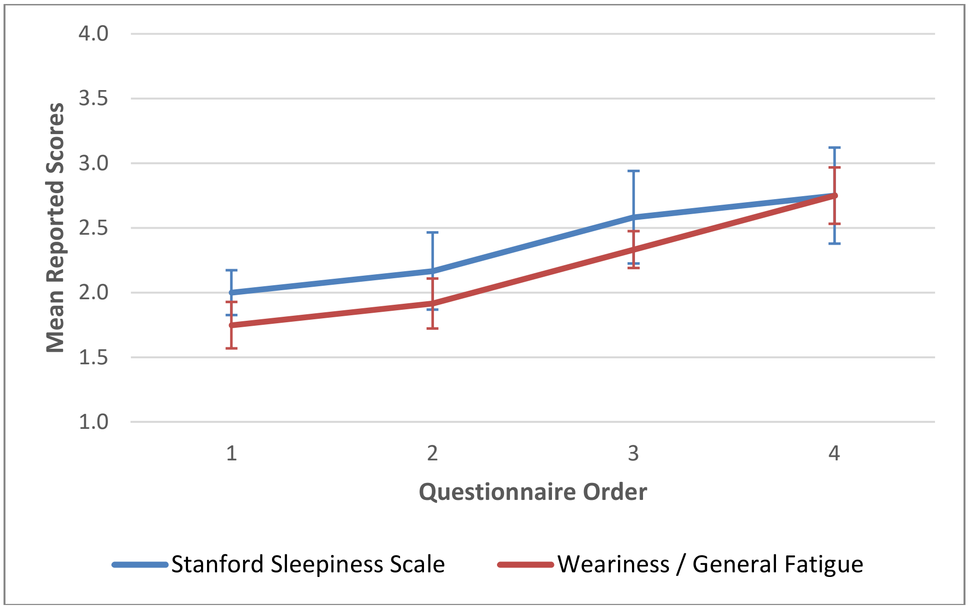

Figure 23. Average Reported Stanford Sleepiness Scale and Weariness/General Fatigue Versus Progression of Experiment

The premise that the monotony of the experiment decreased the participants' alertness is partially statistically confirmed with a multivariate analysis on the effect of questionnaire order to alertness. Table 14 shows that feelings of weariness and general fatigue have a p-value of 0.002, while sleepiness is not significant with relation to the order in which the questionnaires were accomplished. This is probably because the average sleepiness values just do not have enough variation to show significance.

Table 14. Results of Multivariate Analysis on Alertness Versus Progression of Experiment

Tests of Between-Subjects Effects								
Source	Dependent Variable	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Observed Power
Survey Order	Stanford Sleepiness Scale	4.417	3	1.472	1.274	.295	.080	.317
	Weariness / Gen Fatigue	7.229	3	2.410	5.863	.002	.286	.935

In addition to horizontal illumination, the third research question also investigates the relationship of orientation on alertness. Figure 24 shows the average reported Stanford Sleepiness Scale and feelings of weariness or general fatigue for each orientation condition.

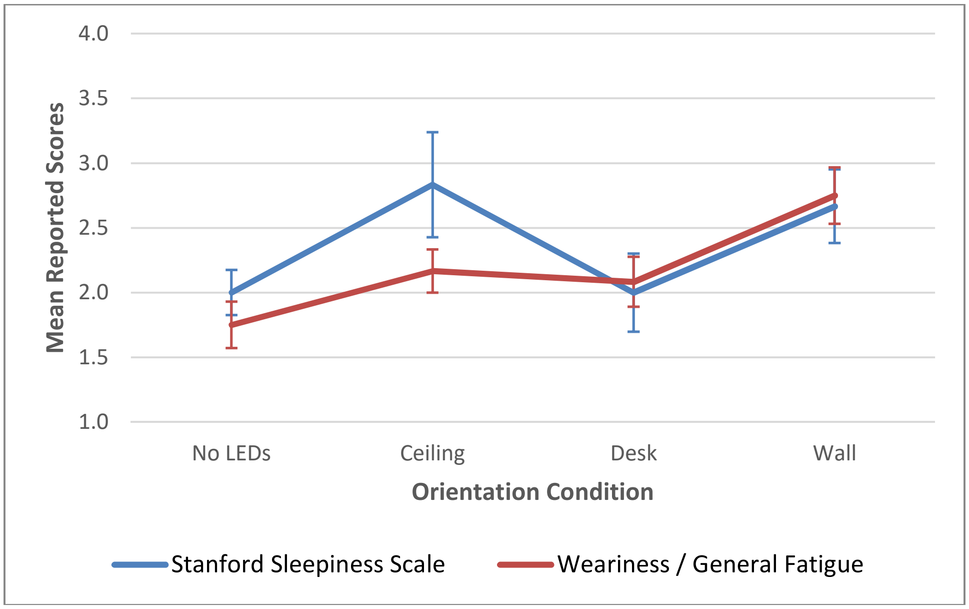

Figure 24. Average Reported Stanford Sleepiness Scale and Weariness/General Fatigue Versus Orientation

The results of examining the relationship between alertness and orientation are also included in Table 13. They are similar to the results found in the alertness and experiment progression investigation in that there was a statistical relationship between orientation and weariness (p=0.048), but not between orientation and sleepiness. But just as before, the alertness is decreasing through the experiment, so this may not show an accurate relationship between the variables.

Discomfort Results

Discomfort is a part of our overall well-being, and thus our health, which is one of the main underlying principles of the study. For example, if changing the lighting condition is linked to increases in visual detectability but causes severe headaches, it would not be a popular solution and could cause negative consequences such as increased error rates. Modern indoor lifestyles have enough potential to negatively influence human health, such as with vitamin D deficiency and circadian disruption, that it is important to examine some of the basic discomfort symptoms related to the lighting conditions. This is why discomfort was included in the third research questions investigated: Do moderate increases in ambient lighting effectively decrease analysts' symptoms of discomfort?

The survey questions related to discomfort include if participants are currently experiencing any headache, pain above the eyes, dryness in the eyes, irritation in the eyelids, sensitivity to light, difficulty seeing sharply, frequent blinking, eyes watering, or dull ache in the eyeballs. See Figure 10 for more detail on the discomfort survey questions.

Analyzing the survey data across all participants with respect to horizontal illumination, it shows that there was not much variation in the responses in each category of discomfort. The average of the reported discomfort levels is shown in Figure 25 for each horizontal illumination condition. It is interesting that the reported discomfort levels in nearly all categories decreased when changing the lighting condition from 0 lux to

21.53 lux, with a few categories declining faster than the rest, such as sensitivity to light and pain above the eyes.

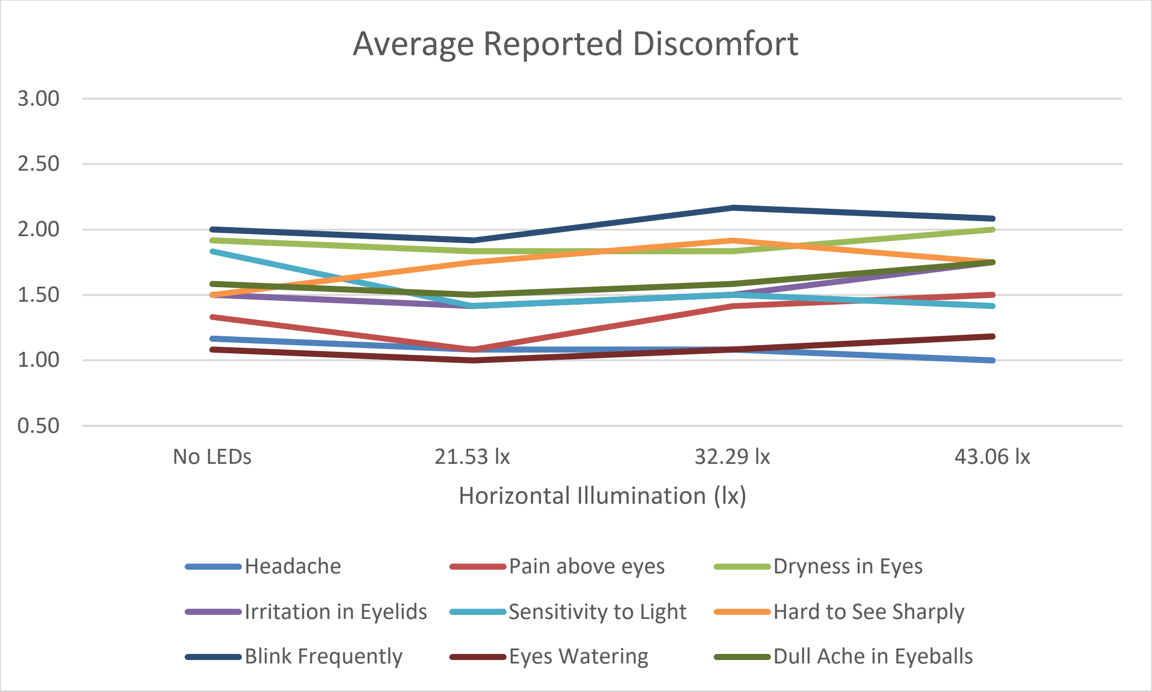

Figure 25. Average Reported Discomfort Versus Horizontal Illumination

The statistical analysis of the horizontal illumination conditions compared to each of the categories of discomfort revealed no significant relationships, as illustrated in Table 15. A probable explanation for the lack of statistical evidence is that the visual similarity in the repeated patterns caused participants eyes to naturally become tired throughout the experiment, which was manifested in many different discomfort symptoms. Since half the participants finished the experiment after the 21.53 lux condition and the other half finished after the 43.06 lux condition, having progressively more tired participants' eyes would confound the horizontal illumination results.

Table 15. Results of Kruskal-Wallis Test on Eye Discomfort Versus Horizontal Illumination

Test Statistics[a,b]									
	Headache	Pain above eyes	Dryness in Eyes	Irritation in Eyelids	Sensitive to Light	Hard to See Sharply	Frequent Blinking	Eyes Watering	Dull Ache in Eyeballs
Kruskal-Wallis H	1.047	4.183	0.139	0.714	1.311	1.943	0.434	2.387	1.257
df	3	3	3	3	3	3	3	3	3
Asymp. Sig.	0.790	0.242	0.987	0.870	0.727	0.584	0.933	0.496	0.739
Exact Sig.	1.000	0.258	0.986	0.883	0.739	0.596	0.924	0.441	0.753
Point Probability	0.400	0.027	0.001	0.008	0.004	0.007	0.001	0.133	0.015

a. Kruskal Wallis Test
b. Grouping Variable: Horizontal Illumination

To verify if the experiment itself could be a factor in discomfort, I compared the data from each discomfort category to the order in which the questionnaires were accomplished. A visual illustration of the average reported discomfort in each category is depicted in Figure 26. As with the horizontal illumination curves, this graph shows very little deviation in responses. However, it does show a decline in discomfort from the first questionnaire to the second, specifically in sensitivity to light, irritation in eyelids, pain above eyes, headache, and eyes watering. Those declines are small but meaningful because it shows that any amount of light at the second questionnaire (depending on the sequence of lighting conditions) is preferred by participants and reduces discomfort levels compared to the 0 lux condition at the first questionnaire.

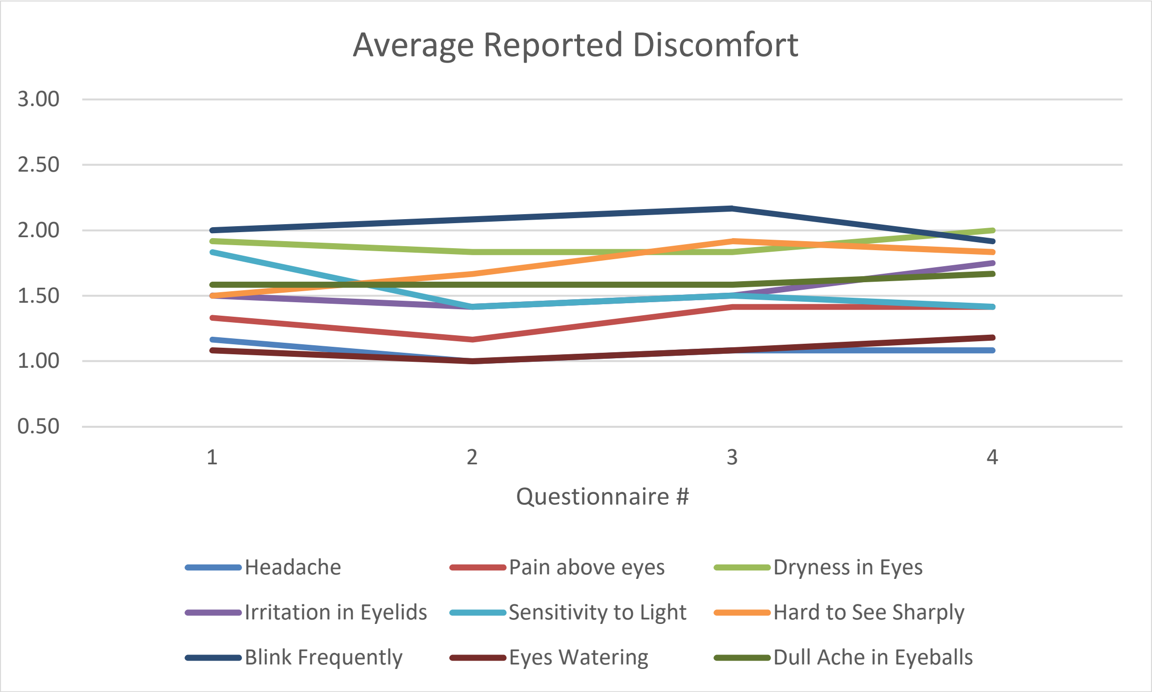

Figure 26. Average Reported Discomfort Versus Progression of Experiment

Using SPSS, the discomfort data related to the questionnaire order showed similar results to horizontal illumination results. None of the discomfort categories were found to be statistically correlated to the questionnaire order.

Table 16. Results of Kruskal-Wallis Test on Eye Discomfort Versus Progression of Experiment

Test Statistics[a,b]

	Headache	Pain above eyes	Dryness in Eyes	Irritation in Eyelids	Sensitive to Light	Hard to See Sharply	Frequent Blinking	Eyes Watering	Dull Ache in Eyeballs
Kruskal-Wallis H	1.047	1.820	0.139	0.714	1.311	2.125	0.434	2.387	0.199
df	3	3	3	3	3	3	3	3	3
Asymp. Sig.	0.790	0.610	0.987	0.870	0.727	0.547	0.933	0.496	0.978
Exact Sig.	1.000	0.668	0.986	0.883	0.739	0.560	0.924	0.441	0.985
Point Probability	0.400	0.068	0.001	0.008	0.004	0.008	0.001	0.133	0.010

a. Kruskal Wallis Test
b. Grouping Variable: Questionnaire #

Scrutinizing the visual representation of the orientation experiment condition, shown in Figure 27, there is a definite depression during the ceiling orientation for virtually every discomfort category. The graph also suggests participants reported experiencing the most symptoms of discomfort during the wall orientation condition.

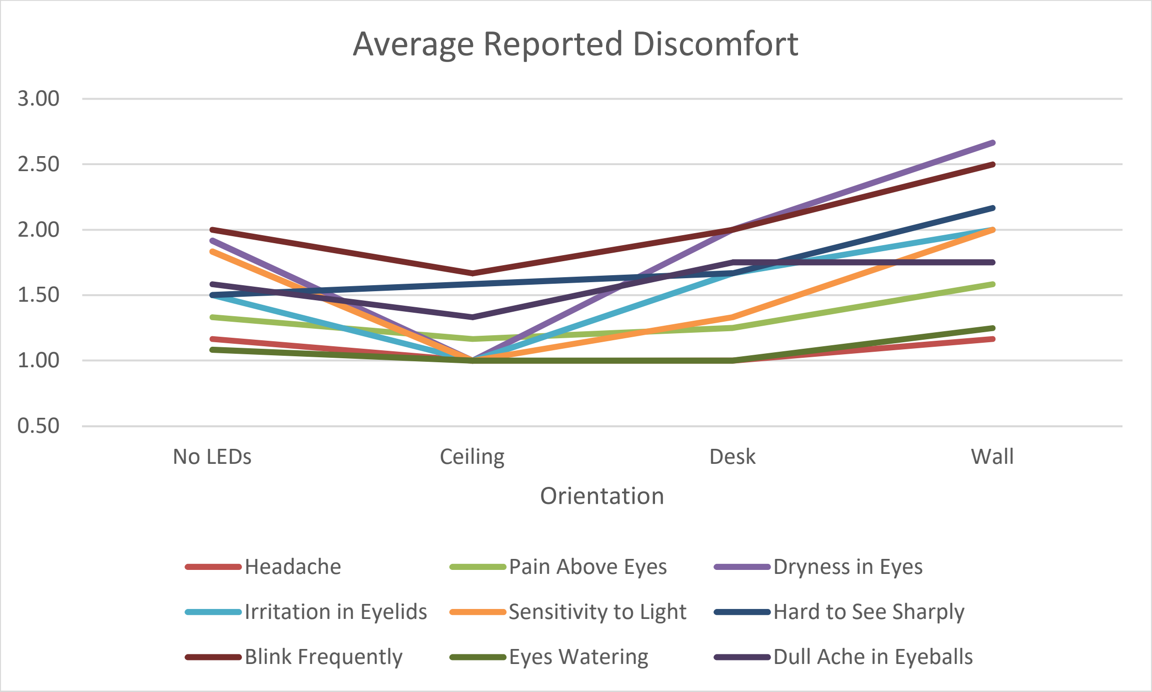

Figure 27. Average Reported Discomfort Versus Orientation

The information in Table 17 also confirms that there is a difference between orientation conditions regarding participants' discomfort. Statistically speaking, the orientation of the light in the experiment was significant at the 0.05 level when predicting dryness in the eyes, irritation in the eyelids, and sensitivity to light, with p-values of 0.000, 0.016, and 0.038 respectively. It is important to note that the reported values of discomfort are subjective. Meaning that these relationships might be at least partially based on personal preference. The overhead-only lighting condition is the most common in our society, and unfamiliar situations are not always well received at first.

Table 17. Results of Kruskal-Wallis Test on Eye Discomfort Versus Orientation

Test Statistics[a,b]									
	Headache	Pain above eyes	Dryness in Eyes	Irritation in Eyelids	Sensitive to Light	Hard to See Sharply	Frequent Blinking	Eyes Watering	Dull Ache in Eyeballs
Kruskal-Wallis H	3.715	3.686	21.831	10.283	8.444	5.188	5.214	6.196	4.086
df	3	3	3	3	3	3	3	3	3
Asymp. Sig.	0.294	0.297	0.000	0.016	0.038	0.159	0.157	0.102	0.252
Exact Sig.	0.600	0.353	0.000	0.013	0.032	0.160	0.159	0.173	0.249
Point Probability	0.183	0.055	0.000	0.000	0.000	0.001	0.000	0.089	0.000

a. Kruskal Wallis Test
b. Grouping Variable: Orientation

Evaluating the results of the horizontal illumination conditions and the orientation conditions, while weighing the outcome of the response time and discomfort analyses, indicates that the 32.29 lux ceiling condition provides the most favorable lighting condition.

Summary

The data discussed in this chapter provided insight into the effects of lighting conditions on participants' performance and alertness. I showed how the Michelson contrast threshold was calculated and analyzed to explore relationships to ambient illumination and orientation conditions. This chapter highlighted a few statistically significant findings, summarized in Figure 28, including that the ambient illumination can positively affect visual performance in some situations, and that the orientation of the light influences many of the items in the discomfort survey. The results from the data analysis in this chapter also show that there was a preference toward overhead lighting on the part of the participants.

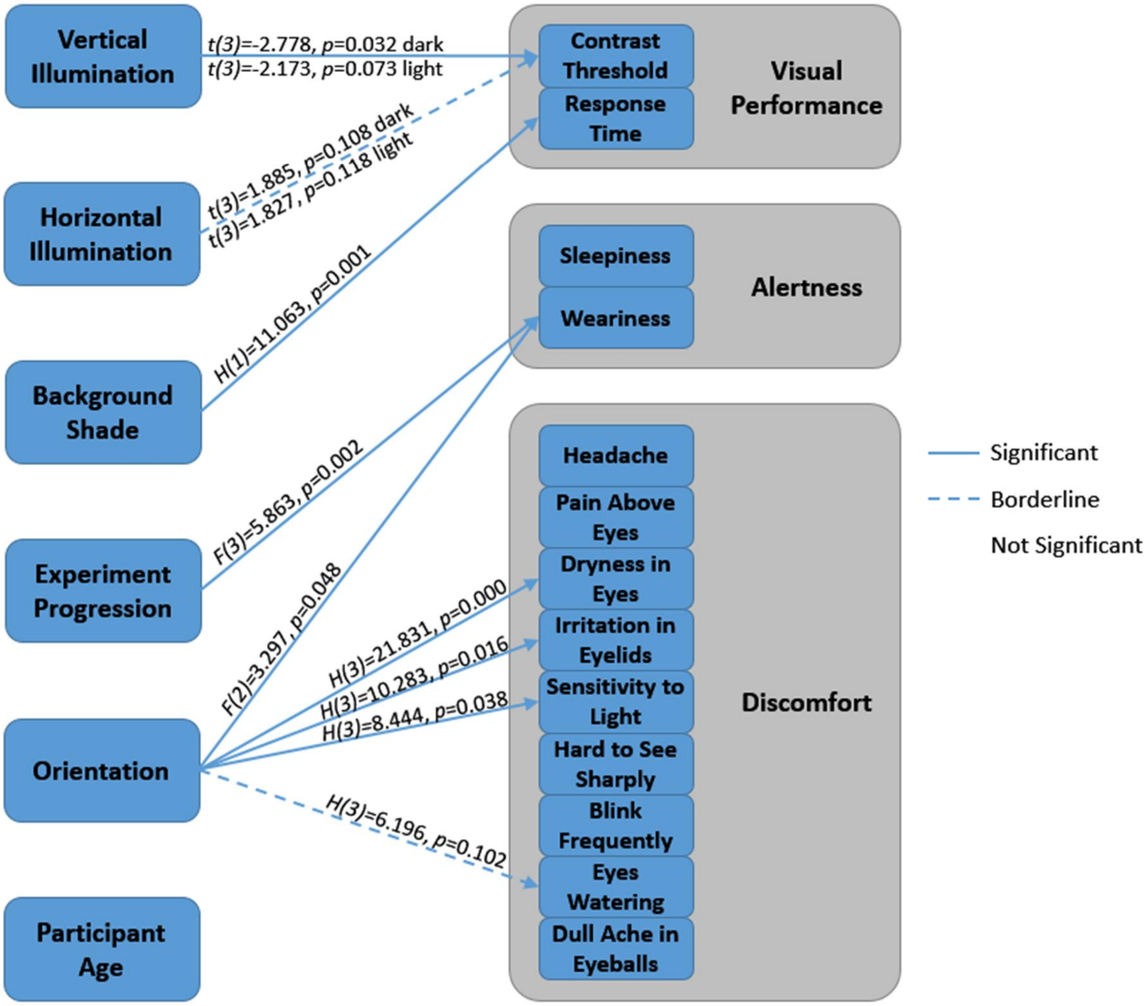

Figure 28. Summary of Significant Results

V. Conclusions and Recommendations

Chapter Overview

Many Air Force career fields require members to closely analyze details in images or information on electronic displays, often in dimly lit or dark work environments. Studies have shown that conducting activities in areas of insufficient light for extended periods has adverse influence on human health. Analysts in the intelligence community in particular are vulnerable to the associated depression and lack of alertness. Without researching solutions and addressing these possible adverse health effects, analysts are likely to suffer from symptoms associated with circadian disruption and alertness at work. Left unchecked, these matters can lead to three times higher rates of absenteeism from work and make employees more prone to errors by 30% to 50% [6], [7]. So without proper emphasis regarding the health of the operators, there is potential that the overall performance of the intelligence community may be limited due to problems with negative health issues.

The brightness of the image analysis environment is largely driven by the stringent NGA lighting standards for softcopy exploitation. The lighting standards were first established over a decade and a half ago and first targeted the highly reflective CRT displays. Originally the standards produced considerable visual performance improvement when using the older CRT technology. However, that display technology is no longer in use and today’s displays are substantially less reflective, which drastically reduces the glare compared to previous electronic display technology. Low-reflectance displays, such as LCDs, do not require lighting as rigorous as the NGA standards were originally created to be. This thesis challenged the long-standing illumination standards and sought a solution to the problem of reducing health concerns without sacrificing visual detection performance. To accomplish this, this research explored the effects of various light conditions and the potential effects on operator performance and alertness.

Based on previous research, specifically in the field of radiology, the hypothesis of this research was that moderate increases in ambient light (approximately 30-50 lux) actually increases detectability over predominantly dark environments. Additionally, it is hypothesized that the orientation of light plays an important role in performance and by shifting some of the light source away from directly shining on the display, while maintaining equivalent total ambient light, display visibility increases. Finally, this research hypothesizes that moderate increases in total ambient light in an environment increases analyst alertness and reduces discomfort associated with the eye's accommodation between the bright display and the dark surroundings.

To evaluate these hypotheses, this research conducted an experiment where participants were asked to identify the alignment of patterns on a display, with increasing subtlety. The experiment was conducted in a controlled lighting environment and participants performed the pattern recognition tasks under various illumination and orientation conditions. For each participant, relative contrast, accuracy, response time, sleepiness, and discomfort was recorded. From these data points a Michelson contrast value was calculated at the threshold of detectability and those contrast thresholds, along with the other data measures were analyzed to evaluate the effect of the lighting conditions on visual performance and alertness.

The rest of this chapter covers a summary of the findings from the analysis and how they apply to the hypotheses. Furthermore, this chapter highlights the significance of this research as well as discusses recommendations for additional actions and possible future research.

Conclusions of Research

To evaluate the effects of light on performance and alertness, the data was subjected to various statistical analyses including linear regression, repeated measures ANOVA, the Kruskal-Wallis H test, and multivariate analysis. The p-values stemming from those tests were used to identify any statistically significant relationships within the data.

This experiment found that visual performance, as measured by the contrast threshold, was statistically significant in response to higher illumination values measured at the display. The illumination measured at the keyboard fell just outside the significance level of 0.1. The reason horizontal illumination was not found to be significant may have been due to insufficient data points with only 12 participants. In this case, a large effect size at power = 0.80 for $\alpha = 0.05$, the minimum required sample size is 28, but would be much larger for smaller effect sizes [76]. The change in contrast threshold with regard to increased illumination is very small, especially on lighter background scenarios. The actual differences in contrast found in this experiment are close to the lower limits of the capabilities of the display. Nevertheless, the results are practically significant because of the nature of the images evaluated in softcopy exploitation. In radiology, the ability to detect the subtlest differences the display can project may lead to earlier discovery of cancerous cells. In the intelligence community, the nation’s security is at risk and increased visual performance may help to protect troops in combat. The statistical significance of increased illumination found in this experiment is consistent with the radiology study that demonstrated better participant performance under 25 and 40 lux illumination compared to when 7 or 100 lux illumination was used [58].

