Adapting an LCD for Weight Generation in an Electro-optic Neural Processor

by

Andrew Charles Lamson

B.S., EE, Massachusetts Institute of Technology (2011)

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

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author .....
Department of Electrical Engineering and Computer Science
May 28, 2013

Certified by .....
Cardinal Warde
Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Prof. Dennis M. Freeman
Chairman, Masters of Engineering Thesis Committee

Adapting an LCD for Weight Generation in an Electro-optic Neural Processor

by

Andrew Charles Lamson

Submitted to the Department of Electrical Engineering and Computer Science
on May 28, 2013, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis discusses adapting and testing an LCD as a weight image display for use in the Hybrid Electro-optical Neural Network (HENN). The HENN project is a proof of concept prototype hybrid neural network that will be used to gather information for a more advanced project in the future. After thoroughly explaining the HENN, this thesis characterizes the LCD selected for adaption. Within the characterization is a grouping of experiments that explore different aspects of the LCD screen. Then a few experiments are conducted to evaluate the interactions of the LCD and fiber optic interconnection plate. After this, the method used to generate the weighting image is explained thoroughly. The experimental evidence is gathered to show how the LCD can be used as a weighting system. Then based on the evidence gathered several recommendations are suggested to redesign the fiber plate and further improve the HENN system.

Thesis Supervisor: Cardinal Warde
Title: Professor of Electrical Engineering

Acknowledgments

I would like to thank my thesis advisor Cardinal Warde for his encouragement and guidance through all of my work. I would also like to thank Bill Herrington for his invaluable help and insight, and being there for every little problem and all the questions. I would like to thank my girlfriend Seoyeon Yang for her inspiration to pull through this busy time and to always aim for the better future. I would like to thank my sister for her consistent encouragement and familial concern, especially during our last semester in Boston. Finally, I would like to thank my parents for all their love and support through the years, I could not have done it without you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The purpose of this thesis is to develop a device that can implement connection weights and be used as an image input device in the Hybrid Electro-optic Neural Network system (HENN). It begins by outlining the project background and purposes and then thoroughly steps through the architecture. After establishing the project, it explains the work done for the development of this connection weight and input device for the HENN and follows with future improvements and new problems to investigate.

While much of the theoretical research is completed or has a strong basis in the group's previous paper, minimal progress has been made thus far on the physical prototyping for proof of concept [1]. Specifically, the creation of a viable method to implement optical weighting, a major hinge point of the project, remains to be addressed. Therefore, the primary contributions of this thesis are the research and development of an LCD weighting system that is constrained by integration with other HENN subsystems. This consists of characterization and testing of an LCD for the weighting system, redesigning the current fiber plate for optimal compatibility with the LCD, and the development of a program to reliably display the weighting image on the LCD screen. Through these steps, the thesis demonstrates how to build an LCD-based weighting system for the HENN (or any other electrical/optical hybrid neural network). Furthermore, utilizing the data and recommendations of this document will facilitate the accelerated construction of the fiber optic interconnection

plate and LCD-based weighting image subsystems.

Chapter 1 begins by introducing and explaining the usefulness of neural networks. The latter half focuses on the reasons for developing the HENN as a precursory step toward the Compact Optoelectronic Integrated Neuroprocessor (COIN). Chapter 2 presents a thorough examination of the HENN, explaining each layer of the system architecture and exemplifying the difference between the ideal HENN and the current version in the lab. Chapter 3 presents the LCD screen characterization for the weighting system. It features the specifications for selecting an LCD, establishment of the LCD pixel dimensions, the intensity wavelength profile analysis, and the relationship between pixel output intensity and RGB value. Chapter 4 builds upon the LCD characterization, by outlining the interfacing of the LCD and fiber plate with three different experiments: position-based light intensity variation, the single fiber loss, and the effects of errant light in the fiber optic interconnection plate. Chapter 5 uses the established information to explain the creation of a weighting image maker program and how it displays the weighting image. Chapter 6 is the culmination of the data gathered in the previous 3 chapters. After recounting the thesis data and knowledge gained, it proves the thesis has demonstrated how to modify an LCD for use as a weighting system. Then it outlines explicit recommendations for future work, including the redesign of the fiber plate component.

1.1 Background on Neural Networks

It is a fact that, when compared with microprocessors, human beings are vastly inferior in speed and accuracy for an overwhelming majority of problems. However, there is a small group of problems where even children can outperform computers. Simple activities like recognizing a friend with a new haircut or glasses, reading the sloppy handwriting of others, and identifying a traffic sign. It is here that humans seem to excel, whereas computers struggle to identify a friend, even without the new haircut or glasses. The reason is that the structure of the brain is much better fitted to deal with these problems. The brain is massively parallel and though not as precise,

it has the ability to deal with that imprecision or obstruction much better than the serial and extremely accurate processor. Consequently, this idea has developed into a brain-based computing structure to solve these types of problems.

The basic idea of neural networks is relatively simple, the units used for computing are artificial neurons [2][3]. Similar to the example shown in figure 1-1, an artificial neuron accepts multiple inputs, but only provides one output. For a four input artificial neuron model, each input signal $x0, ..., x4$ travels across the synapse, which has a specific weight $w0, ..., w4$. This weight amplifies or attenuates the signal before it reaches the neuron, where it is summed with the other weighted inputs. Having summed all inputs, the neuron applies a thresholding function to determine the state of the output y. If the sum of weighted values is above a cutoff, then the neuron determines that it is on, otherwise it is off. These output values are similar to digital 1's and 0's.

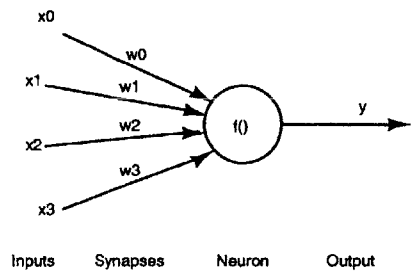

Figure 1-1: Basic neuron model with four inputs. Image taken from [1].

In order to create a computational system, these artificial neurons must be connected into a network [4]. A typical network might look something like figure 1-2. The example system has three inputs that are taken in by three input neurons. Since these are input neurons, they just forward the input value. The forwarded values are weighted by the hidden layer synapses and each hidden layer neuron has three weighted inputs to sum. Then the hidden layer neurons threshold the sums and output those signals to the synapse of each output neuron. The hidden layer outputs are weighted by the output layer synapses and the output neurons each receive two weighted values. Then, each sum is thresholded for the last time, and those values are the three system output values. Of course it becomes more and more complicated as

hidden layers and inputs are added. Eventually, with enough of each, extremely complex functions may be approximated, where their accuracy increases as the hidden layers and inputs do. However, the input and hidden layers are not the determining factor for the function emulated by the neural network. The adaptable characteristic of a neural network is the weights. Each neural network is trained by an algorithm to determine the weighting values. Depending on the function to be approximated or purposed of the neural network, the weightings will be very different. For example, if the neural network is for image pattern recognition, then the weights are valued to make the network output the same value for the same patterns, regardless of whether the overall image is the same or not. This is discussed further in Section 2.5.

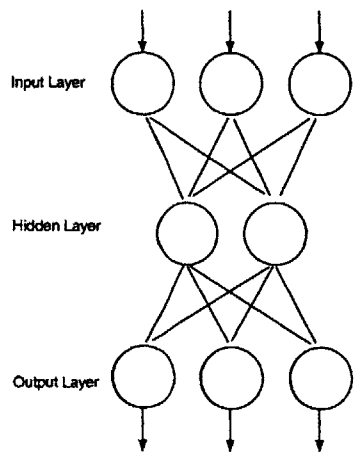

Figure 1-2: Three layer neural network with three inputs, one hidden layer, and three outputs. Image taken from [1].