Despite the fact that the contrast threshold was lowest for the ceiling orientation condition in many instances, there was not enough statistical evidence to confirm any relationship between orientation and performance. Additionally, the age of the participants was accounted for in the analysis, but was ruled out as a contributor to performance on account of the lack of significance across orientation conditions or horizontal illumination conditions. The fact that this study failed to find significant performance effects based on age opposes previous research, which is likely due to the age of participants in this experiment. The oldest participant in this study was 41 years of age, whereas the research that did find significance in older populations incorporated participants over 60 years of age [57]. Further testing with a wider participant age range would be useful to confirm the results in this research.

The second measure of performance, response times, was determined to be an unreliable indicator of visual detectability. The monotonous structure of the identification tasks caused all participants' response times to slow proportionally in relation to the length of the experiment, with a p-value of 0.001. Therefore, any association between illumination or orientation conditions and performance was biased and could not be validated. Another likely explanation for the lack of illumination and response time correlation may have been the narrow set of lighting conditions tested. The study in the literature review that did find a correlation observed faster responses at 50 lux compared to 550 lux [46]. The lighting conditions used in this thesis were all less than 50 lux.

The bias in the response time data also impacted the ability to analyze the lighting conditions' effects on alertness. Two other measures of subjective alertness were recorded from the participants' answers to the Stanford Sleepiness Scale survey and the weariness/general fatigue question on the discomfort survey. The respondents' reported weariness was not significant with respect to horizontal illumination. Similar to response times, the weariness significantly increased throughout the experiment with a p-value of 0.005. Graphs of sleepiness showed comparable increases from the beginning to the end of the experiment, but was not statistically significant. This lack of evidence most likely originates from the fact that the responses ranged from 2 to 3 on a 7-point scale. The research did find that orientation was correlated to sleepiness at the 0.1 level (p=0.1) and correlated to weariness at the 0.05 level, with a p-value of 0.013. Again, the bias in the results prevented definitive conclusions regarding the effects of lighting on alertness.

The examination of questionnaire responses revealed minor variations in all categories of discomfort. As such, there was no statistical evidence linking horizontal illumination to the measures of discomfort used in this experiment. To understand if the experiment itself altered the participants' discomfort during the study, the data was analyzed according to the sequential order the questionnaires were completed, but no significance was found. The implication is that the lack of relationship between different illumination conditions and discomfort were accurate. Conversely, a few of the categories of discomfort were found to be correlated to the light orientation conditions. Specifically, dryness in the eyes, irritation in the eyelids, and sensitivity to light, were all measurably lower when participants were exposed to only overhead lighting, with p-values of 0.000, 0.016, and 0.038, respectively. This suggests that the intense brightness of the display causes discomfort when operating in 0 lux. It also suggests that the illumination of the wall behind the display may have caused some discomfort. The same output of the wall LEDs was used in both the wall and the desk orientation conditions; perhaps it was too bright.

Significance of Research

The research presented in this thesis showed insufficient evidence that ambient light or orientation has a large effect on performance and alertness. Additional research would be necessary to determine whether small or medium effects exist regarding ambient light qualities and visual detectability. Regardless, this research showed that participants in this experiment had a strong preference toward working with some ambient light versus operating in dark environments. Manifestations included the slight trend toward lower contrast thresholds and less reported symptoms of discomfort. As explained in literature, these slight improvements may be more psychological than physical. When employees are happy with their surroundings, task accomplishment increases and workplace complaints decline.

Based on this research, initial results suggest modifying Air Force and Department of Defense softcopy information exploitation facilities to use overhead lighting at 32.29 lux. As nearly all military applications require rapid information processing to support national security objectives, this suggestion has the advantage of similar efficiency to the current operations taking place in dark workplaces, according to the measured response times. Additionally, it has the benefit of often outperforming other conditions, if only slightly, and decreases symptoms of discomfort. In the commercial sector, in applications that do not carry the same hurried handling to prevent the potential life or death consequences, such as photo or video editing, the ceiling condition at 43.06 lux provides more balance between performance improvements and health benefits.

Recommendations for Action

The ideal contrast threshold curves in Figure 12, show better visual performance with increased illumination. The data collected in this experiment confirmed the same statistical relationship in a controlled environment (Table 9 and Table 10), consistent with radiological image analysis studies found in literature. Based on this information from the research, this study did not find any support for the limit on ambient illumination in softcopy exploitation facilities. Therefore, it is recommended that the NGA standards be updated to address the maximum illumination restrictions as well as to incorporate the effects of operator discomfort associated with low light conditions.

Another suggestion is to implement the 32.29 lux overhead lighting condition in a real world softcopy exploitation facility. Using overhead LED lighting with a correlated color temperature of 6500K would require the least modification to existing infrastructure. Implementing this in a non-mission critical, functional environment on a trial basis with realistic images instead of patterns, would provide practical experience on the effects that lighting has in actual operations. It is important to note that the overhead lighting used in this experiment was diffused across the ceiling to prevent point sources that may produce glare on the display. Replacing fluorescent bulbs with LEDs in the same traditional troffers found in a typical office environment would not generate the same results shown in this research. This suggestion requires diffuse light sources across a wide area, or using upward-directed pendant lights that could be diffused by the ceiling.

Recommendations for Future Research

Due to time constraints this research was limited to the number of lighting conditions able to be tested as well as the number of participants included in the experiment. Further research containing more illumination conditions and utilizing more participants may show more significant results. Additionally, similar to many studies conducted in the sphere of radiology, the use of trained analysts from the image analysis community as the participants would give a better understanding to how this research would generalize to operators in the field.

To more easily identify the contrast threshold, replacing the QUEST function with a standardized set of patterns with targeted levels of subtlety, presented in a random order, would allow a more uniform comparison between participants. With that standardized set of patterns, including the possibility of not displaying a pattern and requiring the participants to press a third button decreases the likelihood of lucky guesses and allows for error tracking.

An effort needs to be made to reduce the monotony of the tasks to keep participants mentally engaged in the experiment, possibly with more difficult tasks. Incorporating more mental variety or challenges, along with establishing a baseline sleepiness and weariness score will make the recorded response times and reported sleepiness and weariness data more meaningful. Doing so will provide a more accurate representation of alertness and performance in applications requiring efficient image processing.

The length of the experiment in this research was inadequate to study more in-depth measures, such as the lighting conditions' effects on participants' melatonin and vitamin D levels. These measures and the corresponding health effects were not studied in this research primarily since the participants' sleep-wake cycle and previous sunlight exposure outside of the experiment was not controlled. The study opted instead to analyze more short-term health effects that fit within the timeframe of the experiment, such as discomfort. While seemingly minor, the measures of discomfort are meaningful because the less studied type of feelings, such as unease or malaise, are linked to Sick Building Syndrome. A longer-term study is needed to obtain sufficient exposure durations for analysis of melatonin suppression at each illumination and orientation condition, thus an expanded understanding of alertness and its relationship to workplace lighting. Additionally, incorporating more broad spectrum LEDs with some UV wavelengths (i.e. down to 310 nm) will provide researchers with the opportunity to study the impact of environmental illumination on vitamin D levels in humans.

Summary

This research showed that lighting conditions, specifically the orientation of light, significantly affects participant sleepiness, weariness, and discomfort. It also showed that visual detection performance is improved at higher ambient illumination conditions with certain image types. The data from the experiment conducted did not find any significant changes in response time under varying lighting conditions. The research suggests that 32.29 lux diffuse overhead LED lighting may improve performance and alertness. Further research incorporating more lighting conditions and participants is needed to confirm the impact of lighting on performance and alertness.

Appendix A

Matlab Source Code for LED Output Curves

Luminance and Illuminance Curve Fit Calculations

```
%% Define Parameters
verifyon = true;
lumploton = true;
luxploton = true;
saveon = true;
myfile = 'LuminanceFitData.xlsx';
lumsheet = 'Luminance';
luxsheet = 'Illuminance';

%% Load Decay Values
% Luminance
% Code Values
lumcv = xlsread(myfile, lumsheet, 'A6:A14');
adaptlumcv = xlsread(myfile, lumsheet, 'A49:A59');
% Ceiling Channels
ceilingrlum = xlsread(myfile, lumsheet, 'B6:B14');
ceilingglum = xlsread(myfile, lumsheet, 'C6:C14');
ceilingblum = xlsread(myfile, lumsheet, 'D6:D14');
ceilingwlum = xlsread(myfile, lumsheet, 'E6:E14');
% Wall Channels
wallrlum = xlsread(myfile, lumsheet, 'B21:B29');
wallglum = xlsread(myfile, lumsheet, 'C21:C29');
wallblum = xlsread(myfile, lumsheet, 'D21:D29');
wallwlum = xlsread(myfile, lumsheet, 'E21:E29');
% Desk Channel
deskrlum = xlsread(myfile, lumsheet, 'B36:B44');
deskglum = xlsread(myfile, lumsheet, 'C36:C44');
deskblum = xlsread(myfile, lumsheet, 'D36:D44');
deskwlum = xlsread(myfile, lumsheet, 'E36:E44');
% Wall Channels, measuring at the wall not at the puck
wadaptwlum = xlsread(myfile, lumsheet, 'E49:E59');
% Desk Channel, measuring at the floor with the puck
dadaptwlum = xlsread(myfile, lumsheet, 'E64:E74');

% Illuminance
% Code Values
luxcv = xlsread(myfile, luxsheet, 'A10:A22');
% Ceiling Channels
ceilingrlux = xlsread(myfile, luxsheet, 'B10:B22');
ceilingglux = xlsread(myfile, luxsheet, 'C10:C22');
ceilingblux = xlsread(myfile, luxsheet, 'D10:D22');
ceilingwlux = xlsread(myfile, luxsheet, 'E10:E22');
ceilingwhlux = xlsread(myfile, luxsheet, 'F10:F22');
% Wall Channels
wallrlux = xlsread(myfile, luxsheet, 'B33:B45');
wallglux = xlsread(myfile, luxsheet, 'C33:C45');
wallblux = xlsread(myfile, luxsheet, 'D33:D45');
```

```
wallwlux = xlsread(myfile, luxsheet, 'E33:E45');
wallwhlux = xlsread(myfile, luxsheet, 'F33:F45');
% Desk Channel
deskrlux = xlsread(myfile, luxsheet, 'B56:B68');
deskglux = xlsread(myfile, luxsheet, 'C56:C68');
deskblux = xlsread(myfile, luxsheet, 'D56:D68');
deskwlux = xlsread(myfile, luxsheet, 'E56:E68');
deskwhlux = xlsread(myfile, luxsheet, 'F56:F68');

%% Plot Luminance & Illuminance vs Code Value for Verification
if verifyon
    figure;
    plot(lumcv, ceilingwlum);
    hold on;
    plot(lumcv, wallwlum);
    plot(lumcv, deskwlum);
    plot(adaptlumcv, wadaptwlum);
    plot(adaptlumcv, dadaptwlum);
    title('White Luminance(cd/m^2) vs Code Value');
    legend('Ceiling', 'Wall', 'Desk', 'Wall Adaptation', 'Desk Adaptation', …
       'Location', 'northwest');
    figure;
    plot(luxcv, ceilingwhlux);
    hold on;
    plot(luxcv, wallwhlux);
    plot(luxcv, deskwhlux);
    title('White Horizontal Illuminance(lux) vs Code Value');
    legend('Ceiling', 'Wall', 'Desk', 'Location', 'northwest');
end

%% Define Fittype
stexp2=fittype(@(a,b,c,cv) (a.*(cv.^b))-c, 'independent',{'cv'},'dependent',{'lum'});

%% Fit Curves
% Luminance
% Ceiling Channels
ceilingrlumfit = fit(lumcv, ceilingrlum, stexp2, 'StartPoint', [1 1 1]);
ceilingglumfit = fit(lumcv, ceilingglum, stexp2, 'StartPoint', [1 1 1]);
ceilingblumfit = fit(lumcv, ceilingblum, stexp2, 'StartPoint', [.1 1 1]);
ceilingwlumfit = fit(lumcv, ceilingwlum, stexp2, 'StartPoint', [1 1 1]);
% Wall Channels
wallrlumfit = fit(lumcv, wallrlum, stexp2, 'StartPoint', [.1 .5 1]);
wallglumfit = fit(lumcv, wallglum, stexp2, 'StartPoint', [.1 .5 1]);
wallblumfit = fit(lumcv, wallblum, stexp2, 'StartPoint', [.1 .5 1]);
wallwlumfit = fit(lumcv, wallwlum, stexp2, 'StartPoint', [.1 .5 1]);
% Desk Channel
deskrlumfit = fit(lumcv, deskrlum, stexp2, 'StartPoint', [.1 .25 1]);
deskglumfit = fit(lumcv, deskglum, stexp2, 'StartPoint', [.1 .2 1]);
deskblumfit = fit(lumcv, deskblum, stexp2, 'StartPoint', [.1 .25 1]);
deskwlumfit = fit(lumcv, deskwlum, stexp2, 'StartPoint', [.1 .25 1]);
% Wall Channels, measuring at the wall not at the puck
wadaptwlumfit = fit(adaptlumcv, wadaptwlum, stexp2, 'StartPoint', [1 1 1]);
% Desk Channel, measuring at the floor with the puck
dadaptwlumfit = fit(adaptlumcv, dadaptwlum, stexp2, 'StartPoint', [1 1 1]);
```

```
% Illuminance
% Ceiling Channels
ceilingwhluxfit = fit(luxcv, ceilingwhlux, stexp2, 'StartPoint', [1 3 1]);
% Wall Channels
wallwhluxfit = fit(luxcv, wallwhlux, stexp2, 'StartPoint', [1 1 1]);
% Desk Channel
deskwhluxfit = fit(luxcv, deskwhlux, stexp2, 'StartPoint', [1 .25 1]);

%% Save Coefficients
% Luminance
% Ceiling Channels
coeffceilingrlumfit=coeffvalues(ceilingrlumfit);
coeffceilingglumfit=coeffvalues(ceilingglumfit);
coeffceilingblumfit=coeffvalues(ceilingblumfit);
coeffceilingwlumfit=coeffvalues(ceilingwlumfit);
% Wall Channels
coeffwallrlumfit=coeffvalues(wallrlumfit);
coeffwallglumfit=coeffvalues(wallglumfit);
coeffwallblumfit=coeffvalues(wallblumfit);
coeffwallwlumfit=coeffvalues(wallwlumfit);
% Desk Channel
coeffdeskrlumfit=coeffvalues(deskrlumfit);
coeffdeskglumfit=coeffvalues(deskglumfit);
coeffdeskblumfit=coeffvalues(deskblumfit);
coeffdeskwlumfit=coeffvalues(deskwlumfit);
% Wall Channels, measuring at the wall not at the puck
coeffwadaptwlumfit=coeffvalues(wadaptwlumfit);
% Desk Channel, measuring at the floor with the puck
coeffdadaptwlumfit=coeffvalues(dadaptwlumfit);

% Illuminance
% Ceiling Channels
coeffceilingwhluxfit=coeffvalues(ceilingwhluxfit);
% Wall Channels
coeffwallwhluxfit=coeffvalues(wallwhluxfit);
% Desk Channel
coeffdeskwhluxfit=coeffvalues(deskwhluxfit);

%% Plot Fitted Curves
if lumploton
    % Luminance
    % Ceiling Channels
    figure;
    plot(ceilingrlumfit, 'r', lumcv, ceilingrlum, 'rx');
    hold on;
    plot(ceilingglumfit, 'g', lumcv, ceilingglum, 'g+');
    plot(ceilingblumfit, 'b', lumcv, ceilingblum, 'b*');
    plot(ceilingwlumfit, 'k', lumcv, ceilingwlum, 'ks');
    title('Ceiling Channels Luminance(cd/m2) vs Code Value');
    legend('Red', 'fitted curve', 'Green', 'fitted curve', 'Blue', 'fitted curve', …
        'White', 'fitted curve', 'Location', 'northwest');
    % Wall Channels
    figure;
    plot(wallrlumfit, 'r', lumcv, wallrlum, 'rx');
    hold on;
    plot(wallblumfit, 'g', lumcv, wallblum, 'g+');
    plot(wallglumfit, 'b', lumcv, wallglum, 'b*');
    plot(wallwlumfit, 'k', lumcv, wallwlum, 'ks');
    title('Wall Channels Luminance(cd/m2) vs Code Value');
    legend('Red', 'fitted curve', 'Green', 'fitted curve', 'Blue', 'fitted curve', …
       'White', 'fitted curve', 'Location', 'northwest');
    % Desk Channel
    figure;
    plot(deskrlumfit, 'r', lumcv, deskrlum, 'rx');
    hold on;
    plot(deskglumfit, 'g', lumcv, deskglum, 'g+');
    plot(deskblumfit, 'b', lumcv, deskblum, 'b*');
    plot(deskwlumfit, 'k', lumcv, deskwlum, 'ks');
    title('Desk Channel Luminance(cd/m2) vs Code Value');
    legend('Red', 'fitted curve', 'Green', 'fitted curve', 'Blue', 'fitted curve', …
       'White', 'fitted curve', 'Location', 'northwest');
    % Wall Channels, measuring at the wall not at the puck
    figure;
    plot(wadaptwlumfit, 'k', adaptlumcv, wadaptwlum, 'ks');
    title('Wall Adaptation Luminance(cd/m2) vs Code Value');
    legend('White', 'fitted curve', 'Location', 'northwest');
    % Desk Channel, measuring at the floor with the puck
    figure;
    plot(dadaptwlumfit, 'k', adaptlumcv, dadaptwlum, 'ks');
    title('Desk Adaptation Luminance(cd/m2) vs Code Value');
    legend('White', 'fitted curve', 'Location', 'northwest');
end
if luxploton
    % Illuminance
    % Ceiling Channels
    figure;
    plot(ceilingwhluxfit, 'k', luxcv, ceilingwhlux, 'ks');
    title('Ceiling Channels Horizontal Illuminance(lux) vs Code Value');
    legend('White', 'fitted curve', 'Location', 'northwest');
    % Wall Channels
    figure;
    plot(wallwhluxfit, 'k', luxcv, wallwhlux, 'ks');
    title('Wall Channels Horizontal Illuminance(lux) vs Code Value');
    legend('White', 'fitted curve', 'Location', 'northwest');
    % Desk Channel
    figure;
    plot(deskwhluxfit, 'k', luxcv, deskwhlux, 'ks');
    title('Desk Channel Horizontal Illuminance(lux) vs Code Value');
    legend('White', 'fitted curve', 'Location', 'northwest');
end

%% Provide user option to save coefficients in a mat file
if saveon
    % Luminance
    % Ceiling Channels
    coeffs.coeffceilingrlumfit = coeffceilingrlumfit;
    coeffs.coeffceilingglumfit = coeffceilingglumfit;
    coeffs.coeffceilingblumfit = coeffceilingblumfit;
    coeffs.coeffceilingwlumfit = coeffceilingwlumfit;
    % Wall Channels
    coeffs.coeffwallrlumfit = coeffwallrlumfit;
    coeffs.coeffwallglumfit = coeffwallglumfit;
    coeffs.coeffwallblumfit = coeffwallblumfit;
    coeffs.coeffwallwlumfit = coeffwallwlumfit;
    % Desk Channel
    coeffs.coeffdeskrlumfit = coeffdeskrlumfit;
    coeffs.coeffdeskglumfit = coeffdeskglumfit;
    coeffs.coeffdeskblumfit = coeffdeskblumfit;
    coeffs.coeffdeskwlumfit = coeffdeskwlumfit;
    % Wall Channels, measuring at the wall not at the puck
    coeffs.coeffwadaptwlumfit = coeffwadaptwlumfit;
    % Desk Channel, measuring at the floor with the puck
    coeffs.coeffdadaptwlumfit = coeffdadaptwlumfit;

    % Illuminance
    % Ceiling Channels
    coeffs.coeffceilingwhluxfit = coeffceilingwhluxfit;
    % Wall Channels
    coeffs.coeffwallwhluxfit = coeffwallwhluxfit;
    % Desk Channel
    coeffs.coeffdeskwhluxfit = coeffdeskwhluxfit;

    [fn, pn]=uiputfile('*.mat', 'Save File Containing Coefficients');
    if(fn ~= 0)
        save([pn fn], '-struct', 'coeffs');
    end
end
```

Gamma Correction Curve Fit Calculations

```
%% Define Parameters
verifyon = true;
ploton = true;
saveon = false;
myfile = 'DisplayCalibration.xlsx';
sheet = 'Sheet1';

%% Load Decay Values
% Code Values
rgbcv = xlsread(myfile, sheet, 'A7:A19');
% Luminance Measurement
lum = xlsread(myfile, sheet, 'C7:C19');

%% Plot Luminance vs Code Value for Verification
if verifyon
    figure;
    plot(rgbcv, lum);
    hold on;
    title('Luminance(cd/m^2) vs Code Value');
    legend('Luminance', 'Location', 'northwest');
end
```

```
%% Define Fittype
myfunc = fittype( @(a, b, c, cv) (a.*(((cv)/(256)).^b))-c, 'independent', {'cv'}, …
       'dependent', {'lum'});

%% Fit Curves
% Luminance
lumfit = fit(rgbcv, lum, myfunc, 'StartPoint', [256 1 1]);

%% Save Coefficients
% Luminance
lumcoeff=coeffvalues(lumfit);

%% Plot Fitted Curves
if ploton
    % Luminance
    figure;
    plot(lumfit, 'r', rgbcv, lum, 'rx');
    hold on;
    title('Fitted Luminance(cd/m2) curve vs Code Value, with measured data points');
    legend('Measured Data', 'Fitted Curve', 'Location', 'northwest');
end

%% Provide user option to save coefficients in a mat file
if saveon
    % Luminance
    coeffs.lumcoeff = lumcoeff;

    [fn, pn]=uiputfile('*.mat', 'Save File Containing Coefficients');
    if(fn ~= 0)
        save([pn fn], '-struct', 'coeffs');
    end
end
```

Appendix B

Matlab Source Code for Calibrating Each Illumination Condition

Hunt's Center of Gravity Law of Color Mixture

```
function [rY] = bal_prim(rgb_xy,w_xy, w_Y)
%bal_prim - Calculates the relative luminance values for a 3 primary combination to
%achieve a white point.
% This uses Hunt's Center of Gravity Law, solved from:
% syms m1 m2 m3 x1 x2 x3 y1 y2 y3 x y
% S=solve(m1+m2+m3 == 1, x==(m1*x1/y1 + m2*x2/y2 + m3*x3/y3)/(m1/y1+m2/y2+m3/y3), ...
%   y==(m1*y1/y1 + m2*y2/y2 + m3*y3/y3)/(m1/y1+m2/y2+m3/y3), m1, m2, m3);
%
% INPUTS:
% rgb_xy - xy coordinates for the three primaries
% w_xy - xy coordinates for the white
% w_Y - luminance of the whitepoint, assumed to be 1
%
% OUTPUTS;
% rY - relative luminance values
%
% Created April 29, 2018, Michael Miller
%
if nargin < 3
    w_Y = 1;
end
if nargin < 2
    error('Not enough variables entered');
end
x1 = rgb_xy(1,1);
y1 = rgb_xy(1,2);
x2 = rgb_xy(2,1);
y2 = rgb_xy(2,2);
x3 = rgb_xy(3,1);
y3 = rgb_xy(3,2);

m1 = (w_xy(1)*y1*y2 - x2*w_xy(2)*y1 - w_xy(1)*y1*y3 + x3*w_xy(2)*y1 + x2*y1*y3 - …
       x3*y1*y2)/(x1*w_xy(2)*y2 - x2*w_xy(2)*y1 - x1*w_xy(2)*y3 + x3*w_xy(2)*y1 + …
       x2*w_xy(2)*y3 - x3*w_xy(2)*y2);
m2 = -(w_xy(1)*y1*y2 - x1*w_xy(2)*y2 - w_xy(1)*y2*y3 + x3*w_xy(2)*y2 + x1*y2*y3 - …
       x3*y1*y2)/(x1*w_xy(2)*y2 - x2*w_xy(2)*y1 - x1*w_xy(2)*y3 + x3*w_xy(2)*y1 + …
       x2*w_xy(2)*y3 - x3*w_xy(2)*y2);
m3 = (w_xy(1)*y1*y3 - x1*w_xy(2)*y3 - w_xy(1)*y2*y3 + x2*w_xy(2)*y3 + x1*y2*y3 - …
       x2*y1*y3)/(x1*w_xy(2)*y2 - x2*w_xy(2)*y1 - x1*w_xy(2)*y3 + x3*w_xy(2)*y1 + …
       x2*w_xy(2)*y3 - x3*w_xy(2)*y2);
rY = [m1*w_Y; m2*w_Y; m3*w_Y];
end
```

Ceiling Condition Calculations

```
%% Define Parameters
rgbxy = [.6915 .3054; .1363 .6787; .1464 .0489];
```

```
rgwxy = [.6915 .3054; .1363 .6787; .2491 .2070];
d65xy = [.3127 .3290];

%% Find Color Ratios
rgbY = bal_prim(rgbxy,d65xy,100);
rgwY = bal_prim(rgwxy,d65xy,100);
rgbwY = [.5*rgbY(1)+.5*rgwY(1); .5*rgbY(2)+.5*rgwY(2); .5*rgbY(3); .5*rgwY(3)];

%% Calculate Exponential Curves
load('FitCurves');

%% Find Code Values for Standard Illuminance
disp '100% Standard Illuminance';
syms x;
stdlux = 21.5278;
stdcv=double(solve(stdlux==coeffceilingwhluxfit(1)*x^coeffceilingwhluxfit(2)-…
       coeffceilingwhluxfit(3), 'Real', true));
stdlum=coeffceilingwlumfit(1)*stdcv^coeffceilingwlumfit(2)-coeffceilingwlumfit(3);
stdlumrY=rgbwY(1)*stdlum/100;
stdlumgY=rgbwY(2)*stdlum/100;
stdlumbY=rgbwY(3)*stdlum/100;
stdlumwY=rgbwY(4)*stdlum/100;
rstdcv=double(solve(stdlumrY==coeffceilingrlumfit(1)*x^coeffceilingrlumfit(2)-…
       coeffceilingrlumfit(3), 'Real', true))
gstdcv=double(solve(stdlumgY==coeffceilingglumfit(1)*x^coeffceilingglumfit(2)-…
       coeffceilingglumfit(3), 'Real', true))
bstdcv=double(solve(stdlumbY==coeffceilingblumfit(1)*x^coeffceilingblumfit(2)-…
       coeffceilingblumfit(3), 'Real', true))
wstdcv=double(solve(stdlumwY==coeffceilingwlumfit(1)*x^coeffceilingwlumfit(2)-…
       coeffceilingwlumfit(3), 'Real', true))

%% Find Code Values for Standard Illuminance + 50%
fprintf('\n');
disp '150% Standard Illuminance';
syms y;
fiftystdlux = 1.5*stdlux;
fiftystdcv=double(solve(fiftystdlux==coeffceilingwhluxfit(1)*…
       y^coeffceilingwhluxfit(2)-coeffceilingwhluxfit(3), 'Real', true));
fiftystdlum=coeffceilingwlumfit(1)*fiftystdcv^coeffceilingwlumfit(2)-…
       coeffceilingwlumfit(3);
fiftystdlumrY=rgbwY(1)*fiftystdlum/100;
fiftystdlumgY=rgbwY(2)*fiftystdlum/100;
fiftystdlumbY=rgbwY(3)*fiftystdlum/100;
fiftystdlumwY=rgbwY(4)*fiftystdlum/100;
rfiftystdcv=double(solve(fiftystdlumrY==coeffceilingrlumfit(1)*…
       y^coeffceilingrlumfit(2)-coeffceilingrlumfit(3), 'Real', true))
gfiftystdcv=double(solve(fiftystdlumgY==coeffceilingglumfit(1)*y^…
       coeffceilingglumfit(2)-coeffceilingglumfit(3), 'Real', true))
bfiftystdcv=double(solve(fiftystdlumbY==coeffceilingblumfit(1)*y^…
       coeffceilingblumfit(2)-coeffceilingblumfit(3), 'Real', true))
wfiftystdcv=double(solve(fiftystdlumwY==coeffceilingwlumfit(1)*y^…
       coeffceilingwlumfit(2)-coeffceilingwlumfit(3), 'Real', true))

%% Find Code Values for double the Standard Illuminance
```

```
fprintf('\n');
disp '200% Standard Illuminance';
syms z;
doubstdlux = 2*stdlux;
doubstdcv=double(solve(doubstdlux==coeffceilingwhluxfit(1)*z^coeffceilingwhluxfit(2)-...
       coeffceilingwhluxfit(3), 'Real', true));
doubstdlum=coeffceilingwlumfit(1)*doubstdcv^coeffceilingwlumfit(2)-...
       coeffceilingwlumfit(3);
doubstdlumrY=rgbwY(1)*doubstdlum/100;
doubstdlumgY=rgbwY(2)*doubstdlum/100;
doubstdlumbY=rgbwY(3)*doubstdlum/100;
doubstdlumwY=rgbwY(4)*doubstdlum/100;
rdoubstdcv=double(solve(doubstdlumrY==coeffceilingrlumfit(1)*z^coeffceilingrlumfit(2)...
       -coeffceilingrlumfit(3), 'Real', true))
gdoubstdcv=double(solve(doubstdlumgY==coeffceilingglumfit(1)*z^coeffceilingglumfit(2)...
       -coeffceilingglumfit(3), 'Real', true))
bdoubstdcv=double(solve(doubstdlumbY==coeffceilingblumfit(1)*z^coeffceilingblumfit(2)...
       -coeffceilingblumfit(3), 'Real', true))
wdoubstdcv=double(solve(doubstdlumwY==coeffceilingwlumfit(1)*z^coeffceilingwlumfit(2)...
       -coeffceilingwlumfit(3), 'Real', true))
```