Despite the exponential growth of microprocessor speed in recent history, traditional systems have been unable to replicate one of the key advantages for neural networks, parallelization. Most artificial neural network processing systems are just neural algorithms running on sequential processors. These software systems attempt simulating parallel environments on sequential devices, resulting in sub-optimal performance. Many past attempts to create electrical or electro-optic hybrid parallel physical neural networks have ended prematurely with hardware limitations that relegated processing to the sequential realm. The sheer number of interconnections required for larger systems and the number of necessary components for total parallel

processing makes system development a daunting task. This is not even including the design considerations required for an optical or electro-optic hybrid neural network. Yet, research continues due to the optimization advantage neural algorithms on hardware hold over standard software in areas such as image processing and pattern recognition [5].

1.2 Project Background

The HENN combines optical neuron connection layers with electronic signal processing in a parallel manner. The basis for this project lies in the theory and research for a Compact Optoelectronic Neural Coprocessor (CONCOP)[1]. It was further fleshed out by the first attempts at fabricating optoelectronic components for the CONCOP [6]. The HENN is a prototype precursor to the Compact Optoelectronic Integrated Neuroprocessor (COIN), the evolved successor of the CONCOP. The HENN aims to provide valuable guidance on system design, highlighting problems that need to be solved for the implementation of the COIN. It also provides crucial experience with current and new training algorithms. Ultimately, the goal is to create a fully-functioning prototype that can be a proof of concept. This proof of concept will hopefully generate industry and scientific interest in the COIN project and inspire fabrication and function research of a more industry viable product. The group's current neural system is bulky, immobile, and sequential, as it is the combination of a PC, neuron fiber-optic interconnection plate, and low-grade web camera. The final version of the new HENN system should be somewhat mobile, completely parallel, and capable of performing simple facial recognition, in order to be considered successful. If this is achieved, the prototype will also open the door for the scientific community at large, as a majority of the work in this field has been on the theoretical and algorithms side, as opposed to the physical.

Chapter 2

The HENN System Architecture

2.1 Defining the HENN System

Unit Architecture

A fully operational HENN system is based directly on the multi-layer neural network model. It is comprised of multiple unit architectures that represent the hidden layers plus separate architectures for the input and output neuron layers. The first step to understanding this, is reviewing the HENN hardware representation of an artificial neuron hidden layer that is part of a larger multi-layer neural network. This single neuron layer is broken down into five segments: weighting, synapses, summing, thresholding, and outputs and the subsystems that comprise the HENN unit architecture cover one or more of these segments. The list below provides a quick overview of each neuron layer element and the corresponding HENN subsystem

1. Weighting - LCD without backlight
2. Synapses - Fiber optic interconnection plate
3. Summing - BOD photodetector array
4. Thresholding - BOD comparator array
5. Output - BOD LED array

The neuron layer weighting is represented by an LCD with no backlight (essentially a spatial light modulator) and the synapses are physically manifested by a fiber optic interconnection plate. The last three segments are covered by the bistable optical

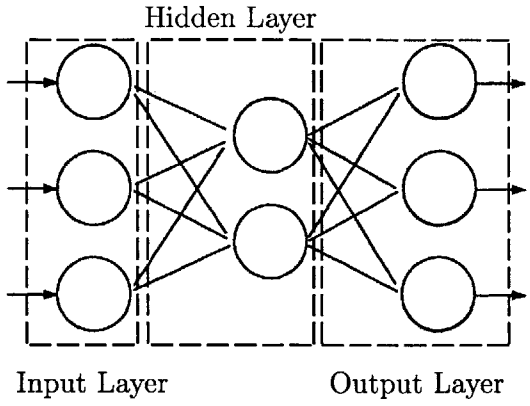

Figure 2-1: This three layer neural network outlines the different partitions of the neural network for each type of layer. The partitions serve as a basis for which HENN subsystems are used for each layer.

device (BOD). The BOD is an optical switch that is composed of a photodetector array, comparator array, and LED array. The summing of weighted inputs is performed by a phototodetector array. Intensity of the light that has propagated through the LCD and fiber optic interconnection plate sums on the surface of each photodetector. The currents resulting from the measured intensities are thresholded by an array of switch circuits. Each switch either completes the circuit and illuminates a single LED within the neuron layer output LED array or fails to reach threshold and turns its corresponding LED off.

Input and Output Neuron Layer Architectures

With the hidden layer architecture explained, the next step is defining the input and output neuron layers. The input and output layers, in terms of neurons, are best visualized by Figure 2-1. The figure also properly describes the hidden layer as well, and can be used as reference. The input layer simply forwards the input values to the hidden layers and the output layer takes the hidden layer outputs and forwards them out of the system. The output neuron layer is simply a hidden layer that also captures the output image at the end. The most likely candidate for that is a charge coupled device (CCD) camera that records the image and then sends it to the microcontroller

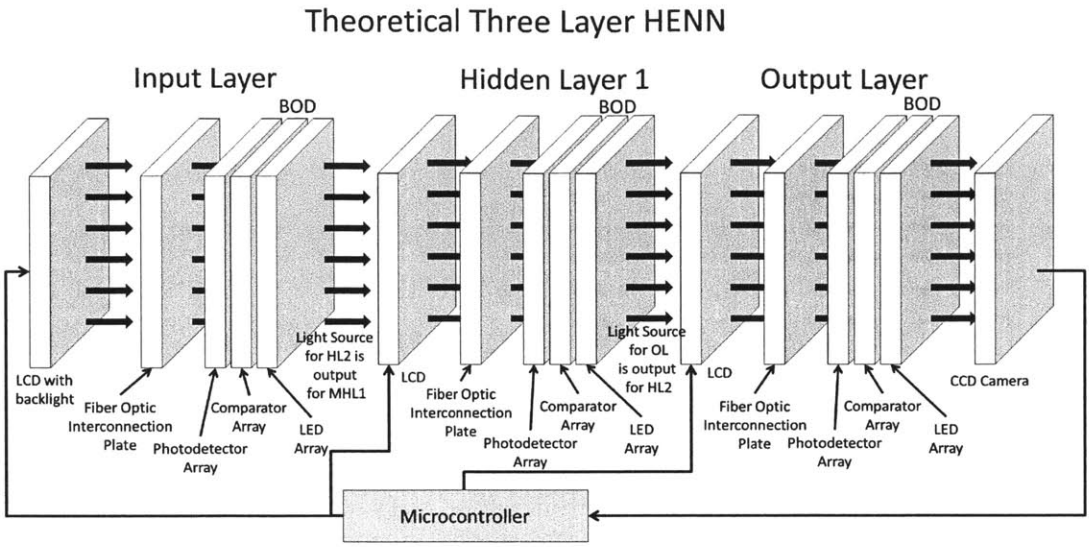

Figure 2-2: Diagram of the theoretical four layer HENN. The black arrows represent propagating light and the layers without light communicate electrically.

or computer that helps run the HENN system. The output neuron layer concept will not be explored beyond this as it is not pertinent to this thesis. On the other hand, the input neuron layer, while much simpler, is more critical to this thesis. The input neuron layer itself is just a regular unit architecture layer except it has an LCD with backlight. The input layer receives a weighting image from the master microntroller or computer and then displays that image on the screen.

The input neuron layer is especially important because it is one of the systems on which this thesis focuses. The thesis analyzes an LCD with a backlight and characterizes its attributes. Once that is completed, it determines how well it integrates with the fiber plate. Lastly, a program that displays the connection weights on the LCD is implemented in the thesis. All three of these sections are crucial to the function of the input neuron layer, which is exactly why one of the points of this thesis is to show how an LCD can be used as an input neuron layer.

Operation Overview of a Theoretical Four Layer HENN

Figure 2-2 depicts the complete HENN architecture for a four layer system:

1. Input Neuron Layer

2. Hidden Neuron Layer 1
3. Output Neuron Layer

For this description of operation it is assumed that this is a one-iteration system. However, in actuality this is very far from the truth. While training, the system must iterate and adjust the weights during each training cycle until the desired output on CCD camera is achieved. The input weight image from the microcontroller or computer is written onto the Input Neuron Layer LCD with backlight. The light travels from the backlight through the LCD and is attenuated by the pixels. These pixels modulate the light for each neuron and are controlled by preset weights from the microcontroller. The fiber optic interconnection plate, in the Input Neuron layer, receives the attenuated light and transfers the weighted light to whichever the neurons the synapses have determined. This light intensity is received by the Input Neuron Layer BOD photodiode array, which translates the intensity of each neuron into a current to determine if the neuron's state is on or off. The current is changed to a voltage and sent through the BOD comparator/optical switch, which outputs either high or low voltage. The resulting voltage from each of the neuron's comparators is attached to the BOD output light source (LED Array), which generates an output based on the voltages. The iteration continues onto Hidden Neuron Layer 1, where the LED output light from the Input Neuron Layer is the light source that propagates through the LCD. The LCD weighting is uploaded by the microcontroller and depends on the system training. The modulated light from the LCD couples into the fiber optic interconnection plate. There the HENN operates in the remaining part of the Output Neuron Layer as it did in the Input Neuron Layer. Once the light reaches the BOD output LED array in the Output Neuron Layer, the CCD camera captures the image and sends it to the microcontroller.

The following sections fully explain each subsystem of the HENN unit architecture and the HENN weighting and training.

2.2 Fiber Optic Interconnection Plate

The fiber optic interconnection plate, hereafter known as the fiber plate, is the crux of the project because it allows the system computation to be determined by the simple weighting and summing of light sent through numerous interconnections, rather than a processor. Each neuron acts in parallel with the others and allows for faster computation with little to no power consumption. As the number of neurons increases, so does the parallelization and computing power. The fiber plate connects two layers of neurons with optical fibers. The first neuron layer is a 17 x 17 array of neuron nodes (often referred to as just nodes or neurons), which are groupings of individual fibers. Figure 2-3 provides an excellent illustration of the layout of the fiber plate's first layer. Each inner neuron is a group of nine D_{Fiber} diameter fibers that are spaced $L_{Interfiber}$ apart center to center. The outer neurons are groups of 2 x 3 with the exception of the corners with are 2 x 2, both have the same spacings as the inner neurons. The reason these edge neurons have less fibers is from the nearest neighbor connection model, which is discussed later in the section. However, the basic idea is the edge neurons have less than nine neighbors, they only have 6 or 4. There is also a set $L_{Internode}$ distance from the origin of the center fiber in one neuron, to the origin of the center fiber in the adjacent neuron. The total length of the active fiber plate area is $L_{Array} = (Nodes - 1) \cdot L_{Internode} + D_{Fiber}$. The distance from the edge of first fiber plate layer to the center of the first fiber hole is L_{Indent}.

The second layer of the fiber plate has the same dimensions and indent as the first, but with a slight node difference. It still has 17 x 17 nodes, each separated by the distance $L_{Internode}$, however, each node is one larger hole of diameter D_{Node} rather than a square grouping of 3 x 3. Each second layer neuron hole is centered at the origin of the center fiber in the equivalent first layer node. The nine fibers for each second layer node are all situated inside this larger neuron hole, this hole will be positioned right above the detctor which will sum the light coming from the fibers. Figure 2-4 depicts the layout of the second fiber plate layer and the positioning of the fibers in a single neuron node.

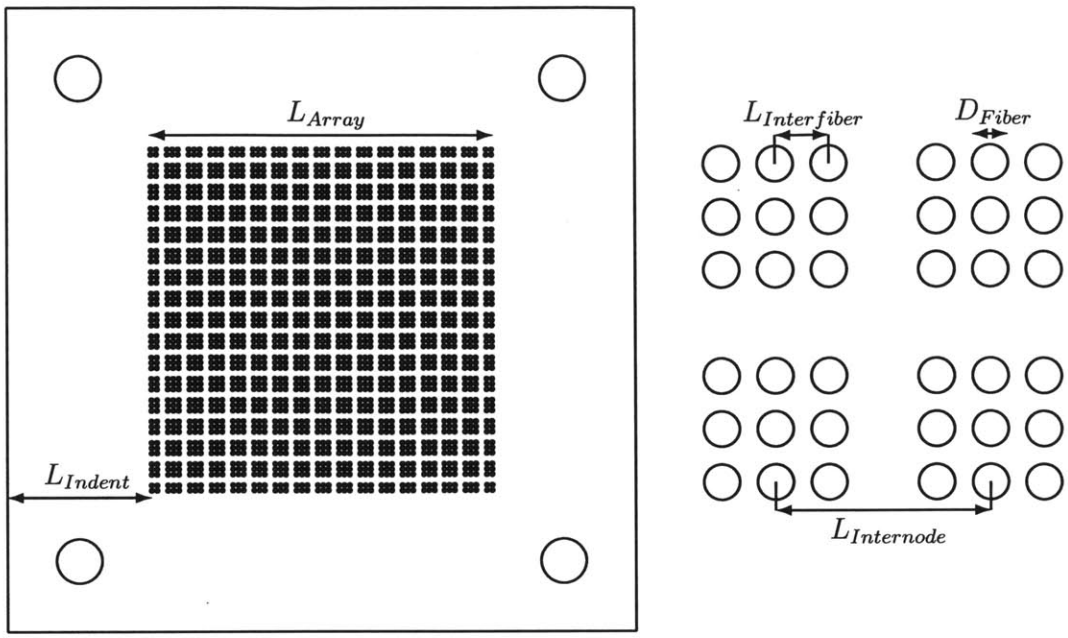

Figure 2-3: Left: Scaled 17 x 17 node array fiber plate first layer. Right: Zoomed in 2 x 2 node array to illustrate sizing.

The fiber type used for interconnections is an unjacketed plastic optical fiber that has a double concentric structure consisting of a polymethylmethcrylate (PMMA) core and a fluorinated polymer cladding. The high core refractive index and low cladding refractive index means the fiber employs total internal reflection to propagate signals. The core and cladding diameters from the manufacturer are listed in Table 2.1.

Diameter	Layer	Avg [μm]	Min [μm]	Max [μm]
.5 mm	core	485	455	515
	cladding	500	470	530
.75 mm	core	735	690	780
	cladding	750	705	795
1 mm	core	980	920	1040
	cladding	1000	940	1060

Table 2.1: Dimensions for three different fiber diameters.

In order to connect the two layers, the design employs nearest neighbor connections for the fibers of each neuron. Each fiber in a node of the first layer would connect

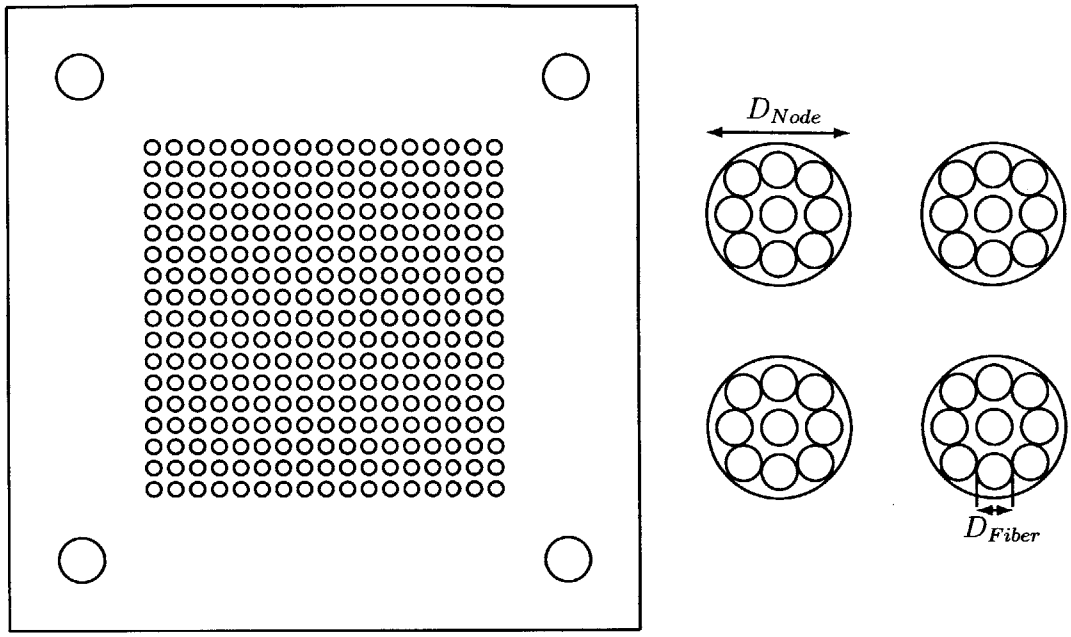

Figure 2-4: Left: Scaled 17 x 17 node array fiber plate second layer. Right: Zoomed in 2 x 2 node array to illustrate sizing and ideal positioning of fibers.

to one adjacent node in the second layer. Those nodes are determined by using the nearest neighbor principle. The central hole in the layer one node connects to the equivalent node in layer two. The other eight holes in the layer one node, connect to the eight unique nodes closest to the central layer two node. A diagram illustrating this concept is shown in Figure 2-5. This unique connection model allows information to propagate perpendicular to the overall system flow. The values for the current machined fiber plate are in Table 2.2.