Wall Condition Calculations

```
%% Define Parameters
rgbxy = [.6915 .3054; .1363 .6787; .1464 .0489];
rgwxy = [.6915 .3054; .1363 .6787; .2491 .2070];
d65xy = [.3127 .3290];

%% Find Color Ratios
rgbY = bal_prim(rgbxy,d65xy,100);
rgwY = bal_prim(rgwxy,d65xy,100);
rgbwY = [.5*rgbY(1)+.5*rgwY(1); .5*rgbY(2)+.5*rgwY(2); .5*rgbY(3); .5*rgwY(3)];

%% Calculate Exponential Curves
load('FitCurves');

%% Find Code Values for Standard Illuminance
% Wall setup
syms x;
adaptlum = 40;
stdcv=double(solve(adaptlum==coeffwadaptwlumfit(1)*x^coeffwadaptwlumfit(2)-...
       coeffwadaptwlumfit(3), 'Real', true));
stdlum=coeffwallwlumfit(1)*stdcv^coeffwallwlumfit(2)-coeffwallwlumfit(3);
stdlumrY=rgbwY(1)*stdlum/100;
stdlumgY=rgbwY(2)*stdlum/100;
stdlumbY=rgbwY(3)*stdlum/100;
stdlumwY=rgbwY(4)*stdlum/100;
wallrstdcv=double(solve(stdlumrY==coeffwallrlumfit(1)*x^coeffwallrlumfit(2)-...
       coeffwallrlumfit(3), 'Real', true))
wallgstdcv=double(solve(stdlumgY==coeffwallglumfit(1)*x^coeffwallglumfit(2)-...
       coeffwallglumfit(3), 'Real', true))
wallbstdcv=double(solve(stdlumbY==coeffwallblumfit(1)*x^coeffwallblumfit(2)-...
       coeffwallblumfit(3), 'Real', true))
```

```
wallwstdcv=double(solve(stdlumwY==coeffwallwlumfit(1)*x^coeffwallwlumfit(2)-...
    coeffwallwlumfit(3), 'Real', true))

% Ceiling setup
fprintf('\n');
disp '100% Standard Illuminance';
stdlux = 21.5278-17.69;
stdcv=double(solve(stdlux==coeffceilingwhluxfit(1)*x^coeffceilingwhluxfit(2)-...
    coeffceilingwhluxfit(3), 'Real', true));
stdlum=coeffceilingwlumfit(1)*stdcv^coeffceilingwlumfit(2)-coeffceilingwlumfit(3);
stdlumrY=rgbwY(1)*stdlum/100;
stdlumgY=rgbwY(2)*stdlum/100;
stdlumbY=rgbwY(3)*stdlum/100;
stdlumwY=rgbwY(4)*stdlum/100;
rstdcv=double(solve(stdlumrY==coeffceilingrlumfit(1)*x^coeffceilingrlumfit(2)-...
    coeffceilingrlumfit(3), 'Real', true))
gstdcv=double(solve(stdlumgY==coeffceilingglumfit(1)*x^coeffceilingglumfit(2)-...
    coeffceilingglumfit(3), 'Real', true))
bstdcv=double(solve(stdlumbY==coeffceilingblumfit(1)*x^coeffceilingblumfit(2)-...
    coeffceilingblumfit(3), 'Real', true))
wstdcv=double(solve(stdlumwY==coeffceilingwlumfit(1)*x^coeffceilingwlumfit(2)-...
    coeffceilingwlumfit(3), 'Real', true))

%% Find Code Values for Standard Illuminance + 50%
fprintf('\n');
disp '150% Standard Illuminance';
syms y;
fiftystdlux = 1.5*21.5278-17.69;
fiftystdcv=double(solve(fiftystdlux==coeffceilingwhluxfit(1)*...
    y^coeffceilingwhluxfit(2)-coeffceilingwhluxfit(3), 'Real', true));
fiftystdlum=coeffceilingwlumfit(1)*fiftystdcv^coeffceilingwlumfit(2)-...
    coeffceilingwlumfit(3);
fiftystdlumrY=rgbwY(1)*fiftystdlum/100;
fiftystdlumgY=rgbwY(2)*fiftystdlum/100;
fiftystdlumbY=rgbwY(3)*fiftystdlum/100;
fiftystdlumwY=rgbwY(4)*fiftystdlum/100;
rfiftystdcv=double(solve(fiftystdlumrY==coeffceilingrlumfit(1)*...
    y^coeffceilingrlumfit(2)-coeffceilingrlumfit(3), 'Real', true))
gfiftystdcv=double(solve(fiftystdlumgY==coeffceilingglumfit(1)*...
    y^coeffceilingglumfit(2)-coeffceilingglumfit(3), 'Real', true))
bfiftystdcv=double(solve(fiftystdlumbY==coeffceilingblumfit(1)*...
    y^coeffceilingblumfit(2)-coeffceilingblumfit(3), 'Real', true))
wfiftystdcv=double(solve(fiftystdlumwY==coeffceilingwlumfit(1)*...
    y^coeffceilingwlumfit(2)-coeffceilingwlumfit(3), 'Real', true))

%% Find Code Values for double the Standard Illuminance
fprintf('\n');
disp '200% Standard Illuminance';
syms z;
doubstdlux = 2*21.5278-17.69;
doubstdcv=double(solve(doubstdlux==coeffceilingwhluxfit(1)*z^coeffceilingwhluxfit(2)-...
    coeffceilingwhluxfit(3), 'Real', true));
doubstdlum=coeffceilingwlumfit(1)*doubstdcv^coeffceilingwlumfit(2)-...
    coeffceilingwlumfit(3);
doubstdlumrY=rgbwY(1)*doubstdlum/100;
```

```
doubstdlumgY=rgbwY(2)*doubstdlum/100;
doubstdlumbY=rgbwY(3)*doubstdlum/100;
doubstdlumwY=rgbwY(4)*doubstdlum/100;
rdoubstdcv=double(solve(doubstdlumrY==coeffceilingrlumfit(1)*z^coeffceilingrlumfit(2)...
       -coeffceilingrlumfit(3), 'Real', true))
gdoubstdcv=double(solve(doubstdlumgY==coeffceilingglumfit(1)*z^coeffceilingglumfit(2)...
       -coeffceilingglumfit(3), 'Real', true))
bdoubstdcv=double(solve(doubstdlumbY==coeffceilingblumfit(1)*z^coeffceilingblumfit(2)...
       -coeffceilingblumfit(3), 'Real', true))
wdoubstdcv=double(solve(doubstdlumwY==coeffceilingwlumfit(1)*z^coeffceilingwlumfit(2)...
       -coeffceilingwlumfit(3), 'Real', true))
```

Desk Condition Calculations

```
%% Define Parameters
rgbxy = [.6915 .3054; .1363 .6787; .1464 .0489];
rgwxy = [.6915 .3054; .1363 .6787; .2491 .2070];
d65xy = [.3127 .3290];

%% Find Color Ratios
rgbY = bal_prim(rgbxy,d65xy,100);
rgwY = bal_prim(rgwxy,d65xy,100);
rgbwY = [.5*rgbY(1)+.5*rgwY(1); .5*rgbY(2)+.5*rgwY(2); .5*rgbY(3); .5*rgwY(3)];

%% Calculate Exponential Curves
load('FitCurves');

%% Find Code Values for Standard Illuminance
disp 'Wall setup';
syms x;
adaptlum = 40;
wstdcv=double(solve(adaptlum==coeffwadaptwlumfit(1)*x^coeffwadaptwlumfit(2)-...
       coeffwadaptwlumfit(3), 'Real', true));
wstdlum=coeffwallwlumfit(1)*wstdcv^coeffwallwlumfit(2)-coeffwallwlumfit(3);
wstdlumrY=rgbwY(1)*wstdlum/100;
wstdlumgY=rgbwY(2)*wstdlum/100;
wstdlumbY=rgbwY(3)*wstdlum/100;
wstdlumwY=rgbwY(4)*wstdlum/100;
wallrstdcv=double(solve(wstdlumrY==coeffwallrlumfit(1)*x^coeffwallrlumfit(2)-...
       coeffwallrlumfit(3), 'Real', true))
wallgstdcv=double(solve(wstdlumgY==coeffwallglumfit(1)*x^coeffwallglumfit(2)-...
       coeffwallglumfit(3), 'Real', true))
wallbstdcv=double(solve(wstdlumbY==coeffwallblumfit(1)*x^coeffwallblumfit(2)-...
       coeffwallblumfit(3), 'Real', true))
wallwstdcv=double(solve(wstdlumwY==coeffwallwlumfit(1)*x^coeffwallwlumfit(2)-...
       coeffwallwlumfit(3), 'Real', true))

fprintf('\n');
disp 'Desk setup';
syms x;
dstdcv=double(solve(adaptlum==coeffdadaptwlumfit(1)*x^coeffdadaptwlumfit(2)-...
       coeffdadaptwlumfit(3), 'Real', true));
dstdlum=coeffwallwlumfit(1)*dstdcv^coeffwallwlumfit(2)-coeffwallwlumfit(3);
dstdlumrY=rgbwY(1)*dstdlum/100;
```

```
dstdlumgY=rgbwY(2)*dstdlum/100;
dstdlumbY=rgbwY(3)*dstdlum/100;
dstdlumwY=rgbwY(4)*dstdlum/100;
deskrstdcv=double(solve(dstdlumrY==coeffwallrlumfit(1)*x^coeffwallrlumfit(2)-...
       coeffwallrlumfit(3), 'Real', true))
deskgstdcv=double(solve(dstdlumgY==coeffwallglumfit(1)*x^coeffwallglumfit(2)-...
       coeffwallglumfit(3), 'Real', true))
deskbstdcv=double(solve(dstdlumbY==coeffwallblumfit(1)*x^coeffwallblumfit(2)-...
       coeffwallblumfit(3), 'Real', true))
deskwstdcv=double(solve(dstdlumwY==coeffwallwlumfit(1)*x^coeffwallwlumfit(2)-...
       coeffwallwlumfit(3), 'Real', true))

fprintf('\n');
disp '100% Standard Illuminance';
stdlux = 21.5278-18.55;
wstdcv=double(solve(stdlux==coeffceilingwhluxfit(1)*x^coeffceilingwhluxfit(2)-...
       coeffceilingwhluxfit(3), 'Real', true));
stdlum=coeffceilingwlumfit(1)*wstdcv^coeffceilingwlumfit(2)-coeffceilingwlumfit(3);
stdlumrY=rgbwY(1)*stdlum/100;
stdlumgY=rgbwY(2)*stdlum/100;
stdlumbY=rgbwY(3)*stdlum/100;
stdlumwY=rgbwY(4)*stdlum/100;
rstdcv=double(solve(stdlumrY==coeffceilingrlumfit(1)*x^coeffceilingrlumfit(2)-...
       coeffceilingrlumfit(3), 'Real', true))
gstdcv=double(solve(stdlumgY==coeffceilingglumfit(1)*x^coeffceilingglumfit(2)-...
       coeffceilingglumfit(3), 'Real', true))
bstdcv=double(solve(stdlumbY==coeffceilingblumfit(1)*x^coeffceilingblumfit(2)-...
       coeffceilingblumfit(3), 'Real', true))
wstdcv=double(solve(stdlumwY==coeffceilingwlumfit(1)*x^coeffceilingwlumfit(2)-...
       coeffceilingwlumfit(3), 'Real', true))

%% Find Code Values for Standard Illuminance + 50%
fprintf('\n');
disp '150% Standard Illuminance';
syms y;
fiftystdlux = 1.5*21.5278-18.55;
fiftystdcv=double(solve(fiftystdlux==coeffceilingwhluxfit(1)*...
       y^coeffceilingwhluxfit(2)-coeffceilingwhluxfit(3), 'Real', true));
fiftystdlum=coeffceilingwlumfit(1)*fiftystdcv^coeffceilingwlumfit(2)-...
       coeffceilingwlumfit(3);
fiftystdlumrY=rgbwY(1)*fiftystdlum/100;
fiftystdlumgY=rgbwY(2)*fiftystdlum/100;
fiftystdlumbY=rgbwY(3)*fiftystdlum/100;
fiftystdlumwY=rgbwY(4)*fiftystdlum/100;
rfiftystdcv=double(solve(fiftystdlumrY==coeffceilingrlumfit(1)*...
       y^coeffceilingrlumfit(2)-coeffceilingrlumfit(3), 'Real', true))
gfiftystdcv=double(solve(fiftystdlumgY==coeffceilingglumfit(1)*...
       y^coeffceilingglumfit(2)-coeffceilingglumfit(3), 'Real', true))
bfiftystdcv=double(solve(fiftystdlumbY==coeffceilingblumfit(1)*...
       y^coeffceilingblumfit(2)-coeffceilingblumfit(3), 'Real', true))
wfiftystdcv=double(solve(fiftystdlumwY==coeffceilingwlumfit(1)*...
       y^coeffceilingwlumfit(2)-coeffceilingwlumfit(3), 'Real', true))

%% Find Code Values for double the Standard Illuminance
fprintf('\n');
```

```
disp '200% Standard Illuminance';
syms z;
doubstdlux = 2*21.5278-18.55;
doubstdcv=double(solve(doubstdlux==coeffceilingwhluxfit(1)*z^coeffceilingwhluxfit(2)-…
       coeffceilingwhluxfit(3), 'Real', true));
doubstdlum=coeffceilingwlumfit(1)*doubstdcv^coeffceilingwlumfit(2)-…
       coeffceilingwlumfit(3);
doubstdlumrY=rgbwY(1)*doubstdlum/100;
doubstdlumgY=rgbwY(2)*doubstdlum/100;
doubstdlumbY=rgbwY(3)*doubstdlum/100;
doubstdlumwY=rgbwY(4)*doubstdlum/100;
rdoubstdcv=double(solve(doubstdlumrY==coeffceilingrlumfit(1)*z^coeffceilingrlumfit(2)…
       -coeffceilingrlumfit(3), 'Real', true))
gdoubstdcv=double(solve(doubstdlumgY==coeffceilingglumfit(1)*z^coeffceilingglumfit(2)…
       -coeffceilingglumfit(3), 'Real', true))
bdoubstdcv=double(solve(doubstdlumbY==coeffceilingblumfit(1)*z^coeffceilingblumfit(2)…
       -coeffceilingblumfit(3), 'Real', true))
wdoubstdcv=double(solve(doubstdlumwY==coeffceilingwlumfit(1)*z^coeffceilingwlumfit(2)…
       -coeffceilingwlumfit(3), 'Real', true))
```

Appendix C

Python Source Code to Control the LEDs When Changing Illumination Conditions

```
# -*- coding: utf-8 -*-
"""
Created on Mon Oct 02 2017 09:14:28
Modified on Fri Nov 22 2019 14:43:34

@author: Sarah Bolton modified by Jeremy Miller
@setup: The DMX controllers should have their dip switches set to the following
        from left to right.
        Left: All up
        Left Middle: 1, 4, 5 down, the rest up
        Right Middle: 1, 5, 6 down, the rest up
        Right: 1, 4, 7 down, the rest up
@notes: this setup will make it so the smallest channels will correspond to the lights
        near the floor. The largest channels will be on the ceiling, with the largest
        channel being at the back of the room
"""

##################################################################
#                     REQUIRED ADMIN FUNCTIONS - Do Not Modify
##################################################################
#serial is PySerial, the serial port software for Python
import serial
import time
import random

#setup the dmx
#char 126 is 7E in hex. It's used to start all DMX512 commands
DMXOPEN=bytes([126])

#char 231 is E7 in hex. It's used to close all DMX512 commands
DMXCLOSE=bytes([231])

#I named the "output only dmx packet request" DMXINTENSITY as I don't have any moving
#fixtures. Char 6 is the label, I don't know what Char 1 and Char 2 mean but my
#sniffer log showed those values always to be the same so I guess #it's good enough.
DMXINTENSITY=bytes([6]) + bytes([1]) + bytes([2])

#This code initializes the communications. Char 3 is a request for the controller's
#parameters. I didn't bother reading that data, I'm just assuming it's an init string.
DMXINIT1=bytes([3]) + bytes([2]) + bytes([0]) + bytes([0]) + bytes([0])

#likewise, char 10 requests the serial number of the unit. I'm not receiving it
#or using it but the other softwares I tested did. You might want to.
DMXINIT2=bytes([10]) + bytes([2]) + bytes([0]) + bytes([0]) + bytes([0])

#open serial port 4. This is where the USB virtual port hangs on my machine. You might
#need to change this number. Find out what com port your DMX controller is on and
#subtract 1, the ports are numbered 0-3 instead of 1-4 (This may not be true on all
computers)
ser=serial.Serial("COM4")
```

```
#this writes the initialization codes to the DMX
ser.write( DMXOPEN+DMXINIT1+DMXCLOSE)
ser.write( DMXOPEN+DMXINIT2+DMXCLOSE)

#this sets up an array of 513 bytes, the first item in the array ( dmxdata[0] )
#is the previously mentioned spacer byte following the header. This makes the
#array math more obvious
dmxdata=[bytes([0])]*513

#senddmx accepts the 513 byte long data string to keep the state of all the
#channels, the channel number and the value for that channel.
#senddmx writes to the serial port then returns the modified 513 byte array
#data = current state of the lights
def senddmx(data):
       # join turns the array data into a string we can send down the DMX
       sdata=b''.join(data)
       # write the data to the serial port, this sends the data to your fixture
       ser.write(DMXOPEN+DMXINTENSITY+sdata+DMXCLOSE)
       # return the data with the new value in place
       return(data)

##################################################################
#                   FUNCTIONS TO MODIFY LIGHT CONDITIONS
##################################################################

#######
## This function turns all of the lights off. To call it type: Off(dmxdata)
def Off(dmxdata, unused=0):
    dmxdata=[bytes([0])]*513           #Creates an empty array of zeros
    # The second variable can be any number less than 513
    # Essentially, you are just writing an empty state to the comm port
    dmxdata = senddmx(dmxdata)  # writes the array to the DMX device
    return dmxdata                  #Return the current state

# This function sets the test condition for using only ceiling lights
def Ceiling(dmxdata, setting=0):
    dmxdata=Off(dmxdata)
    if setting==0:          #2 foot-candles horizontal illuminance of keyboard
           redintensity=24
           greenintensity=23
           blueintensity=17
           whiteintensity=15
    elif setting==1:        #3 foot-candles horizontal illuminance of keyboard
           redintensity=29
           greenintensity=28
           blueintensity=18
           whiteintensity=17
    elif setting==2:        #4 foot-candles horizontal illuminance of keyboard
           redintensity=32
           greenintensity=32
           blueintensity=20
           whiteintensity=17
    else:
           dmxdata=Off(dmxdata)
```

```
        return dmxdata
    for addr in range(0,4):
        if addr==0:
            intensity=redintensity
        elif addr==1:
            intensity=greenintensity
        elif addr==2:
            intensity=blueintensity
        elif addr==3:
            intensity=whiteintensity

        dmxdata[addr+37]=bytes([intensity])
        dmxdata[addr+43]=bytes([intensity])
        dmxdata[addr+53]=bytes([intensity])
        dmxdata[addr+59]=bytes([intensity])
        dmxdata[addr+65]=bytes([intensity])
        dmxdata[addr+71]=bytes([intensity])
        dmxdata[addr+81]=bytes([intensity])
        dmxdata[addr+87]=bytes([intensity])
        dmxdata[addr+93]=bytes([intensity])
    dmxdata = senddmx(dmxdata)  # writes the array to the DMX device

    nexttest = "n"
    while (nexttest.lower()!="y" and nexttest.lower()!="yes"):
        inputcmd = input("Proceed to Next Condition?: ").split()
        nexttest = inputcmd[0]
        if (nexttest.lower()!="y" and nexttest.lower()!="yes"):
            print("     !!! error")

    return dmxdata

#######
## This function sets the test condition for using wall & ceiling lights
def Wall(dmxdata, setting=0):
    dmxdata=Off(dmxdata)
    wallredintensity=47            #40 cd/m2 luminance of wall
    wallgreenintensity=44
    wallblueintensity=29
    wallwhiteintensity=22
    for addr in range(0,4):
        if addr==0:
            intensity=wallredintensity
        elif addr==1:
            intensity=wallgreenintensity
        elif addr==2:
            intensity=wallblueintensity
        elif addr==3:
            intensity=wallwhiteintensity
        dmxdata[addr+1]=bytes([intensity])
        dmxdata[addr+7]=bytes([intensity])
        dmxdata[addr+13]=bytes([intensity])
        dmxdata[addr+19]=bytes([intensity])
        dmxdata[addr+25]=bytes([intensity])
        dmxdata[addr+31]=bytes([intensity])
        dmxdata[addr+37]=bytes([intensity])
    if setting==0:           #2 foot-candles horizontal illuminance of keyboard
        redintensity=17
        greenintensity=16
        blueintensity=13
        whiteintensity=12
    elif setting==1:         #3 foot-candles horizontal illuminance of keyboard
        redintensity=23
        greenintensity=22
        blueintensity=17
        whiteintensity=15
    elif setting==2:         #4 foot-candles horizontal illuminance of keyboard
        redintensity=27
        greenintensity=26
        blueintensity=18
        whiteintensity=15
    else:
        dmxdata=Off(dmxdata)
        return dmxdata
    for addr in range(0,4):
        if addr==0:
            intensity=redintensity
        elif addr==1:
            intensity=greenintensity
        elif addr==2:
            intensity=blueintensity
        elif addr==3:
            intensity=whiteintensity

        dmxdata[addr+37]=bytes([intensity])
        dmxdata[addr+43]=bytes([intensity])
        dmxdata[addr+53]=bytes([intensity])
        dmxdata[addr+59]=bytes([intensity])
        dmxdata[addr+65]=bytes([intensity])
        dmxdata[addr+71]=bytes([intensity])
        dmxdata[addr+81]=bytes([intensity])
        dmxdata[addr+87]=bytes([intensity])
        dmxdata[addr+93]=bytes([intensity])
    dmxdata = senddmx(dmxdata)  # writes the array to the DMX device

    nexttest = "n"
    while (nexttest.lower()!="y" and nexttest.lower()!="yes"):
        inputcmd = input("Proceed to Next Condition?: ").split()
        nexttest = inputcmd[0]
        if (nexttest.lower()!="y" and nexttest.lower()!="yes"):
            print("     !!! error")

    return dmxdata

#######
## This function sets the test condition for using desk, wall & ceiling lights
def Desk(dmxdata, setting=0):
    dmxdata=Off(dmxdata)
    wallredintensity=47            #40 cd/m2 luminance of wall
    wallgreenintensity=44
    wallblueintensity=29
```

```
wallwhiteintensity=22
for addr in range(0,4):
        if addr==0:
                intensity=wallredintensity
        elif addr==1:
                intensity=wallgreenintensity
        elif addr==2:
                intensity=wallblueintensity
        elif addr==3:
                intensity=wallwhiteintensity

        dmxdata[addr+1]=bytes([intensity])
        dmxdata[addr+7]=bytes([intensity])
        dmxdata[addr+13]=bytes([intensity])
        dmxdata[addr+19]=bytes([intensity])
        dmxdata[addr+25]=bytes([intensity])
        dmxdata[addr+31]=bytes([intensity])
        dmxdata[addr+37]=bytes([intensity])
deskredintensity=78            #40 cd/m2 luminance w/ desk channel
deskgreenintensity=75
deskblueintensity=35
deskwhiteintensity=26
for addr in range(0,4):
        if addr==0:
                intensity=deskredintensity
        elif addr==1:
                intensity=deskgreenintensity
        elif addr==2:
                intensity=deskblueintensity
        elif addr==3:
                intensity=deskwhiteintensity
        dmxdata[addr+99]=bytes([intensity])
if setting==0:           #2 foot-candles horizontal illuminance of keyboard
        redintensity=16
        greenintensity=15
        blueintensity=13
        whiteintensity=12
elif setting==1:         #3 foot-candles horizontal illuminance of keyboard
        redintensity=22
        greenintensity=21
        blueintensity=16
        whiteintensity=16
elif setting==2:         #4 foot-candles horizontal illuminance of keyboard
        redintensity=26
        greenintensity=25
        blueintensity=17
        whiteintensity=16
else:
        dmxdata=Off(dmxdata)
        return dmxdata
for addr in range(0,4):
        if addr==0:
                intensity=redintensity
        elif addr==1:
                intensity=greenintensity
        elif addr==2:
            intensity=blueintensity
        elif addr==3:
            intensity=whiteintensity

        dmxdata[addr+37]=bytes([intensity])
        dmxdata[addr+43]=bytes([intensity])
        dmxdata[addr+53]=bytes([intensity])
        dmxdata[addr+59]=bytes([intensity])
        dmxdata[addr+65]=bytes([intensity])
        dmxdata[addr+71]=bytes([intensity])
        dmxdata[addr+81]=bytes([intensity])
        dmxdata[addr+87]=bytes([intensity])
        dmxdata[addr+93]=bytes([intensity])
    dmxdata = senddmx(dmxdata)  # writes the array to the DMX device

    nexttest = "n"
    while (nexttest.lower()!="y" and nexttest.lower()!="yes"):
        inputcmd = input("Proceed to Next Condition?: ").split()
        nexttest = inputcmd[0]
        if (nexttest.lower()!="y" and nexttest.lower()!="yes"):
            print("     !!! error")

    return dmxdata

#######
## This function turns off all lights, closes the serial port & stops the code
def Exit(dmxdata, unused=0):
    dmxdata=Off(dmxdata)
    return dmxdata              #Returns current state of lights

##################################################################
#                               MAIN CODE
##################################################################
#
dmxdata=Off(dmxdata)
cmd = "run"
while (cmd != "exit"):
    input_pid = input("Participant ID: ").split()
    participant = input_pid[0]
    if participant=="1":
        for setting in range(0,3):
            dmxdata=Wall(dmxdata, setting)
            dmxdata=Desk(dmxdata, setting)
            dmxdata=Ceiling(dmxdata, setting)
            dmxdata=Exit(dmxdata)
            cmd = "exit"
    elif participant=="2":
        for setting in range(2,-1,-1):
            dmxdata=Ceiling(dmxdata, setting)
            dmxdata=Wall(dmxdata, setting)
            dmxdata=Desk(dmxdata, setting)
            dmxdata=Exit(dmxdata)
            cmd = "exit"
    elif participant=="3":
       for setting in range(0,3):
            dmxdata=Desk(dmxdata, setting)
            dmxdata=Ceiling(dmxdata, setting)
            dmxdata=Wall(dmxdata, setting)
            dmxdata=Exit(dmxdata)
            cmd = "exit"
elif participant=="4":
       for setting in range(0,3):
            dmxdata=Ceiling(dmxdata, setting)
            dmxdata=Wall(dmxdata, setting)
            dmxdata=Desk(dmxdata, setting)
            dmxdata=Exit(dmxdata)
            cmd = "exit"
elif participant=="5":
       for setting in range(0,3):
            dmxdata=Desk(dmxdata, setting)
            dmxdata=Wall(dmxdata, setting)
            dmxdata=Ceiling(dmxdata, setting)
            dmxdata=Exit(dmxdata)
            cmd = "exit"
elif participant=="6":
       for setting in range(2,-1,-1):
            dmxdata=Wall(dmxdata, setting)
            dmxdata=Desk(dmxdata, setting)
            dmxdata=Ceiling(dmxdata, setting)
            dmxdata=Exit(dmxdata)
            cmd = "exit"
elif participant=="7":
       for setting in range(2,-1,-1):
            dmxdata=Desk(dmxdata, setting)
            dmxdata=Wall(dmxdata, setting)
            dmxdata=Ceiling(dmxdata, setting)
            dmxdata=Exit(dmxdata)
            cmd = "exit"
elif participant=="8":
       for setting in range(0,3):
            dmxdata=Ceiling(dmxdata, setting)
            dmxdata=Desk(dmxdata, setting)
            dmxdata=Wall(dmxdata, setting)
            dmxdata=Exit(dmxdata)
            cmd = "exit"
elif participant=="9":
       for setting in range(2,-1,-1):
            dmxdata=Wall(dmxdata, setting)
            dmxdata=Ceiling(dmxdata, setting)
            dmxdata=Desk(dmxdata, setting)
            dmxdata=Exit(dmxdata)
            cmd = "exit"
elif participant=="10":
       for setting in range(2,-1,-1):
            dmxdata=Desk(dmxdata, setting)
            dmxdata=Ceiling(dmxdata, setting)
            dmxdata=Wall(dmxdata, setting)
            dmxdata=Exit(dmxdata)
            cmd = "exit"
```

```
elif participant=="11":
       for setting in range(0,3):
            dmxdata=Wall(dmxdata, setting)
            dmxdata=Ceiling(dmxdata, setting)
            dmxdata=Desk(dmxdata, setting)
            dmxdata=Exit(dmxdata)
            cmd = "exit"
elif participant=="12":
       for setting in range(2,-1,-1):
            dmxdata=Ceiling(dmxdata, setting)
            dmxdata=Desk(dmxdata, setting)
            dmxdata=Wall(dmxdata, setting)
            dmxdata=Exit(dmxdata)
            cmd = "exit"
else:
       dmxdata=Exit(dmxdata)
       cmd = "exit"
```