Measurement	Value
Dimensions	88 x 88 mm
Nodes	17 x 17
D_{Fiber}	.5 mm
$L_{Interfiber}$	.75 mm
$L_{Internode}$	3 mm
L_{Array}	48.5 mm
L_{Indent}	20 mm
D_{Node}	2 mm

Table 2.2: Dimensions for current machined fiber plate.

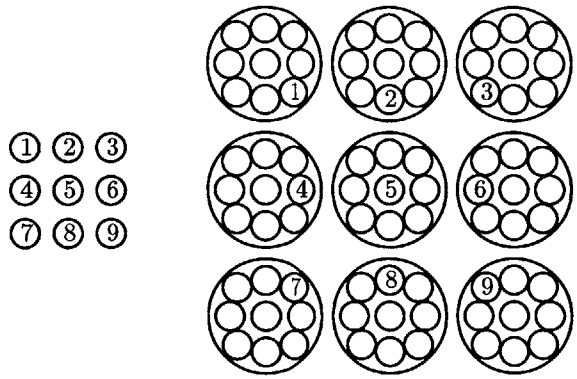

Figure 2-5: **Left: One node on the input side. Right: Nine nodes on the output side.** Diagram for the Nearest Neighbor connection model employed in the HENN fiber plate architecture. The left grouping is a single node in the first/input layer of the fiber plate and the right grouping is the nine nearest nodes in layer 2 to the single node in layer one. Each smaller circle represents a single fiber and circles with the same number are the beginning and end of the same fiber.

The fiber plate operates by each node in the first layer receiving attenuated light from the corresponding LCD pixel area. This light then propagates through the fibers via nearest neighbor connections to the nodes in the second layer.

2.3 LCD as a Spatial Light Modulator

The LCDs in the hidden layers, function as connection weights that modulate the light input to the fiber optic interconnection plates. The light traveling from the source passes through the LCD screen on its way to the fiber plate input. If the pixels on the screen are darkened, then the light signal is attenuated after passing through the LCD. In this way, the connections are weighted. As stated in the previous section, the fiber plate input layer has 17 x 17 nodes that are each made of 3 x 3 groupings of fibers. In the HENN project, the weightings are per fiber, so each individual fiber has its own weight. Since each fiber has its own weight and shaded pixels are the source of that weight, then specific pixels should correlate to a specific fiber. Multiple pixels are needed to fully weight an individual fiber and each should have an equal number of pixels. The number of pixels per fiber is based on the positioning of the fiber plate over the LCD and the size of the pixels.

In general, a more sensitive weighting scheme, allows for higher precision of image identification. The variation in intensity attenuation depends on both the number of unique pixel shades for the LCD and the amount of attenuation achieved by each value. For more information on the weighting refer to Section 2.5.

Note that this is not the system tested by this thesis. The experiments were conducted with the backlight, therefore, the conclusions may not be applicable to a spatial light modulator. The light source for the SLM is the output of the bistable optical device. Therefore, as previously explained, the input light has some lights on and others off, compared to the constant on state of the backlight. While uniformity allows for easier characterization and measurement of the LCD capabilities, the results are inconclusive for determining the SLM functionality. However, the characteristics of an LCD serving as a weighting system or input neuron layer are explored in much greater detail starting in Chapter 3.

2.4 Bistable Optical Device

The Bistable Optical Device (BOD) is one of the most challenging segments as an inexpensive reliable electro-optical switch has never been characterized. Therefore, this document will only briefly touch upon one of the possible designs. The BOD is the physical structure for the intensity summing, voltage thresholding, and hidden layer output light source. It is composed of a photo-sensitive element, a transistor switch, and an LED. The BOD's purpose is as an optical switch that will only illuminate an output node if the intensity of the input node is high enough. Figure 2-6 provides a potential circuit schematic candidate for the individual element that will make up the larger array of switches, with one switch per neural node. Using this potential schematic the BOD array would be a photoresistor array, thresholding circuitry array, and a LED array.

For an individual element, the light from the second fiber plate layer creates an intensity pattern on the photoresistor. This incident intensity correlates to a specific resistance, which results in a voltage divider dependent on incident light intensity.

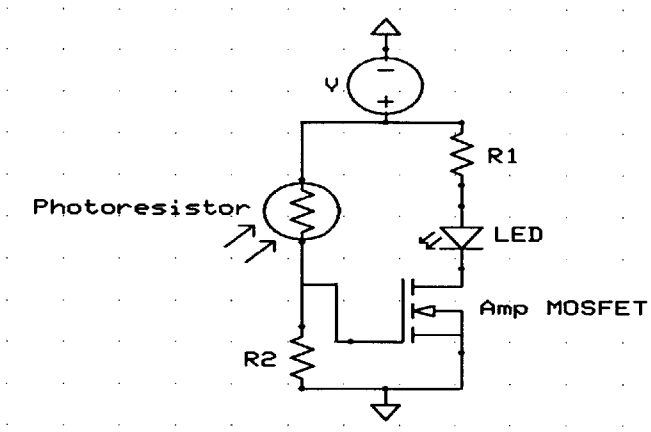

Figure 2-6: Potential BOD Circuit with Photoresistor.

The output voltage of this divider sets the gate voltage of a MOSFET and, if that voltage is high enough, then the switch closes and illuminates the LED. BOD research is still ongoing in the group and the final implementation is undecided.

BOD Considerations

The main concern about this layer is the number of discrete components. The plan is to create an array of BODs, one for each neuron, so that the parallel processing can be maintained. The goal is also to create a prototype that is relatively small and compact, therefore, the discrete component array may be troublesome. Fortunately, the problem can be solved with integrated circuits, however, that is a more expensive alternative to be explored only if necessary. When the photodetector array is flush with the fiber plate, it is important to note the possibility that an individual photodetector element may measure light leakage from neighboring nodes. A preventative measure such as a plastic isolator that fits between the fiber plate and photodetector array, can separate the nodes and completely prevent neighbor light leakage. However, the specific material would require in depth investigation to determine the effects of light absorption and reflection on the perceived intensity by the photodetector. Another concern is the photosensor's range of measurable intensity. Extremely high incident intensities can saturate the sensor, reducing the effective operation range. Similarly, extremely low intensities, may not register on the sensor, also reducing the effective

operation range. Therefore, choosing the proper photosensor is critical for proper operation.

2.5 Training Algorithm and Weighting System

Ideally, the training algorithm will operate in situ or on the HENN hardware. This means that the weighting image is created by the training and the image is generated iteration by iteration. A major benefit to this type of training is that it can accommodate for inherent system flaws and inefficiencies, because the training algorithm is operating on the actual network hardware. One example is misalignment of the LCD and fiber plate, the system considers that part of the training, factoring out any losses it might cause.

While current access to in situ training would be ideal for this project, the reality is that it is just a theoretical desire for the system. The actual HENN system in the lab is composed of an LCD, fiber plate, and a USB port for uploading the weights. The reality of the situation and the testing requirements temporarily forces the training and weights algorithm away from in situ and to a manual USB upload. What was once not even a concern for the in situ is now a problem. The only way to upload weights is to the determine their value and then map the 49 x 49 weight array to an image the size the fiber plate active area. This weighting image chapter thoroughly outlines the program to achieve this.