Appendix D

Matlab Source Code for the Practice Scenario

```
% Clear the workspace and the screen
sca;
close all;
clearvars;
format compact;

hroot=groot; % get the root handle
set(hroot,'PointerLocation',[750,450]); % move cursor away from experiment screen

%% ----------------------------------------
%  Basic Necessary Set Up Functions
%  ----------------------------------------
KbName('UnifyKeyNames'); % Ensures KbName can be used without problems on any computer
PsychDefaultSetup(2); % Sets up PsychToolBox with some default values

screenNumber = 2;

white = WhiteIndex(screenNumber); % Define white
gray = .09; % This is the gray value that will be used for the background

backgrndclr = [gray gray gray 0];

Screen('Preference', 'VisualDebugLevel', 1); % Change the 'Welcome to Psychtoolbox'
screen to dark
Screen('Preference', 'Verbosity', 1);   % Only output critical errors
Screen('Preference', 'SkipSyncTests', 1); % Skip sync tests for demo purposes only

%% ---------------------------------------
%  Gabor Patch information Setup
%  ---------------------------------------
[window, rect] = PsychImaging('OpenWindow', screenNumber, backgrndclr, [], 500, 2, [],
[],  kPsychNeed32BPCFloat); % Opens the PsychToolbox screen

GaborSize = round(rect(4)/13); % Dimension of the region to draw the Gabor in pixels -
--- Value of round(rect(4)/13) gets the Gabor patch size to 0.75 inch.

sigma = GaborSize / 7; % Determines how well the edges of the Gabor blends to the
background, if number is too low then the Gabor will start to distort. 7 seems to be
best value

aspectRatio = 1.0; % Does not change as value is changed. Keeps Gabor with equal
length and width
phase = 0;      % Keep at zero, so the Gabor is in phase

numCycles=16; % One Cycle = Grey-Black-Grey-White-Grey i.e. One Black & One White Lobe

freq = numCycles / GaborSize; % Spatial Frequency (Cycles Per Pixel)
```

```
contrastPreMultiplicator = 1; % Kept at a constant so that only contrast needs change

backgroundOffset = backgrndclr; % This is the background color of the Gabor patch &
needs to match screen background color
disableNorm = 1; % Disables certain settings that may distort the Gabor patch

drawGabor = CreateProceduralGabor(window, GaborSize, GaborSize, [],...
    backgroundOffset, disableNorm, contrastPreMultiplicator);
% Provides the input for the Screen('DrawTextures') function

%% ---------------------------------------
%  Fixation Cross information
%  ---------------------------------------
% This is used between each Gabor patch to keep participant attention in the center of
the screen

fixatngray=((white-gray)*0.2)+gray; % 20% of the difference from the background color
to max white
fixationcolor = [fixatngray fixatngray fixatngray 0];

crossSize = 13; % Size I felt was appropriate, can be changed
crossWidth = 5; % Thickness of the cross. Can be changed if need be

crossLines = [-crossSize, 0; crossSize, 0; 0, -crossSize; 0, crossSize];
crossLines = crossLines'; % Puts the values in a way that the 'DrawLines' function can
read them

xCenter = rect(3)/2; % Defines the center of the screen. Usually done automatically,
but may have problems if different sized screens are used
yCenter = rect(4)/2;

ScreenCenter = [xCenter, yCenter]; % Makes the center point an input for the function
used later

%% ---------------------------------------
%  Beep Information
%  ---------------------------------------
% This is the information needed to make the beep that plays if participant is taking
too long.

Au_sf=48000; % sampling rate of audio beep (Hz)
Au_vol=0.4; % audio volume percent intensity (.2 = 20% intesity)
Au_freq=500; % In Hz
Au_dur=0.1; % time sound plays for in seconds

% Initialize Audio
InitializePsychSound(1);
PsychPortAudio('Verbosity', 1);
pahandle=PsychPortAudio('Open',[],1,1,Au_sf,2); % Like Screen('open') but for Audio
PsychPortAudio('Volume',pahandle,Au_vol); % Sets up the volume input for the function
myBeep=MakeBeep(Au_freq,Au_dur,Au_sf); % Makes the beep with the information provided
PsychPortAudio('FillBuffer',pahandle,[myBeep; myBeep]); % Fills the audio buffer so it
is ready to play the noise
```

```
%% ----------------------------------------
%  Quest Values
%  ----------------------------------------
% Setting up the values for the Quest function being used to determine threshold for
the participant

tGuess= -1.4; % The prior threshold estimate for the participant, based on the
previous run
tGuessSd = -1.9; % Standard deviation for the guess (Kept high for good range)
pThreshold = .92; % The percentage (.82=82%) that the participant is expected to get
correct with the above provided guess
beta = .1;
delta = .01;
gamma = 0;
grain = .00001; % Step size, or the amount change in contrast from one Gabor patch to
the next
range = 5;
plotIt = 0;

q=QuestCreate(tGuess,tGuessSd,pThreshold,beta,delta,gamma,grain,range,plotIt);
% initial creation of the estimate value by Quest
t = QuestQuantile(q); % Function used to get the mean from Quest above

%% ----------------------------------------
%  Values necessary for multiple trials
%  ----------------------------------------
% Values used in the loop that do not belong anywhere specific or need to be
constantly changing

ifi = Screen('GetFlipInterval', window); % Gets time screen was flipped, later used
for accuracy
topPriorityLevel = MaxPriority(window); % Places the window as top priority, or where
the most effort from the computer will be put to ensure highest time accuracy
Priority(topPriorityLevel);
vbl = Screen('Flip', window); % Gets time of when the screen is flipped, also used
later to provide highest accuracy possible.

KeyIsDown = 0; % Later used to help define if a key is pressed or not

le = 45; % These are the two angles used for the Gabor patch (le = left leaning)
ri = -45; % These are the two angles used for the Gabor patch (ri = right leaning)
orientation=[le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri
le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le
ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri
le ri le ri le ri le ri le ri le ri le ri le ri];
orrand = Shuffle(orientation); % Randomizes the order in which the orientations are
provided
fixtime = linspace(1,1.49,100); % Amount of time the fixation cross is on the screen,
between 1 to 1.5 seconds
randfixtime = Shuffle(fixtime); % Randomizing the amount of time each fixation cross
is on the screen
GaborDelay = linspace(1.5,1.99,100); % Time between fixation and Gabor Patch appears
on screen, between 1.5 to 2 seconds
```

```
RandTimeBeforeGabor = Shuffle(GaborDelay);% Randomizing the time between when the beep
noise is made and when each Gabor Patch appears on screen

%% ----------------------------------------
%  Ask if Ready to Begin Scenario
%  ----------------------------------------
go_on=0;
while go_on==0
    Question='Ready to Proceed?';
    Decisioninput=questdlg(Question,'Time to Move Forward','Yes','No','Yes');
    if strcmp(Decisioninput,'Yes')==1
        go_on=1;
    elseif strcmp(Decisioninput,'No')==1
        go_on=0;
    else
        break;
    end % decision if statement
end % ready dialog box while loop

%% ---------------------------------------
%  The Multiple Trials For Loop
%  ---------------------------------------
Screen('Flip', window);
for trial=1:5
    contrast = 10^(t);
    propertiesMat = [phase+180, freq, sigma, contrast, aspectRatio, 0, 0, 0]; %Another
set of inputs for the Gabor Patch
    Screen('DrawLines', window,crossLines,crossWidth,fixationcolor,ScreenCenter);
%Draws Fixation Cross on screen before Gabor
    Screen('Flip', window); % Displays the fixation cross on the screen
    WaitSecs(randfixtime(trial)); % keeps the fixation cross on the screen for a
randomized time between values in (fixtime)

    Screen('Flip', window); % Flips the screen again to get rid of the fixation cross
    WaitSecs(RandTimeBeforeGabor(trial)); %Waits a randomized time within times
provided in TimeBeforeGabor

    [KeyIsDown, pressedSecs, keyCode] = KbCheck(-1); %Defines KbCheck which will be
used to see if a button is pressed later on

    while KeyIsDown % wait for no key press
        [KeyIsDown,~,keyCode]=KbCheck;
    end

    % Starts loops that will draw the Gabor patch
    if ~KeyIsDown
        Screen('DrawTextures',window,drawGabor,[],[],orrand(trial), [], [], [], [],...
        kPsychDontDoRotation, propertiesMat'); % Finally draws the Gabor patch on the
(back) screen
        vbl = Screen('Flip', window, vbl); % 'Flips' Screen, displaying Gabor patch.
        FlipTime = GetSecs; %Records time the screen flipped. Used for response time
        TiltAnswer=0;
        % get the highest and lowest code values of the gabor patch for
        % relative contrast calculations later
        imageArray=Screen('GetImage', window, rect);
        lightgaborcv=max(max(max(imageArray)));
        darkgaborcv=min(min(min(imageArray)));

        while TiltAnswer==0 % Providing values that will be used to determine how the
loop moves forward
            KeyIsDown=0;
            keyCode=0;
            starttimer=GetSecs;
            while KeyIsDown==0 % continue checking until a key is pressed
                [KeyIsDown,~,keyCode]=KbCheck;
                runningtimer=GetSecs;
                elapsedtime=runningtimer-starttimer;
                if (elapsedtime)>=7
                    PlayTime = PsychPortAudio('Start',pahandle,1,0,1); % Plays a tone
to indicate taking too long
                end
            end
            AnswerTime = GetSecs;
            RspnsTime = AnswerTime-FlipTime;
            response=find(keyCode);
            if response==37 % Response = 37 means that the left arrow key was pressed
                TiltAnswer=1; %Stop looking for answer and get out of small while loop
                Tilt='Left'; % Records all the participant's answer
                if orrand(trial) == le  %The Gabor is tilting to the left (45 degrees)
                    Accuracy = 'Correct';
                    reply = true; % Used for Quest funtion
                elseif orrand(trial)==ri % The Gabor is tilting to the right (-45 deg)
                    Accuracy = 'Incorrect';
                    reply = false; % Used for Quest funtion
                end
            elseif response==39 % Response =3 9 means the right arrow key was pressed
                TiltAnswer=1; %Stop looking for answer and get out of small while loop
                Tilt='Right'; % Records all the participant's answer
                if orrand(trial)==ri % The Gabor is tilting to the right (-45 degees)
                    Accuracy = 'Correct';
                    reply = true;
                elseif orrand(trial)==le  % The Gabor is tilting to the left (45 deg)
                    Accuracy = 'Incorrect';
                    reply = false;
                end
            end % right tilt if statement
        end % waiting for tilt answer while statement
        fprintf('Trial: %i, Tilt: %i, Accuracy: %s\n', trial, orrand(trial),Accuracy);
        q=QuestUpdate(q,t,reply); % Updates the Quest function to give new estimated
threshold values
        t=QuestQuantile(q); % gets intensity value of the QUEST function
    end % detection loop
end % Number of trials for loop

%% ----------------------------------------
% Wrap it all up
%  ----------------------------------------
Screen('FillRect', window, fixationcolor); % changes the background to tell
participant to standby
```

```
Screen('Flip', window); %Flips the screen to get rid of any gabor patch
WaitSecs(1); % Waits 1 seconds before resuming normal background test conditions
Screen('FillRect', window, backgrndclr);
Screen('Flip', window);

fclose('all');
sca;     % Clear screen
Screen('CloseAll');

fprintf('Practice code completed\n');

hroot=groot; % get the root handle
set(hroot,'PointerLocation',[750,450]);
```

Appendix E

Matlab Source Code to Control the Experiment

```
%% ----------------------------------------
%  Participant Test Code
%  ----------------------------------------
%  Runs the pattern generating code to run one dark background condition and
%  one light background condition. Calls the first pattern set, increments
%  the run number and calls the second pattern set.

go_on=0;
while go_on==0
    prompt={'Participant ID:','Day Number:','Run Number:','Lighting Condition'};
    dlgboxtitle='Participant Information';
    defaultanswer={'-','-','-','-'};
    answer=inputdlg(prompt, dlgboxtitle, 1, defaultanswer, 'on')';

    ParticipantID=sprintf('%s%s\n','Participant  ',answer{1});
    Day=sprintf('%s%s\n','Day  ',answer{2});
    Run=sprintf('%s%s\n','Run  ',answer{3});
    LightCond=sprintf('%s%s\n\n','Lighting Condition  ',answer{4});

    Question=[ParticipantID Day Run LightCond '         Correct?'];
    Decisioninput=questdlg(Question,'Confirm Info','Yes','No','Yes');
    if strcmp(Decisioninput,'Yes')==1
        go_on=1;
    elseif strcmp(Decisioninput,'No')==1
        go_on=0;

    else
        break;
    end % decision if statement
end % participant info dialog box while loop

info = {answer{1},answer{2},answer{3},answer{4}};

Pattern_Code % run 1st set of patterns

answer{3} = str2double(answer{3});
answer{3} = answer{3}+1;    % increment run number
answer{3} = int2str(answer{3});
answer{4} = answer{3};
info = {answer{1},answer{2},answer{3},answer{4}};
Pattern_Code % run 2nd set of patterns
```

```
%% ----------------------------------------
%  Pattern_Code
%  ----------------------------------------
%  Gabor Pattern Generating Code called from Participant Test Code

format compact;

hroot=groot; % get the root handle
set(hroot,'PointerLocation',[750,450]); % move cursor away from
experiment screen

%% ----------------------------------------
%  Set Background Condition
%  ----------------------------------------
dark=.035;
light=.40;
if (strcmp(answer{3}, '1') | ...
        strcmp(answer{3}, '3') | ...
        strcmp(answer{3}, '5') | ...
        strcmp(answer{3}, '7') | ...
        strcmp(answer{3}, '9') | ...
        strcmp(answer{3}, '11') | ...
        strcmp(answer{3}, '13') | ...
        strcmp(answer{3}, '15') | ...
        strcmp(answer{3}, '17') | ...
        strcmp(answer{3}, '19'))==1
    gray = dark; % This is the gray value used for the background
    contrastPreMultiplicator = 0.43; % Kept at a constant so that only
contrast needs change
else
    gray = light; % This is the gray value used for the background
    contrastPreMultiplicator = 0.5; % Kept at a constant so that only
contrast needs change
end

info{5} = gray;

%% ----------------------------------------
%  Basic Necessary Set Up Functions
%  ----------------------------------------
filename = 'participantdata.xlsx';

emptyrow=xlsread(filename,1,'R1');

KbName('UnifyKeyNames'); % Ensures KbName can be used without problems
on any computer
PsychDefaultSetup(2); % Sets up PsychToolBox with some default values

screenNumber = 2;

white = WhiteIndex(screenNumber); % Define black, white and grey
backgrndclr = [gray gray gray 0];
```

```
Screen('Preference', 'VisualDebugLevel', 1); % Change the 'Welcome to
Psychtoolbox' screen to dark
Screen('Preference', 'Verbosity', 1);    % Only output critical errors
Screen('Preference', 'SkipSyncTests', 1);

%% ---------------------------------------
%  Gabor Patch information Setup
%  ---------------------------------------
[window, rect] = PsychImaging('OpenWindow', screenNumber, backgrndclr,
[], 500, 2,...
    [], [],  kPsychNeed32BPCFloat); % Opens the PsychToolbox screen

GaborSize = round(rect(4)/13); % Dimension of the region to draw the
Gabor in pixels ---- Value of round(rect(4)/12.6) gets the Gabor patch
size to 0.75 inch.

sigma = GaborSize / 7; % Determines how well the edges of the Gabor
blends to the background, if number is too low then the Gabor will
start to distort. 7 seems to be best value

aspectRatio = 1.0; % Does not change as value is changed. Keeps Gabor
with equal length and width
phase = 0;          % Keep at zero, so the Gabor is in phase

numCycles = 16;  % One Cycle = Grey-Black-Grey-White-Grey i.e. One
Black and One White Lobe

freq = numCycles / GaborSize; % Spatial Frequency (Cycles Per Pixel)

backgroundOffset = backgrndclr; % This is the background color of the
Gabor patch & needs to match screen background color
disableNorm = 1; % Disables certain settings that may cause distortion
in the Gabor patch

drawGabor = CreateProceduralGabor(window, GaborSize, GaborSize, [],...
    backgroundOffset, disableNorm, contrastPreMultiplicator);
% Provides the input for the Screen('DrawTextures') function

%% ---------------------------------------
%  Fixation Cross information
%  ---------------------------------------
% This is used between each Gabor patch to keep participant attention
in the center of the screen

fixatngray=((white-gray)*0.2)+gray; % 20% of the difference from the
background color to max white
fixationcolor = [fixatngray fixatngray fixatngray 0];

crossSize = 13; % Size I felt was appropriate, can be changed
crossWidth = 5; % Thickness of the cross. Can be changed if need be

crossLines = [-crossSize, 0; crossSize, 0; 0, -crossSize; 0,
crossSize];
```

```
crossLines = crossLines'; % Puts the values in a way that the
'DrawLines' function can read them

xCenter = rect(3)/2; % Defines the center of the screen. Usually done
automatically, but may have problems if different sized screens are
used
yCenter = rect(4)/2;

ScreenCenter = [xCenter, yCenter]; % Makes the center point an input
for the function used later

%% ---------------------------------------
%  Beep Information
%  ---------------------------------------
% This is the information needed to make the beep that plays if
participant is taking too long.

Au_sf=48000; % sampling rate of audio beep (Hz)
Au_vol=0.4; % audio volume percent intensity (.2 = 20% intesity)
Au_freq=500; % In Hz
Au_dur=0.1; % time sound plays for in seconds

% Initialize Audio
InitializePsychSound(1);
PsychPortAudio('Verbosity', 1);
pahandle=PsychPortAudio('Open',[],1,1,Au_sf,2); % Like Screen('open')
but for Audio
PsychPortAudio('Volume',pahandle,Au_vol); % Sets up the volume input
for the function
myBeep=MakeBeep(Au_freq,Au_dur,Au_sf); % Makes the beep with the
information provided
PsychPortAudio('FillBuffer',pahandle,[myBeep; myBeep]); % Fills the
audio buffer so it is ready to play the noise

%% ---------------------------------------
%  Quest Values
%  ---------------------------------------
%Setting up the values for the Quest function being used to determine
threshold for the participant

tGuess= -1.2; % The prior threshold estimate for the participant, based
on the previous run
tGuessSd = -1.4; % Standard deviation for the guess (Kept high for good
range)
pThreshold = .82; % The percentage (.82=82%) that the participant is
expected to get correct with the above provided guess
beta = 3;
delta = .01;
gamma = 0;
grain = .00001; % Step size, or the amount change in contrast from one
Gabor patch to the next
range = 2.5;
% range = .5;
plotIt = 0;
```

```
q
=QuestCreate(tGuess,tGuessSd,pThreshold,beta,delta,gamma,grain,range,pl
otIt); %initial creation of the estimate value by Quest
t = QuestQuantile(q); % Function used to get the mean from Quest above

%% ---------------------------------------
%  Values necessary for multiple trials
%  ---------------------------------------
% Values used in the loop that do not belong anywhere specific or need
to be constantly changing

ifi = Screen('GetFlipInterval', window); % Gets time screen was
flipped, later used for accuracy
topPriorityLevel = MaxPriority(window); % Places the window as top
priority, or where the most effort from the computer will be put to
ensure highest time accuracy
Priority(topPriorityLevel);
vbl = Screen('Flip', window); % Gets time of when the screen is
flipped, also used later to provide highest accuracy possible.

KeyIsDown = 0; % Later used to help define if a key is pressed or not

le = 45; % These are the two angles used for the Gabor patch (le = left
leaning)
ri = -45; % These are the two angles used for the Gabor patch (ri =
right leaning)
orientation=[le ri le ri le ri le ri le ri le ri le ri le ri le ri le
ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le
ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le
ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le ri le
ri le ri le ri le ri];
orrand = Shuffle(orientation); % Randomizes the order in which the
orientations are provided
fixtime = linspace(1,1.49,100); % Amount of time the fixation cross is
on the screen, between 1 to 1.5 seconds
randfixtime = Shuffle(fixtime); % Randomizing the amount of time each
fixation cross is on the screen
GaborDelay = linspace(1.5,1.99,100); % Time between fixation and Gabor
Patch appears on screen, between 1.5 to 2 seconds
RandTimeBeforeGabor = Shuffle(GaborDelay);% Randomizing the time
between when the beep noise is made and when each Gabor Patch appears
on screen

%% ----------------------------------------
%  Eye Accommodation Timer
%  ----------------------------------------
if ((gray==light) | (strcmp(answer{3}, '1')==1)) % no timer needed -
1st timer done manually during practice round
    go_on=1;
elseif gray==dark
    go_on=0;
end
while go_on==0
```

```
    if (strcmp(answer{3}, '3') | ...
            strcmp(answer{3}, '9') | ...
            strcmp(answer{3}, '15'))==1
        eyetimer=180; % Waits 3 minutes for eye accommodation to new
lighting scenario
    else
        eyetimer=60; % Waits 1 minutes for eye accommodation to similar
lighting scenario
    end
    Question=sprintf('Start %i second Timer?',eyetimer);
    Decisioninput=questdlg(Question,'Accommodation Timer','Yes','No',
'Skip','Yes');
    if strcmp(Decisioninput,'Yes')==1
        go_on=1;
        fprintf('Running %i second timer...\n',eyetimer);
        WaitSecs(eyetimer);
    elseif strcmp(Decisioninput,'No')==1
        go_on=0;
    else
        break;
    end % decision if statement
end % ready dialog box while loop

%% ----------------------------------------
%  Ask if Ready to Begin Scenario
%  ----------------------------------------
go_on=0;
while go_on==0
    Question='Ready to Proceed?';
    Decisioninput=questdlg(Question,'Time Limit
Reached','Yes','No','Yes');
    if strcmp(Decisioninput,'Yes')==1
        go_on=1;
    elseif strcmp(Decisioninput,'No')==1
        go_on=0;
    else
        break;
    end % decision if statement
end % ready dialog box while loop

%% ---------------------------------------
%  The Multiple Trials For Loop
%  ---------------------------------------
Screen('Flip', window);
for trial=1:20
    contrast = 10^(t);
    propertiesMat = [phase+180, freq, sigma, contrast, aspectRatio, 0,
0, 0]; %Another set of inputs for the Gabor Patch
    Screen('DrawLines',
window,crossLines,crossWidth,fixationcolor,ScreenCenter); %Draws
Fixation Cross on screen before Gabor
    Screen('Flip', window); % Displays the fixation cross on the screen
    WaitSecs(randfixtime(trial)); % keeps the fixation cross on the
screen for a randomized time between values in (fixtime)
```

```
    Screen('Flip', window); % Flips the screen again to get rid of the 
fixation cross
    WaitSecs(RandTimeBeforeGabor(trial)); %Waits a randomized time 
within times provided in TimeBeforeGabor

    [KeyIsDown, pressedSecs, keyCode] = KbCheck(-1); %Defines KbCheck 
which will be used to see if a button is pressed later on

    while KeyIsDown % wait for no key press
        [KeyIsDown,~,keyCode]=KbCheck;
    end

    %Starts loops that will draw the Gabor patch
    if ~KeyIsDown
        Screen('DrawTextures', window, drawGabor, [], [], 
orrand(trial), [], [], [], [],...
        kPsychDontDoRotation, propertiesMat'); % Finally draws the 
Gabor patch on the (back) screen
        vbl = Screen('Flip', window, vbl); % 'Flips' Screen, displaying 
Gabor patch.
        FlipTime = GetSecs; %Records time the screen flipped. To be 
used for response time
        TiltAnswer=0;
        % get the highest and lowest code values of the gabor patch for 
relative contrast
        imageArray=Screen('GetImage', window, rect);
        lightgaborcv=max(max(max(imageArray)));
        darkgaborcv=min(min(min(imageArray)));

        while TiltAnswer==0 % Providing values that will be used to 
determine how the loop moves forward
            KeyIsDown=0;
            keyCode=0;
            starttimer=GetSecs;
            while KeyIsDown==0 % continue checking until a key is 
pressed
                [KeyIsDown,~,keyCode]=KbCheck;
                runningtimer=GetSecs;
                elapsedtime=runningtimer-starttimer;
                if (elapsedtime)>=8
                    PlayTime = PsychPortAudio('Start',pahandle,1,0,1); 
% Plays a tone to indicate taking too long
                end
            end
            AnswerTime = GetSecs;
            RspnsTime = AnswerTime-FlipTime;
            response=find(keyCode);
            if response==37 % Response = 37 means that the left arrow 
key was pressed
                TiltAnswer=1; % Stop looking for answer and get out of 
small while loop
                Tilt='Left'; % Records all the participant's answer
                if orrand(trial) == le  % The Gabor is tilting to the 
left (45 degrees)
```

```
                    Accuracy = 'Correct';
                    reply = true; % Used for Quest function
                elseif orrand(trial) == ri % The Gabor is tilting to
the right (-45 degrees)
                    Accuracy = 'Incorrect';
                    reply = false; % Used for Quest function
                end
            elseif response==39 % Response = 39 means that the right
arrow key was pressed
                TiltAnswer=1; % Stop looking for answer and get out of
small while loop
                Tilt='Right'; % Records all the participant's answers
                if orrand(trial) == ri % The Gabor is tilting to the
right (-45 degrees)
                    Accuracy = 'Correct';
                    reply = true;
                elseif orrand(trial) == le  % The Gabor is tilting to
the left (45 degrees)
                    Accuracy = 'Incorrect';
                    reply = false;
                end
            end % right tilt if statement
        end % waiting for tilt answer while statement
        fprintf('Trial: %i, Accuracy: %s\n',trial,Accuracy);
        q=QuestUpdate(q,t,reply); % Updates the Quest function to give
new estimated threshold values
        t=QuestQuantile(q); % gets intensity value of the QUEST
function
    end % detection loop

    % The following commands write all data acquired into Excel file
    nextrow=sprintf('%s%d','A',trial+emptyrow); % find next empty row
to print to
    results = {contrast, orrand(trial), Tilt, Accuracy, RspnsTime,
lightgaborcv, darkgaborcv};
    savedata = [info, results]; % compile data
    xlswrite(filename,savedata,1,nextrow); % write to Excel file

end % Number of trials for loop

xlswrite(filename,emptyrow+trial,1,'R1'); % update next empty row
indicator in excel

%% ----------------------------------------
%  Wrap it all up
%  ----------------------------------------
Screen('FillRect', window, fixationcolor); % changes the background to
tell participant to standby
Screen('Flip', window); % Flips the screen to get rid of any Gabor
patch
WaitSecs(1); % Waits 1 second before resuming normal background test
conditions
Screen('FillRect', window, backgrndclr);
Screen('Flip', window);
```

```
fclose('all');

sca;     % Clear screen
Screen('CloseAll');

fprintf('Run %s completed\n', answer{3});

hroot=groot; % get the root handle
set(hroot,'PointerLocation',[750,450]);
```