The current training algorithm is what enables the weighting system to exist. It is several different things, but the main idea is that it functions as a system to ensure that the neuron layer weights are properly applied to the fiber connections and that the light is weighted and efficiently propagates through the system to the next layer. Since the LCD tested by the thesis still contained the backlight, then the weighting system best applies to the input neuron layer. It is something that couples light into fibers efficiently, has minimal transmittance loss, is used to display a weighting image on, and has a small amount of variation between fiber light intensities. Those are the characteristics of a weighting system for the HENN architecture with the non-ideal

training algorithm. These are characteristics that are tested in the next chapters and are the source of proof for using an LCD as a weighting system at the input neuron layer.

Chapter 3

Characterizing the LCD

Now that the HENN project has been provided with context and the separate subsystems explained, the attention can shift to the LCD weighting subsystem where the majority of research was focused. Recalling the goals from the introduction, the thesis will provide the HENN with a usable LCD weighting system and suggestions to improve its integration with the HENN project. Development of a viable LCD weighting system requires several steps to meet the reliability and integration required by the HENN. The first is the characterization of the LCD, a precursory step to set up guidelines for selecting a specific LCD. Once selected, the commercial LCD is tested to determine various screen properties. This chapter attempts to satisfy the LCD guidelines with a commercial LCD and perform a thorough analysis on the selected model. It will perform a system physical or practicality assessment, individual pixel analysis, and a spectrum profile measurement. The data gathered from these evaluations will then guide the experiments performed in later chapters.

3.1 LCD Specifications

One of the goals was to avoid buying a commercial spatial light modulator. Therefore, it was necessary to find an LCD screen that could be properly disassembled and used as a transmissive LCD filter with and without a backlight. The experiments for this thesis were done with the backlight in place (suitable for the HENN input layer), and

further experiments are required to draw any conclusions about the LCD as a SLM for the hidden layers. Choosing a screen size was based on the current fiber plate dimensions. The fiber plate connection fiber diameter dictated the screen resolution and active diplay area. With those two values the average pixel size could be calculated and used as a guidelines since more than one pixel per fiber area was required. Additionally, the system required the ability to effectively display weighting images and easily interface with a computer or microcontroller. To meet these demands, the Aluratek ADPF08SF digital picture frame was selected for the first LCD weighting system prototype. Its specifications are in Table 3.1.

LCD Specification	Value
Model	Aluratek ADPF08SF
Resolution	800 x 600 pixels
Active Screen Width	16.2 cm
Active Screen Height	12.1 cm
Average Pixel Size	.2025 mm x .2017 mm

Table 3.1: LCD Specifications

The Aluratek ADPF08SF is a digital picture frame with a plastic shell that holds a metal casing and a controller board that receives button, USB, and power input, which it then relays to the LCD and backlight. The metal casing contains a cold cathode fluorescent light (CCFL), two light diffusers, and the LCD screen secured by a machined piece of plastic. The LCD screen has a 16.4 x 12.3 cm surface with an active display area of 16.2 x 12.1 cm. This allows ample coverage of the 50 x 50 mm fiber plate active area and an average pixel size of less than the required .25 mm. While the ADPF08SF does not directly interface with a computer, it does have a USB port for memory sticks. Therefore, the appropriate weighting image may be loaded onto the memory stick and then inserted into the picture frame, immediately displaying the stored image. The metal casing can be pried open and the CCFL backlight removed, an example of the separated LCD panel is shown in Figure 3-1. This allows the LCD to act as the first HENN light source/weighting filter with the backlight or as a thin weighting filter between an LED array and a fiber plate in the

middle stages of the system.

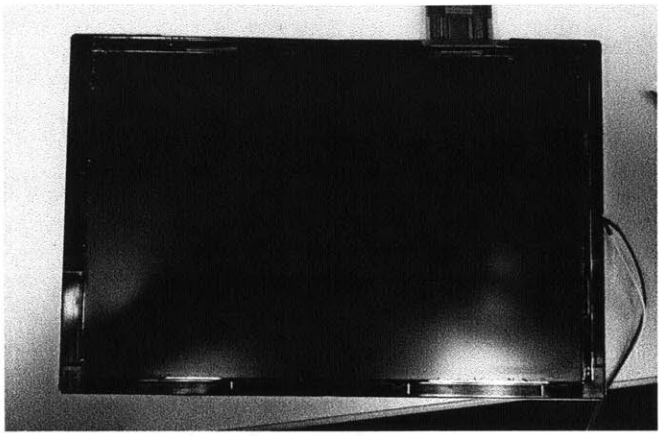

Figure 3-1: Separated LCD panel from the Aluratek ADPF08SF digital picture frame.

For placement between an LED array and fiber plate, the LCD panel must exist with no backlight. The removed backlight leaves a large amount of space within the metal casing, but the machined plastic piece still holds the LCD panel in place without extra reinforcement. To create the desired transmissive spatial light modulator, there must be a hole in the back of the metal casing. Ideally, the LED array would fit into the backlight hole. The adaptability of the Aluratek system is one of the main reasons it successfully met the requirements. Its systems were simple and functional, which provided room for the further modification required by future experiments.

3.2 LCD Pixel Analysis

Liquid crystal displays use the RGB color model to display color. The pixels that compose the screen are further broken down into three sub-pixels with color filters, red, green, and blue. Up close, the individual sub-pixels are discernible, but at a viewing distance, the separate sources are indistinguishable. The images displayed on the LCD can be represented as an array of pixels when stored in memory. Using this representation, pixels are structured similarly to their physical counterparts, a grouping of three binary values ranging from 0 to 255 base ten. Each value represents the amount of red, blue, or green present in the pixel color. Grayscale values occur

when all three values are equivalent, starting with black (0,0,0) and ending with white (255,255,255). There is also area between the pixels themselves, called dead space. It does not contribute any intensity, but affects the intensity input to any smaller object. To measure the pixels and this dead space, the LCD screen was placed under a microscope and sized according to the eyepiece reticule units. Figure 3-2 is two images of different display patterns captured by the microscope.

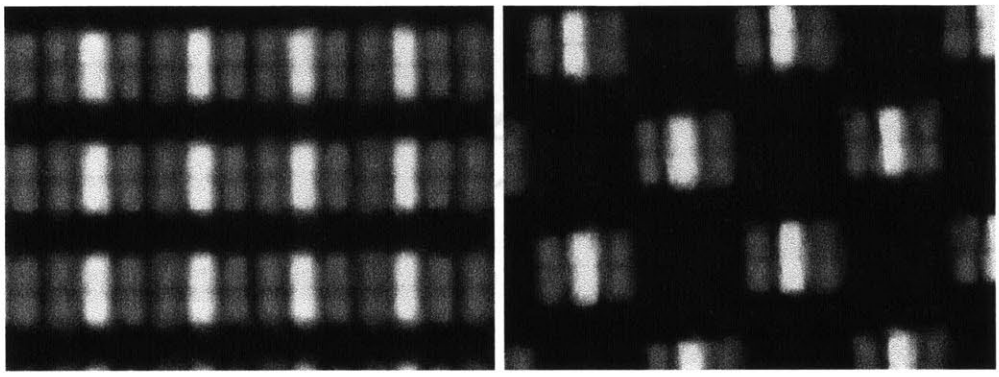

Figure 3-2: Two microscope pictures of the LCD pixels from the Aluratek ADPF08SF. The left is when all pixels are on and white, while the right is a black and white checkerboard pattern.

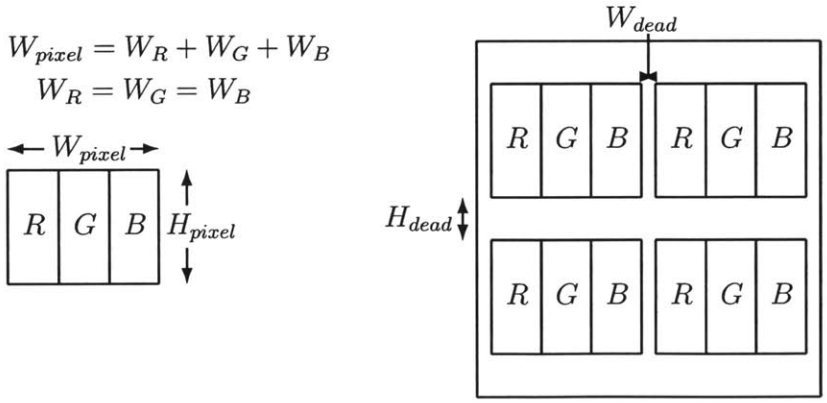

Figure 3-3: On the left: the Red Green Blue composition and sizing of a single LCD pixel. On the right: Sizing of the dead spaces between pixels for the chosen LCD screen.

In order to find the metric equivalents to the measured unit values in Table 3.2, the pixel spacing must be determined. Using the pixel layout illustrated in Figure 3-3

Measurement Name	Variable	Value [Units]	Value [μm]
Pixel Width	W_{pixel}	10.25	184
Pixel Height	H_{pixel}	7.75	139
Sub-Pixel Width	$W_R = W_G = W_B$	3.41667	61.333
Dead Width	W_{dead}	1.0	18
Dead Height	H_{dead}	3.5	63

Table 3.2: Pixel Size and Spacing Measurement Data

and assuming that there is no dead space between the edge of the screen and the edge pixels, a single row can be described by the summation of pixel widths and dead spaces. The same applies for a single column.

$$\begin{aligned} L_{row} &= 800 \cdot W_{pixel} + 799 \cdot W_{dead} = 162\, mm \\ L_{col} &= 600 \cdot H_{pixel} + 599 \cdot H_{dead} = 123\, mm \end{aligned} \tag{3.1}$$

With these equations and the ratios of dead space and pixels to total units, the value for a single unit can be calculated. After performing the computations, the value for 1 $Unit$ is:

$$\boxed{1\, Unit \equiv .0179\, mm} \tag{3.2}$$

With the result of Equation 3.2, the unit values in Table 3.2 may be transformed to metric values.

3.3 LCD Spectrographic Analysis

In order to measure the absolute intensity of a specific light source, it is necessary to know its wavelength intensity profile. The intensity $E_e(\lambda)$ of a single frequency light wave incident on a given surface (also called irradiance) is described approximately by Equation 3.3, where E is the complex electric field amplitude, $n(\lambda)$ is the refractive index of a medium based on a given wavelength, c_0 is the speed of light in a vacuum,

and ϵ_0 is the vacuum permittivity [7].

$$E_e(\lambda) \approx \frac{n(\lambda)\, c_0 \epsilon_0}{2} |E|^2 \tag{3.3}$$

While the most precise answer would involve calculating the LCD output irradiance for each wavelength, equipment and time constraints eliminate the feasibility of this approach. Fortunately, most light sources can be characterized by a small number of peak intensity wavelengths with narrow bandwidths. An accurate estimate of intensity can be calculated by summing the intensity values of these peak wavelengths.

The wavelength distribution for the LCD is determined by the CCFL spectrum and the wavelength absorption of the sub-pixel color filters. The wavelength intensity profile is measured with a grating spectrometer, which passes light from the source through the diffraction grating, which splits and diffracts light, which is captured by a digital camera. The digital camera image is parsed into a MATLAB program that measures the relative intensity of light for each frequency and creates a wavelength intensity profile.

Based on the pixel information in Section 3.2, the color white is created when all three sub-pixels are fully illuminated. This suggests that the white intensity profile should be the sum of the blue, green, and red intensity profiles. However, this does not help determine the wavelength or intensity contributed by each color, since the sub-pixel blue may contain green wavelengths and vice versa. The best method is to illuminate and measure each sub-pixel separately and then all together as white. So for each color, the LCD screen was illuminated with only one of the three sub-pixels, except for white which required all three.

The precision required for obtaining the peak wavelengths for each color and white was not very high. The purpose was to obtain average wavelengths and relative intensities. The relative intensities helped determine the relative intensity distribution for each color in white and the average wavelengths were used to find the absolute power of each color using another instrument. The spectrometer system had to be

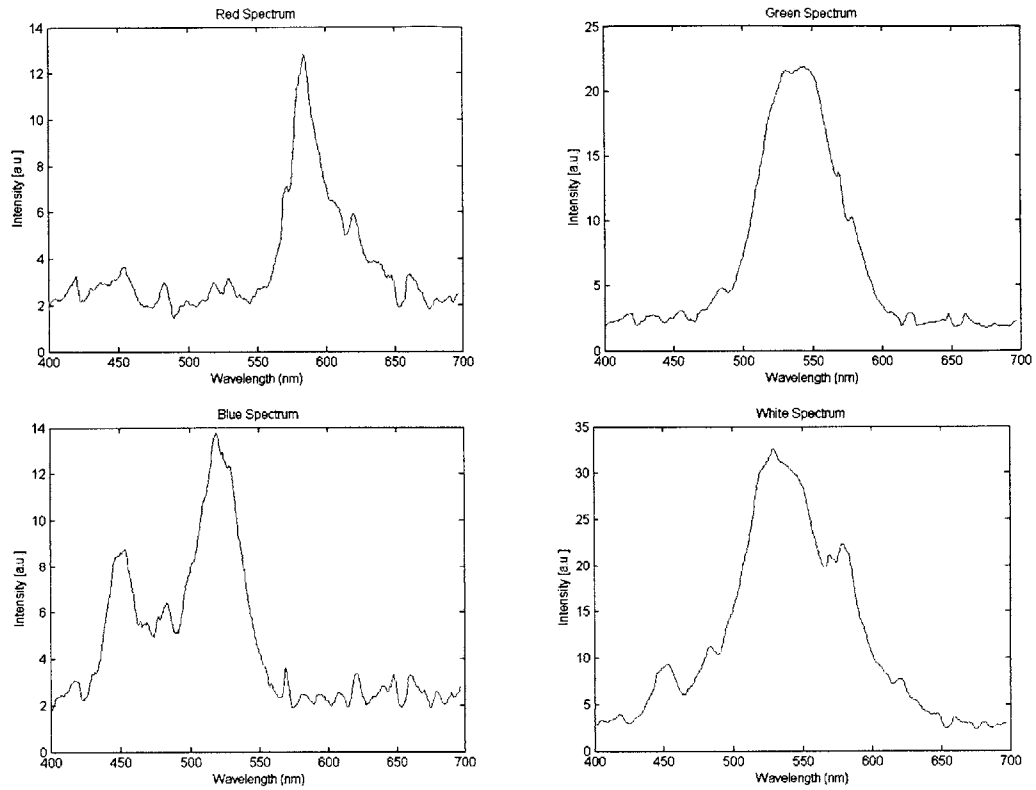

Figure 3-4: Spectra of the LCD screen when all pixels are Red, Green, Blue, or White.

calibrated with a fluorescent light to determine the location of specific colors and had a resolution of .41 nm/pixel and an experimental error of ±4.509 nm. In order to validate these results, it is assumed that the web camera is approximately equi-sensitive to all experimentally measured wavelengths. Additionally, the input intensity of the measured LCD screen is assumed sufficient for proper spectrum calculation and calibration. The average wavelength for red and green are easily determined by examining their peaks in Figure 3-4. For blue, the selected wavelength is a weighted average of the two peak wavelengths to simplify calculations. The two peaks for blue are 453 nm and 519 nm. The relative intensities RI_1 and RI_2 are 8.75 and 13.76 respectively. $\lambda = \frac{\lambda_1 \cdot RI_1 + \lambda_2 \cdot RI_2}{RI_1 + RI_2}$, provides the weighted average wavelength for blue. The selected average wavelengths for each color are listed in Table 3.3.