Appendix F

Matlab Source Code to Calculate Contrast Thresholds

All Participants, Grouped by Background Shade

```
close all;
format compact;

%% Define Parameters
plt2on = true;
myfile = 'participantdata.xlsx';

%% Load Participant Data
VertIllum = xlsread(myfile, 'F2:F4801');
Background = xlsread(myfile, 'K2:K4801');
Accuracy = xlsread(myfile, 'P2:P4801');
ResponseTime = xlsread(myfile, 'Q2:Q4801');
Michelson = xlsread(myfile, 'Z2:Z4801');
RawMichelson = xlsread(myfile, 'AA2:AA4801');

%% Create Data Groups
% All Participants
AllNoDark = [1-Michelson(VertIllum==0 & Background==.035),Accuracy(VertIllum==0 & ...
       Background==.035)];
AllNoGray = [1-Michelson(VertIllum==0 & Background==.4),Accuracy(VertIllum==0 & ...
       Background==.4)];
All2DDark = [1-Michelson(VertIllum==9.58 & Background==.035),Accuracy(VertIllum==9.58...
        & Background==.035)];
All2DGray = [1-Michelson(VertIllum==9.58 & Background==.4),Accuracy(VertIllum==9.58 &...
        Background==.4)];
All2WDark = [1-Michelson(VertIllum==9.75 & Background==.035),Accuracy(VertIllum==9.75...
        & Background==.035)];
All2WGray = [1-Michelson(VertIllum==9.75 & Background==.4),Accuracy(VertIllum==9.75 &...
        Background==.4)];
All2CDark = [1-Michelson(VertIllum==12.24 & Background==.035),Accuracy(VertIllum==...
       12.24 & Background==.035)];
All2CGray = [1-Michelson(VertIllum==12.24 & Background==.4),Accuracy(VertIllum==12.24...
        & Background==.4)];
All3DDark = [1-Michelson(VertIllum==17.18 & Background==.035),Accuracy(VertIllum==...
       17.18 & Background==.035)];
All3DGray = [1-Michelson(VertIllum==17.18 & Background==.4),Accuracy(VertIllum==17.18...
        & Background==.4)];
All3WDark = [1-Michelson(VertIllum==17.21 & Background==.035),Accuracy(VertIllum==...
       17.21 & Background==.035)];
All3WGray = [1-Michelson(VertIllum==17.21 & Background==.4),Accuracy(VertIllum==17.21...
        & Background==.4)];
All3CDark = [1-Michelson(VertIllum==22.11 & Background==.035),Accuracy(VertIllum==...
       22.11 & Background==.035)];
All3CGray = [1-Michelson(VertIllum==22.11 & Background==.4),Accuracy(VertIllum==22.11...
        & Background==.4)];
All4DDark = [1-Michelson(VertIllum==23.67 & Background==.035),Accuracy(VertIllum==...
       23.67 & Background==.035)];
All4DGray = [1-Michelson(VertIllum==23.67 & Background==.4),Accuracy(VertIllum==23.67...
        & Background==.4)];
All4WDark = [1-Michelson(VertIllum==23.93 & Background==.035),Accuracy(VertIllum==…
       23.93 & Background==.035)];
All4WGray = [1-Michelson(VertIllum==23.93 & Background==.4),Accuracy(VertIllum==23.93…
        & Background==.4)];
All4CDark = [1-Michelson(VertIllum==28.9 & Background==.035),Accuracy(VertIllum==28.9…
        & Background==.035)];
All4CGray = [1-Michelson(VertIllum==28.9 & Background==.4),Accuracy(VertIllum==28.9 &…
        Background==.4)];

%% Calculate Survior Functions
[param_AllNoDark, ci_AllNoDark] = wblfit(AllNoDark(:,1),0.05,AllNoDark(:,2));
[param_AllNoGray, ci_AllNoGray] = wblfit(AllNoGray(:,1),0.05,AllNoGray(:,2));
[param_All2DDark, ci_All2DDark] = wblfit(All2DDark(:,1),0.05,All2DDark(:,2));
[param_All2DGray, ci_All2DGray] = wblfit(All2DGray(:,1),0.05,All2DGray(:,2));
[param_All2WDark, ci_All2WDark] = wblfit(All2WDark(:,1),0.05,All2WDark(:,2));
[param_All2WGray, ci_All2WGray] = wblfit(All2WGray(:,1),0.05,All2WGray(:,2));
[param_All2CDark, ci_All2CDark] = wblfit(All2CDark(:,1),0.05,All2CDark(:,2));
[param_All2CGray, ci_All2CGray] = wblfit(All2CGray(:,1),0.05,All2CGray(:,2));
[param_All3DDark, ci_All3DDark] = wblfit(All3DDark(:,1),0.05,All3DDark(:,2));
[param_All3DGray, ci_All3DGray] = wblfit(All3DGray(:,1),0.05,All3DGray(:,2));
[param_All3WDark, ci_All3WDark] = wblfit(All3WDark(:,1),0.05,All3WDark(:,2));
[param_All3WGray, ci_All3WGray] = wblfit(All3WGray(:,1),0.05,All3WGray(:,2));
[param_All3CDark, ci_All3CDark] = wblfit(All3CDark(:,1),0.05,All3CDark(:,2));
[param_All3CGray, ci_All3CGray] = wblfit(All3CGray(:,1),0.05,All3CGray(:,2));
[param_All4DDark, ci_All4DDark] = wblfit(All4DDark(:,1),0.05,All4DDark(:,2));
[param_All4DGray, ci_All4DGray] = wblfit(All4DGray(:,1),0.05,All4DGray(:,2));
[param_All4WDark, ci_All4WDark] = wblfit(All4WDark(:,1),0.05,All4WDark(:,2));
[param_All4WGray, ci_All4WGray] = wblfit(All4WGray(:,1),0.05,All4WGray(:,2));
[param_All4CDark, ci_All4CDark] = wblfit(All4CDark(:,1),0.05,All4CDark(:,2));
[param_All4CGray, ci_All4CGray] = wblfit(All4CGray(:,1),0.05,All4CGray(:,2));

%% Compute Threshold
inv_Thshld_AllNoDark = wblinv(.75, param_AllNoDark(1), param_AllNoDark(2));
lbinv_Thshld_AllNoDark = wblinv(.75, ci_AllNoDark(1,1), ci_AllNoDark(1,2));
ubinv_Thshld_AllNoDark = wblinv(.75, ci_AllNoDark(2,1), ci_AllNoDark(2,2));
inv_Thshld_AllNoGray = wblinv(.75, param_AllNoGray(1), param_AllNoGray(2));
lbinv_Thshld_AllNoGray = wblinv(.75, ci_AllNoGray(1,1), ci_AllNoGray(1,2));
ubinv_Thshld_AllNoGray = wblinv(.75, ci_AllNoGray(2,1), ci_AllNoGray(2,2));
inv_Thshld_All2DDark = wblinv(.75, param_All2DDark(1), param_All2DDark(2));
lbinv_Thshld_All2DDark = wblinv(.75, ci_All2DDark(1,1), ci_All2DDark(1,2));
ubinv_Thshld_All2DDark = wblinv(.75, ci_All2DDark(2,1), ci_All2DDark(2,2));
inv_Thshld_All2DGray = wblinv(.75, param_All2DGray(1), param_All2DGray(2));
lbinv_Thshld_All2DGray = wblinv(.75, ci_All2DGray(1,1), ci_All2DGray(1,2));
ubinv_Thshld_All2DGray = wblinv(.75, ci_All2DGray(2,1), ci_All2DGray(2,2));
inv_Thshld_All2WDark = wblinv(.75, param_All2WDark(1), param_All2WDark(2));
lbinv_Thshld_All2WDark = wblinv(.75, ci_All2WDark(1,1), ci_All2WDark(1,2));
ubinv_Thshld_All2WDark = wblinv(.75, ci_All2WDark(2,1), ci_All2WDark(2,2));
inv_Thshld_All2WGray = wblinv(.75, param_All2WGray(1), param_All2WGray(2));
lbinv_Thshld_All2WGray = wblinv(.75, ci_All2WGray(1,1), ci_All2WGray(1,2));
ubinv_Thshld_All2WGray = wblinv(.75, ci_All2WGray(2,1), ci_All2WGray(2,2));
inv_Thshld_All2CDark = wblinv(.75, param_All2CDark(1), param_All2CDark(2));
lbinv_Thshld_All2CDark = wblinv(.75, ci_All2CDark(1,1), ci_All2CDark(1,2));
ubinv_Thshld_All2CDark = wblinv(.75, ci_All2CDark(2,1), ci_All2CDark(2,2));
inv_Thshld_All2CGray = wblinv(.75, param_All2CGray(1), param_All2CGray(2));
```

```
lbinv_Thshld_All2CGray = wblinv(.75, ci_All2CGray(1,1), ci_All2CGray(1,2));
ubinv_Thshld_All2CGray = wblinv(.75, ci_All2CGray(2,1), ci_All2CGray(2,2));
inv_Thshld_All3DDark = wblinv(.75, param_All3DDark(1), param_All3DDark(2));
lbinv_Thshld_All3DDark = wblinv(.75, ci_All3DDark(1,1), ci_All3DDark(1,2));
ubinv_Thshld_All3DDark = wblinv(.75, ci_All3DDark(2,1), ci_All3DDark(2,2));
inv_Thshld_All3DGray = wblinv(.75, param_All3DGray(1), param_All3DGray(2));
lbinv_Thshld_All3DGray = wblinv(.75, ci_All3DGray(1,1), ci_All3DGray(1,2));
ubinv_Thshld_All3DGray = wblinv(.75, ci_All3DGray(2,1), ci_All3DGray(2,2));
inv_Thshld_All3WDark = wblinv(.75, param_All3WDark(1), param_All3WDark(2));
lbinv_Thshld_All3WDark = wblinv(.75, ci_All3WDark(1,1), ci_All3WDark(1,2));
ubinv_Thshld_All3WDark = wblinv(.75, ci_All3WDark(2,1), ci_All3WDark(2,2));
inv_Thshld_All3WGray = wblinv(.75, param_All3WGray(1), param_All3WGray(2));
lbinv_Thshld_All3WGray = wblinv(.75, ci_All3WGray(1,1), ci_All3WGray(1,2));
ubinv_Thshld_All3WGray = wblinv(.75, ci_All3WGray(2,1), ci_All3WGray(2,2));
inv_Thshld_All3CDark = wblinv(.75, param_All3CDark(1), param_All3CDark(2));
lbinv_Thshld_All3CDark = wblinv(.75, ci_All3CDark(1,1), ci_All3CDark(1,2));
ubinv_Thshld_All3CDark = wblinv(.75, ci_All3CDark(2,1), ci_All3CDark(2,2));
inv_Thshld_All3CGray = wblinv(.75, param_All3CGray(1), param_All3CGray(2));
lbinv_Thshld_All3CGray = wblinv(.75, ci_All3CGray(1,1), ci_All3CGray(1,2));
ubinv_Thshld_All3CGray = wblinv(.75, ci_All3CGray(2,1), ci_All3CGray(2,2));
inv_Thshld_All4DDark = wblinv(.75, param_All4DDark(1), param_All4DDark(2));
lbinv_Thshld_All4DDark = wblinv(.75, ci_All4DDark(1,1), ci_All4DDark(1,2));
ubinv_Thshld_All4DDark = wblinv(.75, ci_All4DDark(2,1), ci_All4DDark(2,2));
inv_Thshld_All4DGray = wblinv(.75, param_All4DGray(1), param_All4DGray(2));
lbinv_Thshld_All4DGray = wblinv(.75, ci_All4DGray(1,1), ci_All4DGray(1,2));
ubinv_Thshld_All4DGray = wblinv(.75, ci_All4DGray(2,1), ci_All4DGray(2,2));
inv_Thshld_All4WDark = wblinv(.75, param_All4WDark(1), param_All4WDark(2));
lbinv_Thshld_All4WDark = wblinv(.75, ci_All4WDark(1,1), ci_All4WDark(1,2));
ubinv_Thshld_All4WDark = wblinv(.75, ci_All4WDark(2,1), ci_All4WDark(2,2));
inv_Thshld_All4WGray = wblinv(.75, param_All4WGray(1), param_All4WGray(2));
lbinv_Thshld_All4WGray = wblinv(.75, ci_All4WGray(1,1), ci_All4WGray(1,2));
ubinv_Thshld_All4WGray = wblinv(.75, ci_All4WGray(2,1), ci_All4WGray(2,2));
inv_Thshld_All4CDark = wblinv(.75, param_All4CDark(1), param_All4CDark(2));
lbinv_Thshld_All4CDark = wblinv(.75, ci_All4CDark(1,1), ci_All4CDark(1,2));
ubinv_Thshld_All4CDark = wblinv(.75, ci_All4CDark(2,1), ci_All4CDark(2,2));
inv_Thshld_All4CGray = wblinv(.75, param_All4CGray(1), param_All4CGray(2));
lbinv_Thshld_All4CGray = wblinv(.75, ci_All4CGray(1,1), ci_All4CGray(1,2));
ubinv_Thshld_All4CGray = wblinv(.75, ci_All4CGray(2,1), ci_All4CGray(2,2));

%% Calculate Contrast
Thshld_AllNoDark = 1 - inv_Thshld_AllNoDark;
lbThshld_AllNoDark = 1 - lbinv_Thshld_AllNoDark;
ubThshld_AllNoDark = 1 - ubinv_Thshld_AllNoDark;
Thshld_AllNoGray = 1 - inv_Thshld_AllNoGray;
lbThshld_AllNoGray = 1 - lbinv_Thshld_AllNoGray;
ubThshld_AllNoGray = 1 - ubinv_Thshld_AllNoGray;
Thshld_All2DDark = 1 - inv_Thshld_All2DDark;
lbThshld_All2DDark = 1 - lbinv_Thshld_All2DDark;
ubThshld_All2DDark = 1 - ubinv_Thshld_All2DDark;
Thshld_All2DGray = 1 - inv_Thshld_All2DGray;
lbThshld_All2DGray = 1 - lbinv_Thshld_All2DGray;
ubThshld_All2DGray = 1 - ubinv_Thshld_All2DGray;
Thshld_All2WDark = 1 - inv_Thshld_All2WDark;
lbThshld_All2WDark = 1 - lbinv_Thshld_All2WDark;
ubThshld_All2WDark = 1 - ubinv_Thshld_All2WDark;
```

```
Thshld_All2WGray = 1 - inv_Thshld_All2WGray;
lbThshld_All2WGray = 1 - lbinv_Thshld_All2WGray;
ubThshld_All2WGray = 1 - ubinv_Thshld_All2WGray;
Thshld_All2CDark = 1 - inv_Thshld_All2CDark;
lbThshld_All2CDark = 1 - lbinv_Thshld_All2CDark;
ubThshld_All2CDark = 1 - ubinv_Thshld_All2CDark;
Thshld_All2CGray = 1 - inv_Thshld_All2CGray;
lbThshld_All2CGray = 1 - lbinv_Thshld_All2CGray;
ubThshld_All2CGray = 1 - ubinv_Thshld_All2CGray;
Thshld_All3DDark = 1 - inv_Thshld_All3DDark;
lbThshld_All3DDark = 1 - lbinv_Thshld_All3DDark;
ubThshld_All3DDark = 1 - ubinv_Thshld_All3DDark;
Thshld_All3DGray = 1 - inv_Thshld_All3DGray;
lbThshld_All3DGray = 1 - lbinv_Thshld_All3DGray;
ubThshld_All3DGray = 1 - ubinv_Thshld_All3DGray;
Thshld_All3WDark = 1 - inv_Thshld_All3WDark;
lbThshld_All3WDark = 1 - lbinv_Thshld_All3WDark;
ubThshld_All3WDark = 1 - ubinv_Thshld_All3WDark;
Thshld_All3WGray = 1 - inv_Thshld_All3WGray;
lbThshld_All3WGray = 1 - lbinv_Thshld_All3WGray;
ubThshld_All3WGray = 1 - ubinv_Thshld_All3WGray;
Thshld_All3CDark = 1 - inv_Thshld_All3CDark;
lbThshld_All3CDark = 1 - lbinv_Thshld_All3CDark;
ubThshld_All3CDark = 1 - ubinv_Thshld_All3CDark;
Thshld_All3CGray = 1 - inv_Thshld_All3CGray;
lbThshld_All3CGray = 1 - lbinv_Thshld_All3CGray;
ubThshld_All3CGray = 1 - ubinv_Thshld_All3CGray;
Thshld_All4DDark = 1 - inv_Thshld_All4DDark;
lbThshld_All4DDark = 1 - lbinv_Thshld_All4DDark;
ubThshld_All4DDark = 1 - ubinv_Thshld_All4DDark;
Thshld_All4DGray = 1 - inv_Thshld_All4DGray;
lbThshld_All4DGray = 1 - lbinv_Thshld_All4DGray;
ubThshld_All4DGray = 1 - ubinv_Thshld_All4DGray;
Thshld_All4WDark = 1 - inv_Thshld_All4WDark;
lbThshld_All4WDark = 1 - lbinv_Thshld_All4WDark;
ubThshld_All4WDark = 1 - ubinv_Thshld_All4WDark;
Thshld_All4WGray = 1 - inv_Thshld_All4WGray;
lbThshld_All4WGray = 1 - lbinv_Thshld_All4WGray;
ubThshld_All4WGray = 1 - ubinv_Thshld_All4WGray;
Thshld_All4CDark = 1 - inv_Thshld_All4CDark;
lbThshld_All4CDark = 1 - lbinv_Thshld_All4CDark;
ubThshld_All4CDark = 1 - ubinv_Thshld_All4CDark;
Thshld_All4CGray = 1 - inv_Thshld_All4CGray;
lbThshld_All4CGray = 1 - lbinv_Thshld_All4CGray;
ubThshld_All4CGray = 1 - ubinv_Thshld_All4CGray;

%% Plot Thresholds
if plt2on
    xaxis = [0, 21.53, 32.29, 43.06];
    deskdark = [Thshld_AllNoDark, Thshld_All2DDark, Thshld_All3DDark, …
       Thshld_All4DDark];
    lbdeskdark = [lbThshld_AllNoDark, lbThshld_All2DDark, lbThshld_All3DDark, …
       lbThshld_All4DDark];
    ubdeskdark = [ubThshld_AllNoDark, ubThshld_All2DDark, ubThshld_All3DDark, …
       ubThshld_All4DDark];
```

```
walldark = [Thshld_AllNoDark, Thshld_All2WDark, Thshld_All3WDark, …
   Thshld_All4WDark];
lbwalldark = [lbThshld_AllNoDark, lbThshld_All2WDark, lbThshld_All3WDark, …
   lbThshld_All4WDark];
ubwalldark = [ubThshld_AllNoDark, ubThshld_All2WDark, ubThshld_All3WDark, …
   ubThshld_All4WDark];
clngdark = [Thshld_AllNoDark, Thshld_All2CDark, Thshld_All3CDark, …
   Thshld_All4CDark];
lbclngdark = [lbThshld_AllNoDark, lbThshld_All2CDark, lbThshld_All3CDark, …
   lbThshld_All4CDark];
ubclngdark = [ubThshld_AllNoDark, ubThshld_All2CDark, ubThshld_All3CDark, …
   ubThshld_All4CDark];
figure;
plot(xaxis, deskdark, 'rx-','LineWidth',1);
hold on;
plot(xaxis, walldark, 'bx-','LineWidth',1);
plot(xaxis, clngdark, 'kx-','LineWidth',1);
fill([xaxis fliplr(xaxis)],[ubdeskdark fliplr(lbdeskdark)],'r', …
   'linestyle','none');
fill([xaxis fliplr(xaxis)],[ubwalldark fliplr(lbwalldark)],'b', …
   'linestyle','none');
fill([xaxis fliplr(xaxis)],[ubclngdark fliplr(lbclngdark)],'k', …
   'linestyle','none');
alpha(0.25);
ylim([0 0.1])
yaxis = 0:.02:.1;
yticks(yaxis);
xticks(xaxis);
xlabel('Illumination Condition (lx)');
ylabel('Contrast Threshold');
legend('Desk Condition','Wall Condition','Ceiling Condition','Desk CI'...
    ,'Wall CI','Ceiling CI','Location','southwest');
title('All Participants Dark Background')

figure;
deskgray = [Thshld_AllNoGray, Thshld_All2DGray, Thshld_All3DGray, …
   Thshld_All4DGray];
lbdeskgray = [lbThshld_AllNoGray, lbThshld_All2DGray, lbThshld_All3DGray, …
   lbThshld_All4DGray];
ubdeskgray = [ubThshld_AllNoGray, ubThshld_All2DGray, ubThshld_All3DGray, …
   ubThshld_All4DGray];
wallgray = [Thshld_AllNoGray, Thshld_All2WGray, Thshld_All3WGray, …
   Thshld_All4WGray];
lbwallgray = [lbThshld_AllNoGray, lbThshld_All2WGray, lbThshld_All3WGray, …
   lbThshld_All4WGray];
ubwallgray = [ubThshld_AllNoGray, ubThshld_All2WGray, ubThshld_All3WGray, …
   ubThshld_All4WGray];
clnggray = [Thshld_AllNoGray, Thshld_All2CGray, Thshld_All3CGray, …
   Thshld_All4CGray];
lbclnggray = [lbThshld_AllNoGray, lbThshld_All2CGray, lbThshld_All3CGray, …
   lbThshld_All4CGray];
ubclnggray = [ubThshld_AllNoGray, ubThshld_All2CGray, ubThshld_All3CGray, …
   ubThshld_All4CGray];
plot(xaxis, deskgray, 'rx-','LineWidth',1);
hold on;
```

```
    plot(xaxis, wallgray, 'bx-','LineWidth',1);
    plot(xaxis, clnggray, 'kx-','LineWidth',1);
    fill([xaxis fliplr(xaxis)],[ubdeskgray fliplr(lbdeskgray)],'r', …
       'linestyle','none');
    fill([xaxis fliplr(xaxis)],[ubwallgray fliplr(lbwallgray)],'b', …
       'linestyle','none');
    fill([xaxis fliplr(xaxis)],[ubclnggray fliplr(lbclnggray)],'k', …
       'linestyle','none');
    alpha(0.25);
    ylim([0.002 0.017])
    yaxis = 0.002:.003:.017;
    ax = gca;
    ax.YRuler.Exponent = 0;
    yticks(yaxis);
    xticks(xaxis);
    xlabel('Illumination Condition (lx)');
    ylabel('Contrast Threshold');
    legend('Desk Condition','Wall Condition','Ceiling Condition','Desk CI'...
        ,'Wall CI','Ceiling CI','Location','southwest');
    title('All Participants Light Background')
end
```

All Participants, Grouped by Age and Background Shade

```
format compact;
clear variables;

%% Define Parameters
plt2on = true;
myfile = 'participantdata.xlsx';

%% Load Participant Data
VertIllum = xlsread(myfile, 'F2:F4801');
Background = xlsread(myfile, 'K2:K4801');
Accuracy = xlsread(myfile, 'P2:P4801');
ResponseTime = xlsread(myfile, 'Q2:Q4801');
Michelson = xlsread(myfile, 'Z2:Z4801');
RawMichelson = xlsread(myfile, 'AA2:AA4801');
Age = xlsread(myfile, 'AB2:AB4801');

%% Create Data Groups
% 30 and Over
OldNoDark = [1-Michelson(VertIllum==0 & Background==.035 & Age>29),Accuracy(VertIllum…
       ==0 & Background==.035 & Age>29)];
OldNoGray = [1-Michelson(VertIllum==0 & Background==.4 & Age>29),Accuracy(VertIllum…
==0 & Background==.4 & Age>29)];
Old2DDark = [1-Michelson(VertIllum==9.58 & Background==.035 & Age>29),Accuracy(…
       VertIllum==9.58 & Background==.035 & Age>29)];
Old2DGray = [1-Michelson(VertIllum==9.58 & Background==.4 & Age>29),Accuracy(…
       VertIllum==9.58 & Background==.4 & Age>29)];
Old2WDark = [1-Michelson(VertIllum==9.75 & Background==.035 & Age>29),Accuracy(…
       VertIllum==9.75 & Background==.035 & Age>29)];
Old2WGray = [1-Michelson(VertIllum==9.75 & Background==.4 & Age>29),Accuracy(…
       VertIllum==9.75 & Background==.4 & Age>29)];
Old2CDark = [1-Michelson(VertIllum==12.24 & Background==.035 & Age>29),Accuracy(…
        VertIllum==12.24 & Background==.035 & Age>29)];
Old2CGray = [1-Michelson(VertIllum==12.24 & Background==.4 & Age>29),Accuracy(...
        VertIllum==12.24 & Background==.4 & Age>29)];
Old3DDark = [1-Michelson(VertIllum==17.18 & Background==.035 & Age>29),Accuracy(...
        VertIllum==17.18 & Background==.035 & Age>29)];
Old3DGray = [1-Michelson(VertIllum==17.18 & Background==.4 & Age>29),Accuracy(...
        VertIllum==17.18 & Background==.4 & Age>29)];
Old3WDark = [1-Michelson(VertIllum==17.21 & Background==.035 & Age>29),Accuracy(...
        VertIllum==17.21 & Background==.035 & Age>29)];
Old3WGray = [1-Michelson(VertIllum==17.21 & Background==.4 & Age>29),Accuracy(...
        VertIllum==17.21 & Background==.4 & Age>29)];
Old3CDark = [1-Michelson(VertIllum==22.11 & Background==.035 & Age>29),Accuracy(...
        VertIllum==22.11 & Background==.035 & Age>29)];
Old3CGray = [1-Michelson(VertIllum==22.11 & Background==.4 & Age>29),Accuracy(...
        VertIllum==22.11 & Background==.4 & Age>29)];
Old4DDark = [1-Michelson(VertIllum==23.67 & Background==.035 & Age>29),Accuracy(...
        VertIllum==23.67 & Background==.035 & Age>29)];
Old4DGray = [1-Michelson(VertIllum==23.67 & Background==.4 & Age>29),Accuracy(...
        VertIllum==23.67 & Background==.4 & Age>29)];
Old4WDark = [1-Michelson(VertIllum==23.93 & Background==.035 & Age>29),Accuracy(...
        VertIllum==23.93 & Background==.035 & Age>29)];
Old4WGray = [1-Michelson(VertIllum==23.93 & Background==.4 & Age>29),Accuracy(...
        VertIllum==23.93 & Background==.4 & Age>29)];
Old4CDark = [1-Michelson(VertIllum==28.9 & Background==.035 & Age>29),Accuracy(...
        VertIllum==28.9 & Background==.035 & Age>29)];
Old4CGray = [1-Michelson(VertIllum==28.9 & Background==.4 & Age>29),Accuracy(...
        VertIllum==28.9 & Background==.4 & Age>29)];

% Under 30
YngNoDark = [1-Michelson(VertIllum==0 & Background==.035 & Age<29),Accuracy(...
        VertIllum==0 & Background==.035 & Age<29)];
YngNoGray = [1-Michelson(VertIllum==0 & Background==.4 & Age<29),Accuracy(...
        VertIllum==0 & Background==.4 & Age<29)];
Yng2DDark = [1-Michelson(VertIllum==9.58 & Background==.035 & Age<29),Accuracy(...
        VertIllum==9.58 & Background==.035 & Age<29)];
Yng2DGray = [1-Michelson(VertIllum==9.58 & Background==.4 & Age<29),Accuracy(...
        VertIllum==9.58 & Background==.4 & Age<29)];
Yng2WDark = [1-Michelson(VertIllum==9.75 & Background==.035 & Age<29),Accuracy(...
        VertIllum==9.75 & Background==.035 & Age<29)];
Yng2WGray = [1-Michelson(VertIllum==9.75 & Background==.4 & Age<29),Accuracy(...
        VertIllum==9.75 & Background==.4 & Age<29)];
Yng2CDark = [1-Michelson(VertIllum==12.24 & Background==.035 & Age<29),Accuracy(...
        VertIllum==12.24 & Background==.035 & Age<29)];
Yng2CGray = [1-Michelson(VertIllum==12.24 & Background==.4 & Age<29),Accuracy(...
        VertIllum==12.24 & Background==.4 & Age<29)];
Yng3DDark = [1-Michelson(VertIllum==17.18 & Background==.035 & Age<29),Accuracy(...
        VertIllum==17.18 & Background==.035 & Age<29)];
Yng3DGray = [1-Michelson(VertIllum==17.18 & Background==.4 & Age<29),Accuracy(...
        VertIllum==17.18 & Background==.4 & Age<29)];
Yng3WDark = [1-Michelson(VertIllum==17.21 & Background==.035 & Age<29),Accuracy(...
        VertIllum==17.21 & Background==.035 & Age<29)];
Yng3WGray = [1-Michelson(VertIllum==17.21 & Background==.4 & Age<29),Accuracy(...
        VertIllum==17.21 & Background==.4 & Age<29)];
Yng3CDark = [1-Michelson(VertIllum==22.11 & Background==.035 & Age<29),Accuracy(...
        VertIllum==22.11 & Background==.035 & Age<29)];
```

```
Yng3CGray = [1-Michelson(VertIllum==22.11 & Background==.4 & Age<29),Accuracy(…
       VertIllum==22.11 & Background==.4 & Age<29)];
Yng4DDark = [1-Michelson(VertIllum==23.67 & Background==.035 & Age<29),Accuracy(…
       VertIllum==23.67 & Background==.035 & Age<29)];
Yng4DGray = [1-Michelson(VertIllum==23.67 & Background==.4 & Age<29),Accuracy(…
       VertIllum==23.67 & Background==.4 & Age<29)];
Yng4WDark = [1-Michelson(VertIllum==23.93 & Background==.035 & Age<29),Accuracy(…
       VertIllum==23.93 & Background==.035 & Age<29)];
Yng4WGray = [1-Michelson(VertIllum==23.93 & Background==.4 & Age<29),Accuracy(…
       VertIllum==23.93 & Background==.4 & Age<29)];
Yng4CDark = [1-Michelson(VertIllum==28.9 & Background==.035 & Age<29),Accuracy(…
       VertIllum==28.9 & Background==.035 & Age<29)];
Yng4CGray = [1-Michelson(VertIllum==28.9 & Background==.4 & Age<29),Accuracy(…
       VertIllum==28.9 & Background==.4 & Age<29)];

%% Calculate Survior Functions
[param_OldNoDark, ci_OldNoDark] = wblfit(OldNoDark(:,1),0.05,OldNoDark(:,2));
[param_OldNoGray, ci_OldNoGray] = wblfit(OldNoGray(:,1),0.05,OldNoGray(:,2));
[param_Old2DDark, ci_Old2DDark] = wblfit(Old2DDark(:,1),0.05,Old2DDark(:,2));
[param_Old2DGray, ci_Old2DGray] = wblfit(Old2DGray(:,1),0.05,Old2DGray(:,2));
[param_Old2WDark, ci_Old2WDark] = wblfit(Old2WDark(:,1),0.05,Old2WDark(:,2));
[param_Old2WGray, ci_Old2WGray] = wblfit(Old2WGray(:,1),0.05,Old2WGray(:,2));
[param_Old2CDark, ci_Old2CDark] = wblfit(Old2CDark(:,1),0.05,Old2CDark(:,2));
[param_Old2CGray, ci_Old2CGray] = wblfit(Old2CGray(:,1),0.05,Old2CGray(:,2));
[param_Old3DDark, ci_Old3DDark] = wblfit(Old3DDark(:,1),0.05,Old3DDark(:,2));
[param_Old3DGray, ci_Old3DGray] = wblfit(Old3DGray(:,1),0.05,Old3DGray(:,2));
[param_Old3WDark, ci_Old3WDark] = wblfit(Old3WDark(:,1),0.05,Old3WDark(:,2));
[param_Old3WGray, ci_Old3WGray] = wblfit(Old3WGray(:,1),0.05,Old3WGray(:,2));
[param_Old3CDark, ci_Old3CDark] = wblfit(Old3CDark(:,1),0.05,Old3CDark(:,2));
[param_Old3CGray, ci_Old3CGray] = wblfit(Old3CGray(:,1),0.05,Old3CGray(:,2));
[param_Old4DDark, ci_Old4DDark] = wblfit(Old4DDark(:,1),0.05,Old4DDark(:,2));
[param_Old4DGray, ci_Old4DGray] = wblfit(Old4DGray(:,1),0.05,Old4DGray(:,2));
[param_Old4WDark, ci_Old4WDark] = wblfit(Old4WDark(:,1),0.05,Old4WDark(:,2));
[param_Old4WGray, ci_Old4WGray] = wblfit(Old4WGray(:,1),0.05,Old4WGray(:,2));
[param_Old4CDark, ci_Old4CDark] = wblfit(Old4CDark(:,1),0.05,Old4CDark(:,2));
[param_Old4CGray, ci_Old4CGray] = wblfit(Old4CGray(:,1),0.05,Old4CGray(:,2));

[param_YngNoDark, ci_YngNoDark] = wblfit(YngNoDark(:,1),0.05,YngNoDark(:,2));
[param_YngNoGray, ci_YngNoGray] = wblfit(YngNoGray(:,1),0.05,YngNoGray(:,2));
[param_Yng2DDark, ci_Yng2DDark] = wblfit(Yng2DDark(:,1),0.05,Yng2DDark(:,2));
[param_Yng2DGray, ci_Yng2DGray] = wblfit(Yng2DGray(:,1),0.05,Yng2DGray(:,2));
[param_Yng2WDark, ci_Yng2WDark] = wblfit(Yng2WDark(:,1),0.05,Yng2WDark(:,2));
[param_Yng2WGray, ci_Yng2WGray] = wblfit(Yng2WGray(:,1),0.05,Yng2WGray(:,2));
[param_Yng2CDark, ci_Yng2CDark] = wblfit(Yng2CDark(:,1),0.05,Yng2CDark(:,2));
[param_Yng2CGray, ci_Yng2CGray] = wblfit(Yng2CGray(:,1),0.05,Yng2CGray(:,2));
[param_Yng3DDark, ci_Yng3DDark] = wblfit(Yng3DDark(:,1),0.05,Yng3DDark(:,2));
[param_Yng3DGray, ci_Yng3DGray] = wblfit(Yng3DGray(:,1),0.05,Yng3DGray(:,2));
[param_Yng3WDark, ci_Yng3WDark] = wblfit(Yng3WDark(:,1),0.05,Yng3WDark(:,2));
[param_Yng3WGray, ci_Yng3WGray] = wblfit(Yng3WGray(:,1),0.05,Yng3WGray(:,2));
[param_Yng3CDark, ci_Yng3CDark] = wblfit(Yng3CDark(:,1),0.05,Yng3CDark(:,2));
[param_Yng3CGray, ci_Yng3CGray] = wblfit(Yng3CGray(:,1),0.05,Yng3CGray(:,2));
[param_Yng4DDark, ci_Yng4DDark] = wblfit(Yng4DDark(:,1),0.05,Yng4DDark(:,2));
[param_Yng4DGray, ci_Yng4DGray] = wblfit(Yng4DGray(:,1),0.05,Yng4DGray(:,2));
[param_Yng4WDark, ci_Yng4WDark] = wblfit(Yng4WDark(:,1),0.05,Yng4WDark(:,2));
[param_Yng4WGray, ci_Yng4WGray] = wblfit(Yng4WGray(:,1),0.05,Yng4WGray(:,2));
```

```
[param_Yng4CDark, ci_Yng4CDark] = wblfit(Yng4CDark(:,1),0.05,Yng4CDark(:,2));
[param_Yng4CGray, ci_Yng4CGray] = wblfit(Yng4CGray(:,1),0.05,Yng4CGray(:,2));

%% Compute Threshold
inv_Thshld_OldNoDark = wblinv(.75, param_OldNoDark(1), param_OldNoDark(2));
inv_Thshld_OldNoGray = wblinv(.75, param_OldNoGray(1), param_OldNoGray(2));
inv_Thshld_Old2DDark = wblinv(.75, param_Old2DDark(1), param_Old2DDark(2));
inv_Thshld_Old2DGray = wblinv(.75, param_Old2DGray(1), param_Old2DGray(2));
inv_Thshld_Old2WDark = wblinv(.75, param_Old2WDark(1), param_Old2WDark(2));
inv_Thshld_Old2WGray = wblinv(.75, param_Old2WGray(1), param_Old2WGray(2));
inv_Thshld_Old2CDark = wblinv(.75, param_Old2CDark(1), param_Old2CDark(2));
inv_Thshld_Old2CGray = wblinv(.75, param_Old2CGray(1), param_Old2CGray(2));
inv_Thshld_Old3DDark = wblinv(.75, param_Old3DDark(1), param_Old3DDark(2));
inv_Thshld_Old3DGray = wblinv(.75, param_Old3DGray(1), param_Old3DGray(2));
inv_Thshld_Old3WDark = wblinv(.75, param_Old3WDark(1), param_Old3WDark(2));
inv_Thshld_Old3WGray = wblinv(.75, param_Old3WGray(1), param_Old3WGray(2));
inv_Thshld_Old3CDark = wblinv(.75, param_Old3CDark(1), param_Old3CDark(2));
inv_Thshld_Old3CGray = wblinv(.75, param_Old3CGray(1), param_Old3CGray(2));
inv_Thshld_Old4DDark = wblinv(.75, param_Old4DDark(1), param_Old4DDark(2));
inv_Thshld_Old4DGray = wblinv(.75, param_Old4DGray(1), param_Old4DGray(2));
inv_Thshld_Old4WDark = wblinv(.75, param_Old4WDark(1), param_Old4WDark(2));
inv_Thshld_Old4WGray = wblinv(.75, param_Old4WGray(1), param_Old4WGray(2));
inv_Thshld_Old4CDark = wblinv(.75, param_Old4CDark(1), param_Old4CDark(2));
inv_Thshld_Old4CGray = wblinv(.75, param_Old4CGray(1), param_Old4CGray(2));

lbinv_Thshld_OldNoDark = wblinv(.75, ci_OldNoDark(1,1), ci_OldNoDark(1,2));
lbinv_Thshld_OldNoGray = wblinv(.75, ci_OldNoGray(1,1), ci_OldNoGray(1,2));
lbinv_Thshld_Old2DDark = wblinv(.75, ci_Old2DDark(1,1), ci_Old2DDark(1,2));
lbinv_Thshld_Old2DGray = wblinv(.75, ci_Old2DGray(1,1), ci_Old2DGray(1,2));
lbinv_Thshld_Old2WDark = wblinv(.75, ci_Old2WDark(1,1), ci_Old2WDark(1,2));
lbinv_Thshld_Old2WGray = wblinv(.75, ci_Old2WGray(1,1), ci_Old2WGray(1,2));
lbinv_Thshld_Old2CDark = wblinv(.75, ci_Old2CDark(1,1), ci_Old2CDark(1,2));
lbinv_Thshld_Old2CGray = wblinv(.75, ci_Old2CGray(1,1), ci_Old2CGray(1,2));
lbinv_Thshld_Old3DDark = wblinv(.75, ci_Old3DDark(1,1), ci_Old3DDark(1,2));
lbinv_Thshld_Old3DGray = wblinv(.75, ci_Old3DGray(1,1), ci_Old3DGray(1,2));
lbinv_Thshld_Old3WDark = wblinv(.75, ci_Old3WDark(1,1), ci_Old3WDark(1,2));
lbinv_Thshld_Old3WGray = wblinv(.75, ci_Old3WGray(1,1), ci_Old3WGray(1,2));
lbinv_Thshld_Old3CDark = wblinv(.75, ci_Old3CDark(1,1), ci_Old3CDark(1,2));
lbinv_Thshld_Old3CGray = wblinv(.75, ci_Old3CGray(1,1), ci_Old3CGray(1,2));
lbinv_Thshld_Old4DDark = wblinv(.75, ci_Old4DDark(1,1), ci_Old4DDark(1,2));
lbinv_Thshld_Old4DGray = wblinv(.75, ci_Old4DGray(1,1), ci_Old4DGray(1,2));
lbinv_Thshld_Old4WDark = wblinv(.75, ci_Old4WDark(1,1), ci_Old4WDark(1,2));
lbinv_Thshld_Old4WGray = wblinv(.75, ci_Old4WGray(1,1), ci_Old4WGray(1,2));
lbinv_Thshld_Old4CDark = wblinv(.75, ci_Old4CDark(1,1), ci_Old4CDark(1,2));
lbinv_Thshld_Old4CGray = wblinv(.75, ci_Old4CGray(1,1), ci_Old4CGray(1,2));

ubinv_Thshld_OldNoDark = wblinv(.75, ci_OldNoDark(2,1), ci_OldNoDark(2,2));
ubinv_Thshld_OldNoGray = wblinv(.75, ci_OldNoGray(2,1), ci_OldNoGray(2,2));
ubinv_Thshld_Old2DDark = wblinv(.75, ci_Old2DDark(2,1), ci_Old2DDark(2,2));
ubinv_Thshld_Old2DGray = wblinv(.75, ci_Old2DGray(2,1), ci_Old2DGray(2,2));
ubinv_Thshld_Old2WDark = wblinv(.75, ci_Old2WDark(2,1), ci_Old2WDark(2,2));
ubinv_Thshld_Old2WGray = wblinv(.75, ci_Old2WGray(2,1), ci_Old2WGray(2,2));
ubinv_Thshld_Old2CDark = wblinv(.75, ci_Old2CDark(2,1), ci_Old2CDark(2,2));
ubinv_Thshld_Old2CGray = wblinv(.75, ci_Old2CGray(2,1), ci_Old2CGray(2,2));
ubinv_Thshld_Old3DDark = wblinv(.75, ci_Old3DDark(2,1), ci_Old3DDark(2,2));
```

```
ubinv_Thshld_Old3DGray = wblinv(.75, ci_Old3DGray(2,1), ci_Old3DGray(2,2));
ubinv_Thshld_Old3WDark = wblinv(.75, ci_Old3WDark(2,1), ci_Old3WDark(2,2));
ubinv_Thshld_Old3WGray = wblinv(.75, ci_Old3WGray(2,1), ci_Old3WGray(2,2));
ubinv_Thshld_Old3CDark = wblinv(.75, ci_Old3CDark(2,1), ci_Old3CDark(2,2));
ubinv_Thshld_Old3CGray = wblinv(.75, ci_Old3CGray(2,1), ci_Old3CGray(2,2));
ubinv_Thshld_Old4DDark = wblinv(.75, ci_Old4DDark(2,1), ci_Old4DDark(2,2));
ubinv_Thshld_Old4DGray = wblinv(.75, ci_Old4DGray(2,1), ci_Old4DGray(2,2));
ubinv_Thshld_Old4WDark = wblinv(.75, ci_Old4WDark(2,1), ci_Old4WDark(2,2));
ubinv_Thshld_Old4WGray = wblinv(.75, ci_Old4WGray(2,1), ci_Old4WGray(2,2));
ubinv_Thshld_Old4CDark = wblinv(.75, ci_Old4CDark(2,1), ci_Old4CDark(2,2));
ubinv_Thshld_Old4CGray = wblinv(.75, ci_Old4CGray(2,1), ci_Old4CGray(2,2));

inv_Thshld_YngNoDark = wblinv(.75, param_YngNoDark(1), param_YngNoDark(2));
inv_Thshld_YngNoGray = wblinv(.75, param_YngNoGray(1), param_YngNoGray(2));
inv_Thshld_Yng2DDark = wblinv(.75, param_Yng2DDark(1), param_Yng2DDark(2));
inv_Thshld_Yng2DGray = wblinv(.75, param_Yng2DGray(1), param_Yng2DGray(2));
inv_Thshld_Yng2WDark = wblinv(.75, param_Yng2WDark(1), param_Yng2WDark(2));
inv_Thshld_Yng2WGray = wblinv(.75, param_Yng2WGray(1), param_Yng2WGray(2));
inv_Thshld_Yng2CDark = wblinv(.75, param_Yng2CDark(1), param_Yng2CDark(2));
inv_Thshld_Yng2CGray = wblinv(.75, param_Yng2CGray(1), param_Yng2CGray(2));
inv_Thshld_Yng3DDark = wblinv(.75, param_Yng3DDark(1), param_Yng3DDark(2));
inv_Thshld_Yng3DGray = wblinv(.75, param_Yng3DGray(1), param_Yng3DGray(2));
inv_Thshld_Yng3WDark = wblinv(.75, param_Yng3WDark(1), param_Yng3WDark(2));
inv_Thshld_Yng3WGray = wblinv(.75, param_Yng3WGray(1), param_Yng3WGray(2));
inv_Thshld_Yng3CDark = wblinv(.75, param_Yng3CDark(1), param_Yng3CDark(2));
inv_Thshld_Yng3CGray = wblinv(.75, param_Yng3CGray(1), param_Yng3CGray(2));
inv_Thshld_Yng4DDark = wblinv(.75, param_Yng4DDark(1), param_Yng4DDark(2));
inv_Thshld_Yng4DGray = wblinv(.75, param_Yng4DGray(1), param_Yng4DGray(2));
inv_Thshld_Yng4WDark = wblinv(.75, param_Yng4WDark(1), param_Yng4WDark(2));
inv_Thshld_Yng4WGray = wblinv(.75, param_Yng4WGray(1), param_Yng4WGray(2));
inv_Thshld_Yng4CDark = wblinv(.75, param_Yng4CDark(1), param_Yng4CDark(2));
inv_Thshld_Yng4CGray = wblinv(.75, param_Yng4CGray(1), param_Yng4CGray(2));

lbinv_Thshld_YngNoDark = wblinv(.75, ci_YngNoDark(1,1), ci_YngNoDark(1,2));
lbinv_Thshld_YngNoGray = wblinv(.75, ci_YngNoGray(1,1), ci_YngNoGray(1,2));
lbinv_Thshld_Yng2DDark = wblinv(.75, ci_Yng2DDark(1,1), ci_Yng2DDark(1,2));
lbinv_Thshld_Yng2DGray = wblinv(.75, ci_Yng2DGray(1,1), ci_Yng2DGray(1,2));
lbinv_Thshld_Yng2WDark = wblinv(.75, ci_Yng2WDark(1,1), ci_Yng2WDark(1,2));
lbinv_Thshld_Yng2WGray = wblinv(.75, ci_Yng2WGray(1,1), ci_Yng2WGray(1,2));
lbinv_Thshld_Yng2CDark = wblinv(.75, ci_Yng2CDark(1,1), ci_Yng2CDark(1,2));
lbinv_Thshld_Yng2CGray = wblinv(.75, ci_Yng2CGray(1,1), ci_Yng2CGray(1,2));
lbinv_Thshld_Yng3DDark = wblinv(.75, ci_Yng3DDark(1,1), ci_Yng3DDark(1,2));
lbinv_Thshld_Yng3DGray = wblinv(.75, ci_Yng3DGray(1,1), ci_Yng3DGray(1,2));
lbinv_Thshld_Yng3WDark = wblinv(.75, ci_Yng3WDark(1,1), ci_Yng3WDark(1,2));
lbinv_Thshld_Yng3WGray = wblinv(.75, ci_Yng3WGray(1,1), ci_Yng3WGray(1,2));
lbinv_Thshld_Yng3CDark = wblinv(.75, ci_Yng3CDark(1,1), ci_Yng3CDark(1,2));
lbinv_Thshld_Yng3CGray = wblinv(.75, ci_Yng3CGray(1,1), ci_Yng3CGray(1,2));
lbinv_Thshld_Yng4DDark = wblinv(.75, ci_Yng4DDark(1,1), ci_Yng4DDark(1,2));
lbinv_Thshld_Yng4DGray = wblinv(.75, ci_Yng4DGray(1,1), ci_Yng4DGray(1,2));
lbinv_Thshld_Yng4WDark = wblinv(.75, ci_Yng4WDark(1,1), ci_Yng4WDark(1,2));
lbinv_Thshld_Yng4WGray = wblinv(.75, ci_Yng4WGray(1,1), ci_Yng4WGray(1,2));
lbinv_Thshld_Yng4CDark = wblinv(.75, ci_Yng4CDark(1,1), ci_Yng4CDark(1,2));
lbinv_Thshld_Yng4CGray = wblinv(.75, ci_Yng4CGray(1,1), ci_Yng4CGray(1,2));

ubinv_Thshld_YngNoDark = wblinv(.75, ci_YngNoDark(2,1), ci_YngNoDark(2,2));
```

```
ubinv_Thshld_YngNoGray = wblinv(.75, ci_YngNoGray(2,1), ci_YngNoGray(2,2));
ubinv_Thshld_Yng2DDark = wblinv(.75, ci_Yng2DDark(2,1), ci_Yng2DDark(2,2));
ubinv_Thshld_Yng2DGray = wblinv(.75, ci_Yng2DGray(2,1), ci_Yng2DGray(2,2));
ubinv_Thshld_Yng2WDark = wblinv(.75, ci_Yng2WDark(2,1), ci_Yng2WDark(2,2));
ubinv_Thshld_Yng2WGray = wblinv(.75, ci_Yng2WGray(2,1), ci_Yng2WGray(2,2));
ubinv_Thshld_Yng2CDark = wblinv(.75, ci_Yng2CDark(2,1), ci_Yng2CDark(2,2));
ubinv_Thshld_Yng2CGray = wblinv(.75, ci_Yng2CGray(2,1), ci_Yng2CGray(2,2));
ubinv_Thshld_Yng3DDark = wblinv(.75, ci_Yng3DDark(2,1), ci_Yng3DDark(2,2));
ubinv_Thshld_Yng3DGray = wblinv(.75, ci_Yng3DGray(2,1), ci_Yng3DGray(2,2));
ubinv_Thshld_Yng3WDark = wblinv(.75, ci_Yng3WDark(2,1), ci_Yng3WDark(2,2));
ubinv_Thshld_Yng3WGray = wblinv(.75, ci_Yng3WGray(2,1), ci_Yng3WGray(2,2));
ubinv_Thshld_Yng3CDark = wblinv(.75, ci_Yng3CDark(2,1), ci_Yng3CDark(2,2));
ubinv_Thshld_Yng3CGray = wblinv(.75, ci_Yng3CGray(2,1), ci_Yng3CGray(2,2));
ubinv_Thshld_Yng4DDark = wblinv(.75, ci_Yng4DDark(2,1), ci_Yng4DDark(2,2));
ubinv_Thshld_Yng4DGray = wblinv(.75, ci_Yng4DGray(2,1), ci_Yng4DGray(2,2));
ubinv_Thshld_Yng4WDark = wblinv(.75, ci_Yng4WDark(2,1), ci_Yng4WDark(2,2));
ubinv_Thshld_Yng4WGray = wblinv(.75, ci_Yng4WGray(2,1), ci_Yng4WGray(2,2));
ubinv_Thshld_Yng4CDark = wblinv(.75, ci_Yng4CDark(2,1), ci_Yng4CDark(2,2));
ubinv_Thshld_Yng4CGray = wblinv(.75, ci_Yng4CGray(2,1), ci_Yng4CGray(2,2));

%% Calculate Contrast
Thshld_OldNoDark = 1 - inv_Thshld_OldNoDark;
Thshld_OldNoGray = 1 - inv_Thshld_OldNoGray;
Thshld_Old2DDark = 1 - inv_Thshld_Old2DDark;
Thshld_Old2DGray = 1 - inv_Thshld_Old2DGray;
Thshld_Old2WDark = 1 - inv_Thshld_Old2WDark;
Thshld_Old2WGray = 1 - inv_Thshld_Old2WGray;
Thshld_Old2CDark = 1 - inv_Thshld_Old2CDark;
Thshld_Old2CGray = 1 - inv_Thshld_Old2CGray;
Thshld_Old3DDark = 1 - inv_Thshld_Old3DDark;
Thshld_Old3DGray = 1 - inv_Thshld_Old3DGray;
Thshld_Old3WDark = 1 - inv_Thshld_Old3WDark;
Thshld_Old3WGray = 1 - inv_Thshld_Old3WGray;
Thshld_Old3CDark = 1 - inv_Thshld_Old3CDark;
Thshld_Old3CGray = 1 - inv_Thshld_Old3CGray;
Thshld_Old4DDark = 1 - inv_Thshld_Old4DDark;
Thshld_Old4DGray = 1 - inv_Thshld_Old4DGray;
Thshld_Old4WDark = 1 - inv_Thshld_Old4WDark;
Thshld_Old4WGray = 1 - inv_Thshld_Old4WGray;
Thshld_Old4CDark = 1 - inv_Thshld_Old4CDark;
Thshld_Old4CGray = 1 - inv_Thshld_Old4CGray;

lbThshld_OldNoDark = 1 - lbinv_Thshld_OldNoDark;
lbThshld_OldNoGray = 1 - lbinv_Thshld_OldNoGray;
lbThshld_Old2DDark = 1 - lbinv_Thshld_Old2DDark;
lbThshld_Old2DGray = 1 - lbinv_Thshld_Old2DGray;
lbThshld_Old2WDark = 1 - lbinv_Thshld_Old2WDark;
lbThshld_Old2WGray = 1 - lbinv_Thshld_Old2WGray;
lbThshld_Old2CDark = 1 - lbinv_Thshld_Old2CDark;
lbThshld_Old2CGray = 1 - lbinv_Thshld_Old2CGray;
lbThshld_Old3DDark = 1 - lbinv_Thshld_Old3DDark;
lbThshld_Old3DGray = 1 - lbinv_Thshld_Old3DGray;
lbThshld_Old3WDark = 1 - lbinv_Thshld_Old3WDark;
lbThshld_Old3WGray = 1 - lbinv_Thshld_Old3WGray;
lbThshld_Old3CDark = 1 - lbinv_Thshld_Old3CDark;
```

```
lbThshld_Old3CGray = 1 - lbinv_Thshld_Old3CGray;
lbThshld_Old4DDark = 1 - lbinv_Thshld_Old4DDark;
lbThshld_Old4DGray = 1 - lbinv_Thshld_Old4DGray;
lbThshld_Old4WDark = 1 - lbinv_Thshld_Old4WDark;
lbThshld_Old4WGray = 1 - lbinv_Thshld_Old4WGray;
lbThshld_Old4CDark = 1 - lbinv_Thshld_Old4CDark;
lbThshld_Old4CGray = 1 - lbinv_Thshld_Old4CGray;

ubThshld_OldNoDark = 1 - ubinv_Thshld_OldNoDark;
ubThshld_OldNoGray = 1 - ubinv_Thshld_OldNoGray;
ubThshld_Old2DDark = 1 - ubinv_Thshld_Old2DDark;
ubThshld_Old2DGray = 1 - ubinv_Thshld_Old2DGray;
ubThshld_Old2WDark = 1 - ubinv_Thshld_Old2WDark;
ubThshld_Old2WGray = 1 - ubinv_Thshld_Old2WGray;
ubThshld_Old2CDark = 1 - ubinv_Thshld_Old2CDark;
ubThshld_Old2CGray = 1 - ubinv_Thshld_Old2CGray;
ubThshld_Old3DDark = 1 - ubinv_Thshld_Old3DDark;
ubThshld_Old3DGray = 1 - ubinv_Thshld_Old3DGray;
ubThshld_Old3WDark = 1 - ubinv_Thshld_Old3WDark;
ubThshld_Old3WGray = 1 - ubinv_Thshld_Old3WGray;
ubThshld_Old3CDark = 1 - ubinv_Thshld_Old3CDark;
ubThshld_Old3CGray = 1 - ubinv_Thshld_Old3CGray;
ubThshld_Old4DDark = 1 - ubinv_Thshld_Old4DDark;
ubThshld_Old4DGray = 1 - ubinv_Thshld_Old4DGray;
ubThshld_Old4WDark = 1 - ubinv_Thshld_Old4WDark;
ubThshld_Old4WGray = 1 - ubinv_Thshld_Old4WGray;
ubThshld_Old4CDark = 1 - ubinv_Thshld_Old4CDark;
ubThshld_Old4CGray = 1 - ubinv_Thshld_Old4CGray;

Thshld_YngNoDark = 1 - inv_Thshld_YngNoDark;
Thshld_YngNoGray = 1 - inv_Thshld_YngNoGray;
Thshld_Yng2DDark = 1 - inv_Thshld_Yng2DDark;
Thshld_Yng2DGray = 1 - inv_Thshld_Yng2DGray;
Thshld_Yng2WDark = 1 - inv_Thshld_Yng2WDark;
Thshld_Yng2WGray = 1 - inv_Thshld_Yng2WGray;
Thshld_Yng2CDark = 1 - inv_Thshld_Yng2CDark;
Thshld_Yng2CGray = 1 - inv_Thshld_Yng2CGray;
Thshld_Yng3DDark = 1 - inv_Thshld_Yng3DDark;
Thshld_Yng3DGray = 1 - inv_Thshld_Yng3DGray;
Thshld_Yng3WDark = 1 - inv_Thshld_Yng3WDark;
Thshld_Yng3WGray = 1 - inv_Thshld_Yng3WGray;
Thshld_Yng3CDark = 1 - inv_Thshld_Yng3CDark;
Thshld_Yng3CGray = 1 - inv_Thshld_Yng3CGray;
Thshld_Yng4DDark = 1 - inv_Thshld_Yng4DDark;
Thshld_Yng4DGray = 1 - inv_Thshld_Yng4DGray;
Thshld_Yng4WDark = 1 - inv_Thshld_Yng4WDark;
Thshld_Yng4WGray = 1 - inv_Thshld_Yng4WGray;
Thshld_Yng4CDark = 1 - inv_Thshld_Yng4CDark;
Thshld_Yng4CGray = 1 - inv_Thshld_Yng4CGray;

lbThshld_YngNoDark = 1 - lbinv_Thshld_YngNoDark;
lbThshld_YngNoGray = 1 - lbinv_Thshld_YngNoGray;
lbThshld_Yng2DDark = 1 - lbinv_Thshld_Yng2DDark;
lbThshld_Yng2DGray = 1 - lbinv_Thshld_Yng2DGray;
lbThshld_Yng2WDark = 1 - lbinv_Thshld_Yng2WDark;
```

```
lbThshld_Yng2WGray = 1 - lbinv_Thshld_Yng2WGray;
lbThshld_Yng2CDark = 1 - lbinv_Thshld_Yng2CDark;
lbThshld_Yng2CGray = 1 - lbinv_Thshld_Yng2CGray;
lbThshld_Yng3DDark = 1 - lbinv_Thshld_Yng3DDark;
lbThshld_Yng3DGray = 1 - lbinv_Thshld_Yng3DGray;
lbThshld_Yng3WDark = 1 - lbinv_Thshld_Yng3WDark;
lbThshld_Yng3WGray = 1 - lbinv_Thshld_Yng3WGray;
lbThshld_Yng3CDark = 1 - lbinv_Thshld_Yng3CDark;
lbThshld_Yng3CGray = 1 - lbinv_Thshld_Yng3CGray;
lbThshld_Yng4DDark = 1 - lbinv_Thshld_Yng4DDark;
lbThshld_Yng4DGray = 1 - lbinv_Thshld_Yng4DGray;
lbThshld_Yng4WDark = 1 - lbinv_Thshld_Yng4WDark;
lbThshld_Yng4WGray = 1 - lbinv_Thshld_Yng4WGray;
lbThshld_Yng4CDark = 1 - lbinv_Thshld_Yng4CDark;
lbThshld_Yng4CGray = 1 - lbinv_Thshld_Yng4CGray;

ubThshld_YngNoDark = 1 - ubinv_Thshld_YngNoDark;
ubThshld_YngNoGray = 1 - ubinv_Thshld_YngNoGray;
ubThshld_Yng2DDark = 1 - ubinv_Thshld_Yng2DDark;
ubThshld_Yng2DGray = 1 - ubinv_Thshld_Yng2DGray;
ubThshld_Yng2WDark = 1 - ubinv_Thshld_Yng2WDark;
ubThshld_Yng2WGray = 1 - ubinv_Thshld_Yng2WGray;
ubThshld_Yng2CDark = 1 - ubinv_Thshld_Yng2CDark;
ubThshld_Yng2CGray = 1 - ubinv_Thshld_Yng2CGray;
ubThshld_Yng3DDark = 1 - ubinv_Thshld_Yng3DDark;
ubThshld_Yng3DGray = 1 - ubinv_Thshld_Yng3DGray;
ubThshld_Yng3WDark = 1 - ubinv_Thshld_Yng3WDark;
ubThshld_Yng3WGray = 1 - ubinv_Thshld_Yng3WGray;
ubThshld_Yng3CDark = 1 - ubinv_Thshld_Yng3CDark;
ubThshld_Yng3CGray = 1 - ubinv_Thshld_Yng3CGray;
ubThshld_Yng4DDark = 1 - ubinv_Thshld_Yng4DDark;
ubThshld_Yng4DGray = 1 - ubinv_Thshld_Yng4DGray;
ubThshld_Yng4WDark = 1 - ubinv_Thshld_Yng4WDark;
ubThshld_Yng4WGray = 1 - ubinv_Thshld_Yng4WGray;
ubThshld_Yng4CDark = 1 - ubinv_Thshld_Yng4CDark;
ubThshld_Yng4CGray = 1 - ubinv_Thshld_Yng4CGray;

%% Plot Thresholds
if plt2on
    xaxis = [0, 21.53, 32.29, 43.06];
    % Old
    olddeskdark = [Thshld_OldNoDark, Thshld_Old2DDark, Thshld_Old3DDark, …
       Thshld_Old4DDark];
    oldwalldark = [Thshld_OldNoDark, Thshld_Old2WDark, Thshld_Old3WDark, …
       Thshld_Old4WDark];
    oldclngdark = [Thshld_OldNoDark, Thshld_Old2CDark, Thshld_Old3CDark, …
       Thshld_Old4CDark];
    lbolddeskdark = [lbThshld_OldNoDark, lbThshld_Old2DDark, lbThshld_Old3DDark, …
       lbThshld_Old4DDark];
    lboldwalldark = [lbThshld_OldNoDark, lbThshld_Old2WDark, lbThshld_Old3WDark, …
       lbThshld_Old4WDark];
    lboldclngdark = [lbThshld_OldNoDark, lbThshld_Old2CDark, lbThshld_Old3CDark, …
       lbThshld_Old4CDark];
    ubolddeskdark = [ubThshld_OldNoDark, ubThshld_Old2DDark, ubThshld_Old3DDark, …
       ubThshld_Old4DDark];
```

```
uboldwalldark = [ubThshld_OldNoDark, ubThshld_Old2WDark, ubThshld_Old3WDark, …
   ubThshld_Old4WDark];
uboldclngdark = [ubThshld_OldNoDark, ubThshld_Old2CDark, ubThshld_Old3CDark, …
   ubThshld_Old4CDark];
figure;
plot(xaxis, olddeskdark, 'rx-','LineWidth',1);
hold on;
plot(xaxis, oldwalldark, 'bx-','LineWidth',1);
plot(xaxis, oldclngdark, 'kx-','LineWidth',1);
fill([xaxis fliplr(xaxis)],[ubolddeskdark fliplr(lbolddeskdark)],'r', …
   'linestyle','none');
fill([xaxis fliplr(xaxis)],[uboldwalldark fliplr(lboldwalldark)],'b', …
   'linestyle','none');
fill([xaxis fliplr(xaxis)],[uboldclngdark fliplr(lboldclngdark)],'k', …
   'linestyle','none');
alpha(0.25);
ylim([0 0.1])
yaxis = 0:.02:.1;
yticks(yaxis);
xticks(xaxis);
xlabel('Illumination Condition (lx)');
ylabel('Contrast Threshold');
legend('Desk Condition','Wall Condition','Ceiling Condition','Desk CI'...
    ,'Wall CI','Ceiling CI','Location','southwest');
title('Over 29 Years Dark Background')

figure;
olddeskgray = [Thshld_OldNoGray, Thshld_Old2DGray, Thshld_Old3DGray, …
   Thshld_Old4DGray];
oldwallgray = [Thshld_OldNoGray, Thshld_Old2WGray, Thshld_Old3WGray, …
   Thshld_Old4WGray];
oldclnggray = [Thshld_OldNoGray, Thshld_Old2CGray, Thshld_Old3CGray, …
   Thshld_Old4CGray];
lbolddeskgray = [lbThshld_OldNoGray, lbThshld_Old2DGray, lbThshld_Old3DGray, …
   lbThshld_Old4DGray];
lboldwallgray = [lbThshld_OldNoGray, lbThshld_Old2WGray, lbThshld_Old3WGray, …
   lbThshld_Old4WGray];
lboldclnggray = [lbThshld_OldNoGray, lbThshld_Old2CGray, lbThshld_Old3CGray, …
   lbThshld_Old4CGray];
ubolddeskgray = [ubThshld_OldNoGray, ubThshld_Old2DGray, ubThshld_Old3DGray, …
   ubThshld_Old4DGray];
uboldwallgray = [ubThshld_OldNoGray, ubThshld_Old2WGray, ubThshld_Old3WGray, …
   ubThshld_Old4WGray];
uboldclnggray = [ubThshld_OldNoGray, ubThshld_Old2CGray, ubThshld_Old3CGray, …
   ubThshld_Old4CGray];
plot(xaxis, olddeskgray, 'rx-','LineWidth',1);
hold on;
plot(xaxis, oldwallgray, 'bx-','LineWidth',1);
plot(xaxis, oldclnggray, 'kx-','LineWidth',1);
fill([xaxis fliplr(xaxis)],[ubolddeskgray fliplr(lbolddeskgray)],'r', …
   'linestyle','none');
fill([xaxis fliplr(xaxis)],[uboldwallgray fliplr(lboldwallgray)],'b', …
   'linestyle','none');
fill([xaxis fliplr(xaxis)],[uboldclnggray fliplr(lboldclnggray)],'k', …
   'linestyle','none');
```

```
alpha(0.25);
ylim([0.002 0.017])
yaxis = 0.002:.003:.017;
ax = gca;
ax.YRuler.Exponent = 0;
yticks(yaxis);
xticks(xaxis);
xlabel('Illumination Condition (lx)');
ylabel('Contrast Threshold');
legend('Desk Condition','Wall Condition','Ceiling Condition','Desk CI'...
    ,'Wall CI','Ceiling CI','Location','southwest');
title('Over 29 Years Light Background')

% Young
yngdeskdark = [Thshld_YngNoDark, Thshld_Yng2DDark, Thshld_Yng3DDark, …
   Thshld_Yng4DDark];
yngwalldark = [Thshld_YngNoDark, Thshld_Yng2WDark, Thshld_Yng3WDark, …
   Thshld_Yng4WDark];
yngclngdark = [Thshld_YngNoDark, Thshld_Yng2CDark, Thshld_Yng3CDark, …
   Thshld_Yng4CDark];
lbyngdeskdark = [lbThshld_YngNoDark, lbThshld_Yng2DDark, lbThshld_Yng3DDark, …
   lbThshld_Yng4DDark];
lbyngwalldark = [lbThshld_YngNoDark, lbThshld_Yng2WDark, lbThshld_Yng3WDark, …
   lbThshld_Yng4WDark];
lbyngclngdark = [lbThshld_YngNoDark, lbThshld_Yng2CDark, lbThshld_Yng3CDark, …
   lbThshld_Yng4CDark];
ubyngdeskdark = [ubThshld_YngNoDark, ubThshld_Yng2DDark, ubThshld_Yng3DDark, …
   ubThshld_Yng4DDark];
ubyngwalldark = [ubThshld_YngNoDark, ubThshld_Yng2WDark, ubThshld_Yng3WDark, …
   ubThshld_Yng4WDark];
ubyngclngdark = [ubThshld_YngNoDark, ubThshld_Yng2CDark, ubThshld_Yng3CDark, …
   ubThshld_Yng4CDark];
figure;
plot(xaxis, yngdeskdark, 'rx-','LineWidth',1);
hold on;
plot(xaxis, yngwalldark, 'bx-','LineWidth',1);
plot(xaxis, yngclngdark, 'kx-','LineWidth',1);
fill([xaxis fliplr(xaxis)],[ubyngdeskdark fliplr(lbyngdeskdark)],'r', …
   'linestyle','none');
fill([xaxis fliplr(xaxis)],[ubyngwalldark fliplr(lbyngwalldark)],'b', …
   'linestyle','none');
fill([xaxis fliplr(xaxis)],[ubyngclngdark fliplr(lbyngclngdark)],'k', …
   'linestyle','none');
alpha(0.25);
ylim([0 0.1])
yaxis = 0:.02:.1;
yticks(yaxis);
xticks(xaxis);
xlabel('Illumination Condition (lx)');
ylabel('Contrast Threshold');
legend('Desk Condition','Wall Condition','Ceiling Condition','Desk CI'...
    ,'Wall CI','Ceiling CI','Location','southwest');
title('Under 29 Years Dark Background')

figure;
```

```
    yngdeskgray = [Thshld_YngNoGray, Thshld_Yng2DGray, Thshld_Yng3DGray, …
       Thshld_Yng4DGray];
    yngwallgray = [Thshld_YngNoGray, Thshld_Yng2WGray, Thshld_Yng3WGray, …
       Thshld_Yng4WGray];
    yngclnggray = [Thshld_YngNoGray, Thshld_Yng2CGray, Thshld_Yng3CGray, …
       Thshld_Yng4CGray];
    lbyngdeskgray = [lbThshld_YngNoGray, lbThshld_Yng2DGray, lbThshld_Yng3DGray, …
       lbThshld_Yng4DGray];
    lbyngwallgray = [lbThshld_YngNoGray, lbThshld_Yng2WGray, lbThshld_Yng3WGray, …
       lbThshld_Yng4WGray];
    lbyngclnggray = [lbThshld_YngNoGray, lbThshld_Yng2CGray, lbThshld_Yng3CGray, …
       lbThshld_Yng4CGray];
    ubyngdeskgray = [ubThshld_YngNoGray, ubThshld_Yng2DGray, ubThshld_Yng3DGray, …
       ubThshld_Yng4DGray];
    ubyngwallgray = [ubThshld_YngNoGray, ubThshld_Yng2WGray, ubThshld_Yng3WGray, …
       ubThshld_Yng4WGray];
    ubyngclnggray = [ubThshld_YngNoGray, ubThshld_Yng2CGray, ubThshld_Yng3CGray, …
       ubThshld_Yng4CGray];
    plot(xaxis, yngdeskgray, 'rx-','LineWidth',1);
    hold on;
    plot(xaxis, yngwallgray, 'bx-','LineWidth',1);
    plot(xaxis, yngclnggray, 'kx-','LineWidth',1);
    fill([xaxis fliplr(xaxis)],[ubyngdeskgray fliplr(lbyngdeskgray)],'r', …
       'linestyle','none');
    fill([xaxis fliplr(xaxis)],[ubyngwallgray fliplr(lbyngwallgray)],'b', …
       'linestyle','none');
    fill([xaxis fliplr(xaxis)],[ubyngclnggray fliplr(lbyngclnggray)],'k', …
       'linestyle','none');
    alpha(0.25);
    ylim([0.002 0.017])
    yaxis = 0.002:.003:.017;
    ax = gca;
    ax.YRuler.Exponent = 0;
    yticks(yaxis);
    xticks(xaxis);
    xlabel('Illumination Condition (lx)');
    ylabel('Contrast Threshold');
    legend('Desk Condition','Wall Condition','Ceiling Condition','Desk CI'...
        ,'Wall CI','Ceiling CI','Location','southwest');
    title('Under 29 Light Background')
end
```