Color	Selected Wavelength λ [nm]
Red	584
Green	537
Blue	493

Table 3.3: Selected Wavelengths for each color on the LCD

3.4 Absolute Intensity and Pixel RGB Values

The spectrometer provided the relative intensity in arbitrary units, however, to determine the light coupling efficiency from the LCD to the fibers, fiber losses, and the intensity of a single pixel, the absolute intensity is required. The absolute intensity can be measured using a silicon power sensor and the LCD screen. The sensor operates by generating a specific current proportional to the incident intensity. Since the sensor cannot determine the intensity wavelength, the intensity-current proportionality constant varies with the wavelength. Before the measurement is taken, the assumed wavelength is input, selecting the constant and, therefore, determining the absolute intensity. It was for this reason that the spectrum analysis was performed and the average wavelength per color, shown in Table 3.3, was selected. For the measurement, an apparatus was constructed to secure the LCD screen position and prevent movement. To account the illumination of the entire LCD for each data point, a piece of black construction paper with a 20 x 20 mm square cut out was taped onto the LCD. Then the power sensor origin was aligned with the center of the square, placed flush against LCD/paper and locked into place. The square cut out was slightly larger than the 1 cm^2 active area of the circular sensor. This ensured that any measurement of intensity was in terms of $\mu W/cm^2$, which can be used to calculate incident intensity for other input areas. To determine the variation of each color's intensity over the entire range of 0 to 255 with acceptable precision, a data point was collected every eight values, so R, G, or B = [0,8,16,...,240,248] and then also at the value maximum, 255. The resulting intensity measurements and their trends are displayed in Figure 3-5. As a precaution, the environmental intensity (intensity of LCD with backlight off) was measured at 0 picoWatts. However, the power

sensor has a resolution of 100 pW, so it was guaranteed less than that and therefore negligible to this measurement.

Color	λ [nm]	Abs I_{max} $[\mu W/cm^2]$
Red	584	7.077
Green	537	10.21
Blue	493	9.285

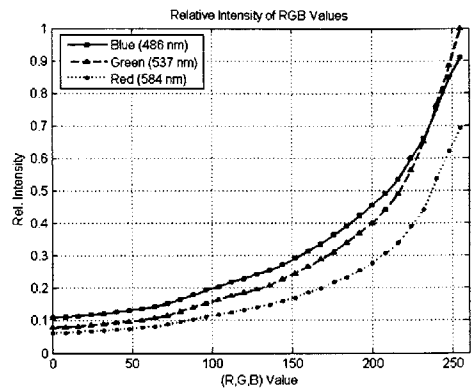

Figure 3-5: Peak Wavelength for each color on the LCD

The results show a clear trend correlating to increasing intensity with increasing R,G, or B value. More specifically, the change in intensity ΔI increases as the R, G, or B value increases, creating a nonlinear relationship. Despite the color specificity of the data, it is reasonable to infer the same relationship for mixed RGB values as well, specifically, the gray scale. However, the accuracy of this exact nonlinear relationship is only guaranteed for the backlight LCD combination, and may not be a valid observation in all cases. The experiment was performed using a backlight and LCD that were fabricated to function as a single unit, compensating for the others weaknesses. Despite this, generally, it is safe to assume that for any light source, varying the RGB value, will produce a result where transmitted intensity is directly proportional to RGB value. This attenuation can be attributed wholly to the LCD as the backlight illuminates the entire LCD with a uniform intensity. If this were not the case, then it be impossible to display multiple color shades on the screen at a time.

The key implication from these results is the validation of using an LCD to modulate the light intensity for the input neuron layer. Spatial light modulators attenuate the light that propagates through them and the LCD could not act as one if it was unable to do this. The LCD generates a range of unique intensities with corresponding RGB values adequate for neural network weighting. With this evidence, it is

feasible to imagine a weighting image that applies different attenuation to different RGB values and is mapped to align with the fiber plate input fibers. This software based weighting image would turn the LCD into a reprogrammable and size adaptable connection weighting system. The software development is discussed further in Chapter 5. The only caveat that must be considered for neural network adequacy is selecting a photoelement with the correct measurement range and sensitivity. This validation of neural network adequacy applies to any combination of LCD and light source, but the relationship plotted in Figure 3-5 does not. More investigation is needed to construct the exact RGB relations for other lighting sources, such as LEDs and OLEDs.

Chapter 4

Interfacing the Fiber Plate and LCD

The previous chapter on LCD Characterization performed a base characterization and investigation of the LCD system and how it functions. These preliminary actions were necessary in order to determine the proper experiments for more in depth characterization and to discover specifications needed to perform more experiments. Comprehensive characterization of the LCD screen for the purposes of the HENN project required several intensity tests be conducted. This chapter will cover each of the experiments in detail, including their purpose, procedure, results, and impact on the HENN's evolution.

4.1 Position-based Variation of Fiber Input Light Intensity

Purpose

One of the largest concerns with employing an LCD screen was the existence of dead spaces between the rows and columns of pixels. Due to the small fiber diameter used for the fiber plate interconnects, it was suggested that the positioning of the fiber over the LCD would have a non-negligible effect on input intensity. Therefore, it was

necessary to determine the single fiber intensity variation created by different fiber area and pixel/dead space overlap. For ideal neural network functionality, the input intensity should be uniform across all fiber input holes. However, that was not possible with the current fiber plate, since the inter-fiber distance $L_{Interfiber}$ and inter-node distance $L_{Internode}$ were not multiples of the distance between two pixels. The results from this test would provide the maximum intensity variation expected between two fiber's information and the optimal plate placement to minimize variation between different fibers on the plate. Also, it would reveal the alignment required for the maximal input intensity of a single fiber.

Setup and Method

To test this variation, the pixel measurements were mapped to a matrix in MATLAB that represents an 8 x 8 pixel array. Then using a circular logic mask, an equivalently dimensioned matrix was generated with a circle of diameter equal to the average core diameter of the specific fiber. The experiment was split into two groups: uniform intensity and varying intensity. The uniform intensity, or control group, had a pixel mask that assumed the intensity resulting from the red, green, and blue sub-pixels were all equal. The varying intensity group adjusted the relative values for each sub-pixel, based on the wavelength intensity profile of the light source. The relative intensity values for the red, green, and blue sub-pixels were gathered from the intensity versus RGB measurement table in Figure 3-5. The weightings for the colors are normalized to the value of green, i.e. green is 1.0, blue is 0.91, and red is 0.69. Within each group, three different fiber diameters were tested: .5 mm, .75 mm, and 1 mm. As discussed earlier, the fibers are composed of a cladding and a core, therefore the actual diameter that accepts light is smaller than the total diameter. The average core diameter from Table 2.1 was selected for each fiber size's circle mask diameter in the code. This simulation accurately represents the plausible input intensities for a system where the fiber end and LCD screen are perfectly flush with no air gap and zero light contribution from pixels or parts of pixels outside the fiber core diameter.

The method for testing was greatly simplified by the periodic nature of the LED

pixel pattern. There is only a small amount of variation in the possible coverages by a single fiber. Figure 4-1 illustrates the basic experiment for the uniform group. The RGB rectangles represent the pixels and the white in between is dead space. The black circle represents the fiber core to be shifted. The dashed rectangle and all the area inside of it, represents the all possible fiber origin locations for unique pixel arrangements. In fact, one unit pixel contains all the unique pixel arrangements, the reason for mapping over all four was to illustrate the intensity's periodic nature. The test itself, calculated the fiber mask at each point within the dashed rectangle and then multiplied the mask with the pixel array. The sum of all the elements in this new matrix was the relative intensity of that specific positioning. After iterating through all the positions, the intensities were all scaled relative to the maximum positional intensity value. The pixel sizing in the code was scaled to the actual pixel measurements in Table 3.2, with one matrix element being equal to one micron. The test assumed that this resolution was fine enough to ignore the shape error caused by the generation of a circle on a discrete square matrix. The discussion of exact code functionality along with the test code itself are available in Appendix Section A.2.