Appendix G

Matlab Source Code to Calculate Response Times

All Participants, Grouped by Background Shade

```
close all;
format compact;
clear variables;

%% Define Parameters
plt2on = true;
saveon = false;
myfile = 'participantdata.xlsx';

%% Load Participant Data
VertIllum = xlsread(myfile, 'F2:F4801');
Background = xlsread(myfile, 'K2:K4801');
Accuracy = xlsread(myfile, 'P2:P4801');
ResponseTime = xlsread(myfile, 'Q2:Q4801');
Michelson = xlsread(myfile, 'Z2:Z4801');
RawMichelson = xlsread(myfile, 'AA2:AA4801');

%% Create Data Groups
AllNoDark = ResponseTime(VertIllum==0 & Background==.035);
AllNoGray = ResponseTime(VertIllum==0 & Background==.4);
All2DDark = ResponseTime(VertIllum==9.58 & Background==.035);
All2DGray = ResponseTime(VertIllum==9.58 & Background==.4);
All2WDark = ResponseTime(VertIllum==9.75 & Background==.035);
All2WGray = ResponseTime(VertIllum==9.75 & Background==.4);
All2CDark = ResponseTime(VertIllum==12.24 & Background==.035);
All2CGray = ResponseTime(VertIllum==12.24 & Background==.4);
All3DDark = ResponseTime(VertIllum==17.18 & Background==.035);
All3DGray = ResponseTime(VertIllum==17.18 & Background==.4);
All3WDark = ResponseTime(VertIllum==17.21 & Background==.035);
All3WGray = ResponseTime(VertIllum==17.21 & Background==.4);
All3CDark = ResponseTime(VertIllum==22.11 & Background==.035);
All3CGray = ResponseTime(VertIllum==22.11 & Background==.4);
All4DDark = ResponseTime(VertIllum==23.67 & Background==.035);
All4DGray = ResponseTime(VertIllum==23.67 & Background==.4);
All4WDark = ResponseTime(VertIllum==23.93 & Background==.035);
All4WGray = ResponseTime(VertIllum==23.93 & Background==.4);
All4CDark = ResponseTime(VertIllum==28.9 & Background==.035);
All4CGray = ResponseTime(VertIllum==28.9 & Background==.4);

%% Calculate Average Response Times
ART_AllNoDark = mean(AllNoDark);
ART_AllNoGray = mean(AllNoGray);
ART_All2DDark = mean(All2DDark);
ART_All2DGray = mean(All2DGray);
ART_All2WDark = mean(All2WDark);
ART_All2WGray = mean(All2WGray);
ART_All2CDark = mean(All2CDark);
ART_All2CGray = mean(All2CGray);
```

```
ART_All3DDark = mean(All3DDark);
ART_All3DGray = mean(All3DGray);
ART_All3WDark = mean(All3WDark);
ART_All3WGray = mean(All3WGray);
ART_All3CDark = mean(All3CDark);
ART_All3CGray = mean(All3CGray);
ART_All4DDark = mean(All4DDark);
ART_All4DGray = mean(All4DGray);
ART_All4WDark = mean(All4WDark);
ART_All4WGray = mean(All4WGray);
ART_All4CDark = mean(All4CDark);
ART_All4CGray = mean(All4CGray);

%% Plot Response Times
if plt2on
    xaxis = [0, 21.53, 32.29, 43.06];
    deskdark = [ART_AllNoDark, ART_All2DDark, ART_All3DDark, ART_All4DDark];
    walldark = [ART_AllNoDark, ART_All2WDark, ART_All3WDark, ART_All4WDark];
    clngdark = [ART_AllNoDark, ART_All2CDark, ART_All3CDark, ART_All4CDark];
    figure;
    plot(xaxis, deskdark, 'rx-','LineWidth',1);
    hold on;
    plot(xaxis, walldark, 'bx-','LineWidth',1);
    plot(xaxis, clngdark, 'kx-','LineWidth',1);
    xticks(xaxis);
    xlabel('Illumination Condition (lx)');
    ylabel('Average Response Time');
    legend('Desk Condition','Wall Condition','Ceiling Condition','Location', ...
        'southwest');
    title('All Participants Dark Background')

    figure;
    deskgray = [ART_AllNoGray, ART_All2DGray, ART_All3DGray, ART_All4DGray];
    wallgray = [ART_AllNoGray, ART_All2WGray, ART_All3WGray, ART_All4WGray];
    clnggray = [ART_AllNoGray, ART_All2CGray, ART_All3CGray, ART_All4CGray];
    plot(xaxis, deskgray, 'rx-','LineWidth',1);
    hold on;
    plot(xaxis, wallgray, 'bx-','LineWidth',1);
    plot(xaxis, clnggray, 'kx-','LineWidth',1);
    xticks(xaxis);
    xlabel('Illumination Condition (lx)');
    ylabel('Average Response Time');
    legend('Desk Condition','Wall Condition','Ceiling Condition','Location', ...
        'southwest');
    title('All Participants Light Background')
end
```

All Participants, Grouped by Age and Background Shade

```
close all;
format compact;
clear variables;

%% Define Parameters
plt2on = true;
```

```
saveon = false;
myfile = 'participantdata.xlsx';

%% Load Participant Data
VertIllum = xlsread(myfile, 'F2:F4801');
Background = xlsread(myfile, 'K2:K4801');
Accuracy = xlsread(myfile, 'P2:P4801');
ResponseTime = xlsread(myfile, 'Q2:Q4801');
Michelson = xlsread(myfile, 'Z2:Z4801');
RawMichelson = xlsread(myfile, 'AA2:AA4801');
Age = xlsread(myfile, 'AB2:AB4801');

%% Create Data Groups
% Over 29
OldNoDark = ResponseTime(VertIllum==0 & Background==.035 & Age>29);
OldNoGray = ResponseTime(VertIllum==0 & Background==.4 & Age>29);
Old2DDark = ResponseTime(VertIllum==9.58 & Background==.035 & Age>29);
Old2DGray = ResponseTime(VertIllum==9.58 & Background==.4 & Age>29);
Old2WDark = ResponseTime(VertIllum==9.75 & Background==.035 & Age>29);
Old2WGray = ResponseTime(VertIllum==9.75 & Background==.4 & Age>29);
Old2CDark = ResponseTime(VertIllum==12.24 & Background==.035 & Age>29);
Old2CGray = ResponseTime(VertIllum==12.24 & Background==.4 & Age>29);
Old3DDark = ResponseTime(VertIllum==17.18 & Background==.035 & Age>29);
Old3DGray = ResponseTime(VertIllum==17.18 & Background==.4 & Age>29);
Old3WDark = ResponseTime(VertIllum==17.21 & Background==.035 & Age>29);
Old3WGray = ResponseTime(VertIllum==17.21 & Background==.4 & Age>29);
Old3CDark = ResponseTime(VertIllum==22.11 & Background==.035 & Age>29);
Old3CGray = ResponseTime(VertIllum==22.11 & Background==.4 & Age>29);
Old4DDark = ResponseTime(VertIllum==23.67 & Background==.035 & Age>29);
Old4DGray = ResponseTime(VertIllum==23.67 & Background==.4 & Age>29);
Old4WDark = ResponseTime(VertIllum==23.93 & Background==.035 & Age>29);
Old4WGray = ResponseTime(VertIllum==23.93 & Background==.4 & Age>29);
Old4CDark = ResponseTime(VertIllum==28.9 & Background==.035 & Age>29);
Old4CGray = ResponseTime(VertIllum==28.9 & Background==.4 & Age>29);

% Under 29
YngNoDark = ResponseTime(VertIllum==0 & Background==.035 & Age<29);
YngNoGray = ResponseTime(VertIllum==0 & Background==.4 & Age<29);
Yng2DDark = ResponseTime(VertIllum==9.58 & Background==.035 & Age<29);
Yng2DGray = ResponseTime(VertIllum==9.58 & Background==.4 & Age<29);
Yng2WDark = ResponseTime(VertIllum==9.75 & Background==.035 & Age<29);
Yng2WGray = ResponseTime(VertIllum==9.75 & Background==.4 & Age<29);
Yng2CDark = ResponseTime(VertIllum==12.24 & Background==.035 & Age<29);
Yng2CGray = ResponseTime(VertIllum==12.24 & Background==.4 & Age<29);
Yng3DDark = ResponseTime(VertIllum==17.18 & Background==.035 & Age<29);
Yng3DGray = ResponseTime(VertIllum==17.18 & Background==.4 & Age<29);
Yng3WDark = ResponseTime(VertIllum==17.21 & Background==.035 & Age<29);
Yng3WGray = ResponseTime(VertIllum==17.21 & Background==.4 & Age<29);
Yng3CDark = ResponseTime(VertIllum==22.11 & Background==.035 & Age<29);
Yng3CGray = ResponseTime(VertIllum==22.11 & Background==.4 & Age<29);
Yng4DDark = ResponseTime(VertIllum==23.67 & Background==.035 & Age<29);
Yng4DGray = ResponseTime(VertIllum==23.67 & Background==.4 & Age<29);
Yng4WDark = ResponseTime(VertIllum==23.93 & Background==.035 & Age<29);
Yng4WGray = ResponseTime(VertIllum==23.93 & Background==.4 & Age<29);
Yng4CDark = ResponseTime(VertIllum==28.9 & Background==.035 & Age<29);
```

```
Yng4CGray = ResponseTime(VertIllum==28.9 & Background==.4 & Age<29);

%% Calculate Average Response Times
ART_OldNoDark = mean(OldNoDark);
ART_OldNoGray = mean(OldNoGray);
ART_Old2DDark = mean(Old2DDark);
ART_Old2DGray = mean(Old2DGray);
ART_Old2WDark = mean(Old2WDark);
ART_Old2WGray = mean(Old2WGray);
ART_Old2CDark = mean(Old2CDark);
ART_Old2CGray = mean(Old2CGray);
ART_Old3DDark = mean(Old3DDark);
ART_Old3DGray = mean(Old3DGray);
ART_Old3WDark = mean(Old3WDark);
ART_Old3WGray = mean(Old3WGray);
ART_Old3CDark = mean(Old3CDark);
ART_Old3CGray = mean(Old3CGray);
ART_Old4DDark = mean(Old4DDark);
ART_Old4DGray = mean(Old4DGray);
ART_Old4WDark = mean(Old4WDark);
ART_Old4WGray = mean(Old4WGray);
ART_Old4CDark = mean(Old4CDark);
ART_Old4CGray = mean(Old4CGray);

ART_YngNoDark = mean(YngNoDark);
ART_YngNoGray = mean(YngNoGray);
ART_Yng2DDark = mean(Yng2DDark);
ART_Yng2DGray = mean(Yng2DGray);
ART_Yng2WDark = mean(Yng2WDark);
ART_Yng2WGray = mean(Yng2WGray);
ART_Yng2CDark = mean(Yng2CDark);
ART_Yng2CGray = mean(Yng2CGray);
ART_Yng3DDark = mean(Yng3DDark);
ART_Yng3DGray = mean(Yng3DGray);
ART_Yng3WDark = mean(Yng3WDark);
ART_Yng3WGray = mean(Yng3WGray);
ART_Yng3CDark = mean(Yng3CDark);
ART_Yng3CGray = mean(Yng3CGray);
ART_Yng4DDark = mean(Yng4DDark);
ART_Yng4DGray = mean(Yng4DGray);
ART_Yng4WDark = mean(Yng4WDark);
ART_Yng4WGray = mean(Yng4WGray);
ART_Yng4CDark = mean(Yng4CDark);
ART_Yng4CGray = mean(Yng4CGray);

%% Plot Thresholds
if plt2on
    xaxis = [0, 21.53, 32.29, 43.06];
    % Over 29
    olddeskdark = [ART_OldNoDark, ART_Old2DDark, ART_Old3DDark, ART_Old4DDark];
    oldwalldark = [ART_OldNoDark, ART_Old2WDark, ART_Old3WDark, ART_Old4WDark];
    oldclngdark = [ART_OldNoDark, ART_Old2CDark, ART_Old3CDark, ART_Old4CDark];
    figure;
    plot(xaxis, olddeskdark, 'rx-','LineWidth',1);
    hold on;
    plot(xaxis, oldwalldark, 'bx-','LineWidth',1);
    plot(xaxis, oldclngdark, 'kx-','LineWidth',1);
    xticks(xaxis);
    xlabel('Illumination Condition (lx)');
    ylabel('Average Response Time');
    legend('Desk Condition','Wall Condition','Ceiling Condition','Location', …
        'southwest');
    title('Over 29 Years Dark Background')

    figure;
    olddeskgray = [ART_OldNoGray, ART_Old2DGray, ART_Old3DGray, ART_Old4DGray];
    oldwallgray = [ART_OldNoGray, ART_Old2WGray, ART_Old3WGray, ART_Old4WGray];
    oldclnggray = [ART_OldNoGray, ART_Old2CGray, ART_Old3CGray, ART_Old4CGray];
    plot(xaxis, olddeskgray, 'rx-','LineWidth',1);
    hold on;
    plot(xaxis, oldwallgray, 'bx-','LineWidth',1);
    plot(xaxis, oldclnggray, 'kx-','LineWidth',1);
    xticks(xaxis);
    xlabel('Illumination Condition (lx)');
    ylabel('Average Response Time');
    legend('Desk Condition','Wall Condition','Ceiling Condition','Location', …
        'southwest');
    title('Over 29 Years Light Background')

    % Under 29
    yngdeskdark = [ART_YngNoDark, ART_Yng2DDark, ART_Yng3DDark, ART_Yng4DDark];
    yngwalldark = [ART_YngNoDark, ART_Yng2WDark, ART_Yng3WDark, ART_Yng4WDark];
    yngclngdark = [ART_YngNoDark, ART_Yng2CDark, ART_Yng3CDark, ART_Yng4CDark];
    figure;
    plot(xaxis, yngdeskdark, 'rx-','LineWidth',1);
    hold on;
    plot(xaxis, yngwalldark, 'bx-','LineWidth',1);
    plot(xaxis, yngclngdark, 'kx-','LineWidth',1);
    xticks(xaxis);
    xlabel('Illumination Condition (lx)');
    ylabel('Average Response Time');
    legend('Desk Condition','Wall Condition','Ceiling Condition','Location', …
        'southwest');
    title('Under 29 Years Dark Background')

    figure;
    yngdeskgray = [ART_YngNoGray, ART_Yng2DGray, ART_Yng3DGray, ART_Yng4DGray];
    yngwallgray = [ART_YngNoGray, ART_Yng2WGray, ART_Yng3WGray, ART_Yng4WGray];
    yngclnggray = [ART_YngNoGray, ART_Yng2CGray, ART_Yng3CGray, ART_Yng4CGray];
    plot(xaxis, yngdeskgray, 'rx-','LineWidth',1);
    hold on;
    plot(xaxis, yngwallgray, 'bx-','LineWidth',1);
    plot(xaxis, yngclnggray, 'kx-','LineWidth',1);
    xticks(xaxis);
    xlabel('Illumination Condition (lx)');
    ylabel('Average Response Time');
    legend('Desk Condition','Wall Condition','Ceiling Condition','Location', …
        'southwest');
    title('Under 29 Light Background')
end
```

Bibliography

[1] R. Sithravel, R. Ibrahim, N. D. Dahlan, N. Ibrahim, and E. K. Perimal, "Identifying daily lighting environment to enhance occupant's psychophysiological wellbeing for healthy green building design," *Adv. Environ. Biol.*, vol. 9, no. 23, pp. 31–39, 2015.

[2] P. R. Mills, S. C. Tomkins, and L. J. Schlangen, "The effect of high correlated colour temperature office lighting on employee wellbeing and work performance," 2007.

[3] National Center For Geospatial Intelligence Standards, "Softcopy Exploitation Facility Standard," 2019.

[4] J. Hraska, "Chronobiological aspects of green buildings daylighting," *Renew. Energy*, vol. 73, pp. 109–114, 2015.

[5] M. Wacker and M. F. Holick, "Sunlight and Vitamin D: A global perspective for health," *Dermato-Endocrinology*, vol. 5, no. 1. pp. 51–108, 2013.

[6] S. E. P. Munce, S. A. Stansfeld, E. R. Blackmore, and D. E. Stewart, "The Role of Depression and Chronic Pain Conditions in Absenteeism: Results From a National Epidemiologic Survey," *J. Occup. Environ. Med.*, vol. 49, no. 11, pp. 1206–1211, Nov. 2007.

[7] T. Åkerstedt, "Shift work and disturbed sleep/wakefulness," *Sleep Med. Rev.*, vol. 2, no. 2, pp. 117–128, 1998.

[8] B. Rusak and I. Zucker, "Neural Regulation of Circadian Rhythms," *Lancet*, vol. 59, no. 3, pp. 449–526, 1979.

[9] L. Bellia, F. Bisegna, and G. Spada, "Lighting in indoor environments: Visual and non-visual effects of light sources with different spectral power distributions," *Build. Environ.*, vol. 46, no. 10, pp. 1984–1992, 2011.

[10] R. Hardeland, S. R. Pandi-Perumal, and D. P. Cardinali, "Melatonin," *Int. J. Biochem. Cell Biol.*, vol. 38, no. 3, pp. 313–316, 2006.

[11] R. Hardeland, D. P. Cardinali, V. Srinivasan, D. W. Spence, G. M. Brown, and S. R. Pandi-Perumal, "Melatonin-A pleiotropic, orchestrating regulator molecule," *Prog. Neurobiol.*, vol. 93, no. 3, pp. 350–384, 2011.

[12] S. M. Reppert and D. R. Weaver, “Melatonin madness,” *Cell*, vol. 83, no. 7, pp. 1059–1062, 1995.

[13] K. Mishima, M. Okawa, T. Shimizu, and Y. Hishikawa, “Diminished melatonin secretion in the elderly caused by insufficient environmental illumination,” *J. Clin. Endocrinol. Metab.*, vol. 86, no. 1, pp. 129–134, 2001.

[14] C. E. Rash, M. B. Russo, T. R. Letowski, and E. T. Schmeisser, “Helmet-Mounted Displays: Sensation, Perception and Cognition Issues,” Fort Rucker, Alabama, 2009.

[15] M. S. Rea, M. G. Figueiro, A. Bierman, and R. Hamner, “Modelling the spectral sensitivity of the human circadian system,” *Light. Res. Technol.*, vol. 44, no. 4, pp. 386–396, 2012.

[16] C. Cajochen, “Alerting effects of light,” *Sleep Med. Rev.*, vol. 11, no. 6, pp. 453–464, 2007.

[17] G. C. Brainard *et al.*, “Action Spectrum for Melatonin Regulation in Humans: Evidence for a Novel Circadian Photoreceptor,” *J. Neurosci.*, vol. 21, no. 16, pp. 6405–6412, 2001.

[18] K. Thapan, J. Arendt, and D. J. Skene, “An action spectrum for melatonin suppression: evidence for a novel non-rod, non-cone photoreceptor system in humans,” *J. Physiol.*, vol. 535, no. 1, pp. 261–267, Aug. 2001.

[19] C. Gronfier, K. P. Wright, R. E. Kronauer, M. E. Jewett, and C. A. Czeisler, “Efficacy of a single sequence of intermittent bright light pulses for delaying circadian phase in humans,” *Am. J. Physiol. - Endocrinol. Metab.*, vol. 287, no. 1 50-1, pp. 174–181, 2004.

[20] P. Khademagha, M. B. C. Aries, A. L. P. Rosemann, and E. J. van Loenen, “Implementing non-image-forming effects of light in the built environment: A review on what we need,” *Build. Environ.*, vol. 108, pp. 263–272, Nov. 2016.

[21] C. Lok, “Vision science: Seeing without seeing,” *Nature*, vol. 469, no. 7330, pp. 284–285, 2011.

[22] M. S. Rea, M. G. Figueiro, J. D. Bullough, and A. Bierman, “A model of phototransduction by the human circadian system,” *Brain Res. Rev.*, vol. 50, no. 2, pp. 213–228, 2005.

[23] M. G. Figueiro, B. Wood, B. Plitnick, and M. S. Rea, “The impact of light from computer monitors on melatonin levels in college students,” *Neuroendocrinol. Lett.*, vol. 32, no. 2, pp. 158–163, 2011.

[24] G. Glickman, J. P. Hanifin, M. D. Rollag, J. Wang, H. Cooper, and G. C. Brainard, “Inferior retinal light exposure is more effective than superior retinal exposure in suppressing melatonin in humans,” *J. Biol. Rhythms*, vol. 18, no. 1, pp. 71–79, 2003.

[25] T. A. Lasko, D. F. Kripke, and J. A. Elliot, “Melatonin suppression by illumination of upper and lower visual fields,” *J. Biol. Rhythms*, vol. 14, no. 2, pp. 122–125, 1999.

[26] M. Rüger, M. C. M. Gordijn, D. G. M. Beersma, B. De Vries, and S. Daan, “Nasal versus temporal illumination of the human retina: Effects on core body temperature, melatonin, and circadian phase,” *J. Biol. Rhythms*, vol. 20, no. 1, pp. 60–70, 2005.

[27] E. K. Visser, D. G. M. Beersma, and S. Daan, “Melatonin suppression by light in humans is maximal when the nasal part of the retina is illuminated,” *J. Biol. Rhythms*, vol. 14, no. 2, pp. 116–121, 1999.

[28] M. B. C. Aries, M. P. J. Aarts, and J. Van Hoof, “Daylight and health: A review of the evidence and consequences for the built environment,” *Light. Res. Technol.*, vol. 47, no. 1, pp. 6–27, 2015.

[29] R. C. Espiritu *et al.*, “Low Illumination Experienced by San Diego Adults: Association with Atypical Depressive Symptoms,” *Biol. Psychiatry*, vol. 35, no. 6, pp. 403–407, 1994.

[30] R. R. Sakamoto, “Sunlight in Vitamin D Deficiency: Clinical Implications,” *J. Nurse Pract.*, vol. 15, no. 4, pp. 282–285, 2019.

[31] A. Alloubani, L. Akhu-Zaheya, R. Samara, I. Abdulhafiz, A. Saleh, and A. Altowijri, “Relationship between Vitamin D Deficiency, Diabetes, and Obesity,” *Diabetes Metab. Syndr. Clin. Res. Rev.*, vol. 13, no. 2, pp. 1457–1461, 2019.

[32] W. B. Grant, H. P. Bhattoa, and P. Pludowski, *Determinants of Vitamin D Deficiency From Sun Exposure: A Global Perspective*, Fourth Edi., vol. 2. Elsevier, 2017.

[33] A. Ascherio, K. L. Munger, and K. C. Simon, “Vitamin D and multiple sclerosis,”

Lancet Neurol., vol. 9, no. 6, pp. 599–612, 2010.

[34] M. T. Kampman and L. H. Steffensen, “The role of vitamin D in multiple sclerosis,” *J. Photochem. Photobiol. B Biol.*, vol. 101, no. 2, pp. 137–141, 2010.

[35] P. Badia, B. Myers, M. Boecker, J. Culpepper, and J. R. Harsh, “Bright light effects on body temperature, alertness, EEG and behavior,” *Physiol. Behav.*, vol. 50, no. 3, pp. 583–588, 1991.

[36] F. A. J. L. Scheer, L. J. P. Van Doornen, and R. M. Buijs, “Light and diurnal cycle affect human heart rate: Possible role for the circadian pacemaker,” *J. Biol. Rhythms*, vol. 14, no. 3, pp. 202–212, 1999.

[37] F. A. J. L. Scheer, L. J. P. Van Doornen, and R. M. Buijs, “Light and diurnal cycle affect autonomic cardiac balance in human; possible role for the biological clock,” *Auton. Neurosci. Basic Clin.*, vol. 110, no. 1, pp. 44–48, 2004.

[38] M. S. Rea and M. G. Figueiro, “Light as a circadian stimulus for architectural lighting,” *Light. Res. Technol.*, vol. 50, no. 4, pp. 497–510, 2018.

[39] C. Cajochen, J. M. Zeitzer, C. A. Czeisler, and D. J. Dijk, “Dose-response relationship for light intensity and alertness and its ocular and EEG correlates,” *Behav. Brain Res.*, vol. 115, no. 1, pp. 75–83, 2000.

[40] Q. Dai, W. Cai, W. Shi, L. Hao, and M. Wei, “A proposed lighting-design space: circadian effect versus visual illuminance,” *Build. Environ.*, vol. 122, pp. 287–293, 2017.

[41] J. Rostron, “Sick building syndrome: A review of causes, consequences and remedies,” *J. Retail Leis. Prop.*, vol. 7, no. 4, pp. 291–303, 2008.

[42] D. Hafiz, “Daylighting, Space, and Architecture: A Literature Review,” *Enq. A J. Archit. Res.*, vol. 12, no. 1, pp. 1–8, 2015.

[43] G. V. Hultgren and B. Knave, “Discomfort glare and disturbances from light reflections in an office landscape with CRT display terminals,” *Appl. Ergon.*, vol. 5, no. 1, pp. 2–8, 1974.

[44] O. Östberg, H. Shahnavaz, and R. Stenberg, “CRT flicker and scan-line direction,” *Displays*, vol. 8, no. 2, pp. 75–78, 1987.

[45] M. Menozzi, F. Lang, U. Näpflin, C. Zeller, and H. Krueger, “CRT versus LCD:

Effects of refresh rate, display technology and background luminance in visual performance," *Displays*, vol. 22, no. 3, pp. 79–85, 2001.

[46] M. Menozzi, U. Näpflin, and H. Krueger, "CRT versus LCD: A pilot study on visual performance and suitability of two display technologies for use in office work," *Displays*, vol. 20, pp. 3–10, 1999.

[47] K.-K. Shieh and C.-C. Lin, "Effects of screen type, ambient illumination, and color combination on VDT visual performance and subjective preference," *Int. J. Ind. Ergon.*, vol. 26, no. 5, pp. 527–536, 2000.

[48] Y. M. Sung, M. J. Chung, K. S. Lee, and B. K. Choe, "The influence of liquid crystal display monitors on observer performance for the detection of interstitial lung markings on both storage phosphor and flat-panel-detector chest radiography," *Eur. J. Radiol.*, vol. 74, no. 1, pp. 275–279, 2010.

[49] M.-T. Chen and C.-C. Lin, "Comparison of TFT-LCD and CRT on visual recognition and subjective preference," *Int. J. Ind. Ergon.*, vol. 34, pp. 167–174, 2004.

[50] R. Näsänen, J. Karlsson, and H. Ojanpää, "Display quality and the speed of visual letter search," *Displays*, vol. 22, no. 4, pp. 107–113, 2001.

[51] C. Oglevee and O. Pianykh, "Losing Images in Digital Radiology: More than You Think," *J. Digit. Imaging*, vol. 28, no. 3, pp. 264–271, Jun. 2015.

[52] A. S. Chawla and E. Samei, "Ambient illumination revisited: A new adaptation-based approach for optimizing medical imaging reading environments," *Med. Phys.*, vol. 34, no. 1, pp. 81–90, 2007.

[53] A. Wolska and M. Śwituta, "Luminance of the surround and visual fatigue of vdt operators," *Int. J. Occup. Saf. Ergon.*, vol. 5, no. 4, pp. 553–580, 1999.

[54] B. J. Pollard, E. Samei, A. S. Chawla, J. Baker, S. Ghate, and C. Kim, "The Influence of Increased Ambient Lighting on Mass Detection in Mammograms," *Acad. Radiol.*, vol. 16, no. 3, pp. 299–304, 2009.

[55] B. J. Pollard *et al.*, "The effects of ambient lighting in chest radiology reading rooms," *J. Digit. Imaging*, vol. 25, pp. 520–526, 2012.

[56] J. H. Krantz, L. D. Silverstein, and Y.-Y. Yeh, "Visibility of Transmissive Liquid Crystal Displays under Dynamic Lighting Conditions," *Hum. Factors*, vol. 34, no.

5, pp. 615–632, 1992.

[57] A. Wang, H.-T. Kuo, and S. Jeng, “Effects of ambient illuminance on users ’ visual performance using various electronic displays,” *J. Soc. Inf. Disp.*, vol. 17, no. 8, pp. 665–669, 2009.

[58] P. C. Brennan, M. McEntee, M. Evanoff, P. Phillips, W. T. O’Connor, and D. J. Manning, “Ambient lighting: effect of illumination on soft-copy viewing of radiographs of the wrist.,” *AJR. Am. J. Roentgenol.*, vol. 188, no. 2, pp. 177–180, 2007.

[59] A. R. Webb, “Considerations for lighting in the built environment: Non-visual effects of light,” *Energy Build.*, vol. 38, no. 7, pp. 721–727, 2006.

[60] National Center For Geospatial Intelligence Standards, “Display Performance Standard,” 2017.

[61] M. Miller, *Color in Electronic Display Systems: Advantages of Multi-primary Displays*. Springer, 2018.

[62] International Telecommunication Union, “Parameter values for the HDTV standards for production and international programme exchange BT Series Broadcasting service,” *Recomm. ITU-R BT.709-5*, vol. 5, pp. 1–32, 2002.

[63] B. J. Pollard, A. S. Chawla, D. M. Delong, N. Hashimoto, and E. Samei, “Object detectability at increased ambient lighting conditions,” *Med. Phys.*, vol. 35, no. 6, pp. 2204–2213, 2008.

[64] F.-Y. Tseng, C.-J. Chao, W.-Y. Feng, and S.-L. Hwang, “Assessment of human color discrimination based on illuminant color, ambient illumination and screen background color for visual display terminal workers.,” *Ind. Health*, vol. 48, no. 4, pp. 438–46, 2010.

[65] N. I. Lisdiani and Yassierli, “Blue light exposure improves awareness during monotonous activities at night,” in *2012 Southeast Asian Network of Ergonomics Societies Conference: Ergonomics Innovations Leveraging User Experience and Sustainability, SEANES 2012*, 2012.

[66] S. Van Der Lely *et al.*, “Blue blocker glasses as a countermeasure for alerting effects of evening light-emitting diode screen exposure in male teenagers,” *J. Adolesc. Heal.*, vol. 56, no. 1, pp. 113–119, 2015.

[67] B. Wood, M. S. Rea, B. Plitnick, and M. G. Figueiro, “Light level and duration of exposure determine the impact of self-luminous tablets on melatonin suppression,” *Appl. Ergon.*, vol. 44, no. 2, pp. 237–240, 2013.

[68] T. Sakurada, T. Kawase, T. Komatsu, and K. Kansaku, “Use of high-frequency visual stimuli above the critical flicker frequency in a SSVEP-based BMI,” *Clin. Neurophysiol.*, vol. 126, no. 10, pp. 1972–1978, 2015.

[69] B. J. Pollard *et al.*, “The Influence of Increased Ambient Lighting on Mass Detection in Mammograms,” *Acad. Radiol.*, vol. 16, no. 3, pp. 299–304, 2009.

[70] M. G. Figueiro, A. Bierman, B. Plitnick, and M. S. Rea, “Preliminary evidence that both blue and red light can induce alertness at night.,” *BMC Neurosci.*, vol. 10, no. 1, p. 105, Jan. 2009.

[71] B. J. Pollard, A. S. Chawla, N. Hashimoto, and E. Samei, “The effect of increased ambient lighting on detection accuracy in uniform and anatomical backgrounds,” *Med. Imaging 2008 PACS Imaging Informatics*, vol. 6919, no. 69190R, pp. 1–13, 2008.

[72] A. B. Watson and D. G. Pelli, “QUEST: A Bayesian adaptive psychometric method,” *Percept. Psychophys.*, vol. 33, no. 2, pp. 113–120, 1983.

[73] H. T. Zwahlen and N. Kothari, “Effects of Positive and Negative Image Polarity VDT Screens,” in *Proceedings of the Human Factors Society Annual Meeting*, 1986, vol. 30, no. 2, pp. 170–174.

[74] H. Hori and J. Kondo, “Contrast ratio for transmissive-type TFT-addressed LCDs under ambient-light illumination,” *J. Soc. Inf. Disp.*, vol. 1, no. 3, pp. 325–327, 1993.

[75] M. M. Schroeder *et al.*, “The roles of rods, cones, and melanopsin in photoresponses of M4 intrinsically photosensitive retinal ganglion cells (ipRGCs) and optokinetic visual behavior,” *Front. Cell. Neurosci.*, vol. 12, no. July, pp. 1–14, 2018.

[76] J. Cohen, “Quantitative Methods in Psychology: A Power Primer,” *Psychol. Bull.*, vol. 112, no. 1, pp. 155–159, 1992.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 26-03-2020	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From – To)* September 2018 – March 2020

TITLE AND SUBTITLE Electronic Image Detectability Under Varying Illumination Conditions	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Miller, Jeremy J., Major, USAF	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT-ENV-MS-20-M-228
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) 480th Intelligence, Surveillance and Reconnaissance Wing Joint Base Langley-Eustis 757-504-7734 alan.ogle@us.af.mil ATTN: Lt Col Alan Ogle	**10. SPONSOR/MONITOR'S ACRONYM(S)** 480 ISRW/SG **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRUBTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT
Light in the built environment plays an essential role in the vision and health of humans through non-visual receptors in the eyes. Unfortunately, image analysts who engage in the detection of objects on displays are often required to work in dimly-lit environments as higher illumination reduces display contrast. Literature has shown that increases in light exposure improves circadian rhythm entrainment and reduces the negative health consequences of insufficient lighting. This research examines the effects of indoor lighting to determine if increases in ambient illumination or changes to the orientation of light fixtures improves or degrades alertness and visual performance. Positive findings of this study would lead to changes in the environment of image analysts, increasing productivity and long-term health. In a controlled environment, twelve participants were exposed to D65 ambient light at levels of 0, 21, 32, and 43 lux. During exposures, participants were asked to visually identify variations in low-contrast Gabor patches on a display. The data was used to calculate the contrast threshold of detectability and response times of participants, thus indicating visual performance. Surveys also measured the subjective alertness and discomfort of participants. Results showed that the orientation of light fixtures significantly affects participant sleepiness, weariness, and discomfort. Additionally, the results indicate that visual detection performance is improved with higher ambient illumination levels employed within this research.

15. SUBJECT TERMS
ambient illumination, light, display, images, performance, alertness, orientation, health, non-image-forming effects, built environment

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. Michael E. Miller, AFIT/ENV
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, ext 4651 (NOT DSN) mmiller@afit.edu
U	U	U	UU	166	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18