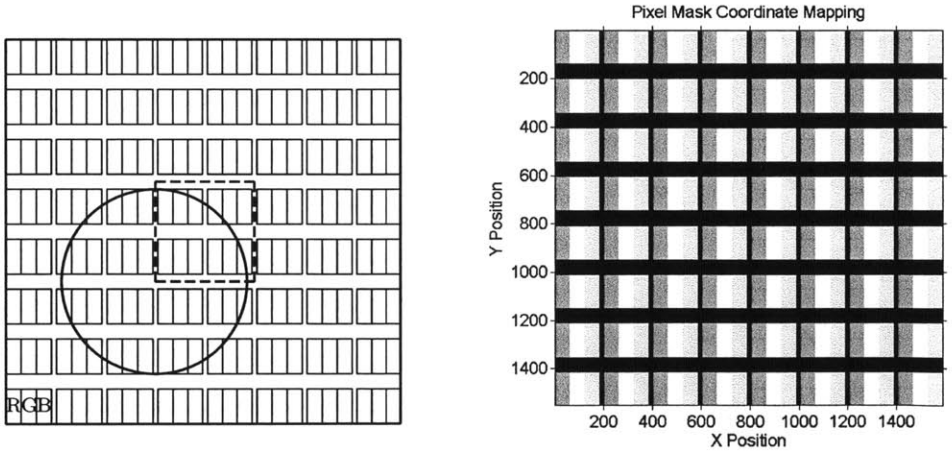

Figure 4-1: Left: Positional Intensity testing setup. The fiber core circle origin traces every point along and inside the 4 unit area dashed rectangle. At each point, the total intensity covered by the fiber core is calculated and mapped to that point. Right: Coordinate mapping of the 8 x 8 pixel mask used in the positional intensity variation calculation. The darker color corresponds to a lower the relative intensity value for both the dead space and RGB pixel.

Results

After all six tests had been conducted, the difference between the control group and varying intensity group was deemed negligible enough that only one group need be further analyzed. Therefore, the relative intensity group was chosen to field slightly more realistic results. Figures 4-2, 4-3, and 4-4 show the relative intensity resulting from placing the circle mask origin at each point inside the dashed rectangle. The visible periodic nature in all three figures reinforced the assumption that one fourth of the dashed rectangle (one unit area) was the extent of differing intensity positions.

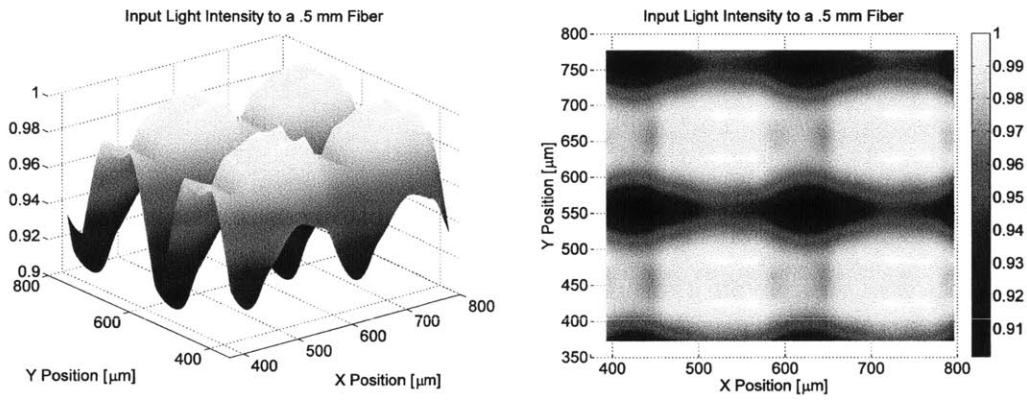

Figure 4-2: The position-based variation of total input light intensity for a .5 mm fiber when the pixel intensities are relatively scaled.

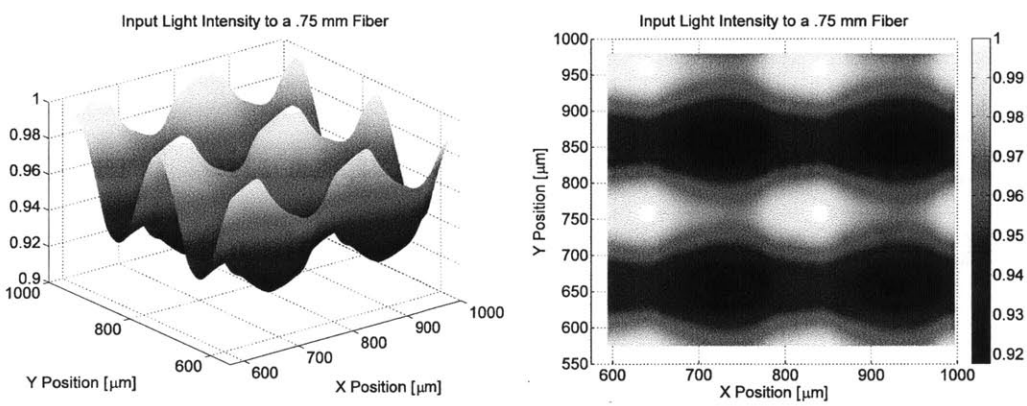

Figure 4-3: The position-based variation of total input light intensity for a .75 mm fiber when the pixel intensities are relatively scaled.

More specifically, the range of intensity variation decreased as the fiber core diameter

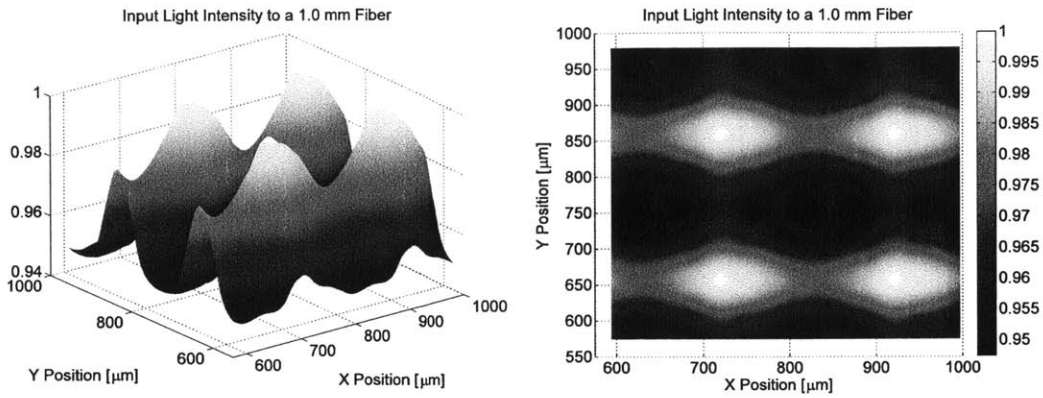

Figure 4-4: The position-based variation of total input light intensity for a 1.0 mm fiber when the pixel intensities are relatively scaled.

increased. The larger fiber area averaged out the dead space and intensity variation. The variation for the 1 mm was slightly over 5%, compared to the 10% variation for the .5 mm fiber. This small variation helps to prove the plausibility of the LCD as a spatial light modulator. Since the value of variation is so small, it would be reasonable to program it into the weighting algorithm.

The general location of the data maxima was inferred by mapping the 8 x 8 pixel array to the figure coordinate grids as illustrated by Figure 4-1. The exact maxima locations are unimportant because of the possible matrix resolution error and their heavy dependence on the relative intensity values, but the general area is consistent per fiber diameter. The horizontally asymmetrical sub-pixel intensities resulted in a shift of the maxima toward the blue sub-pixel. It is also essential to note the robustness for fiber placement, a concept that will become pertinent later on. For the .5 mm fiber, the allowance is 85 microns to remain within 2-3% of the maximum value. Logically, the allowance decreases to 75 microns and 2% for .75 mm fiber, but the 1 mm fiber still boasts a range of 50 microns with only a .5-1% intensity reduction.

Another value that commands attention is the period length of the mapping. To maximize each fiber's intensity, they must be positioned near the origin coordinates of the graph maximum values. However, every fiber cannot be placed near those coordinate because the inter-fiber $L_{Interfiber}$ and inter-node $L_{Internode}$ distances of the

fiber plate are not multiples of the period length. Therefore, the offset fiber positions guarantee both a non-uniform input intensity for all node array fibers and a suboptimal intensity value for the majority of fibers. The reason why this is critical will, again, be explained when redesigning the fiber plate in Section 6.2. Lastly, speaking to the practicality of the experiment, if the inter-fiber and inter-node distance are known, then the data from this graph can determine the input intensity for any fiber in the array regardless of the dimensions.

4.2 Single Fiber Loss

Purpose

The previous section's positional intensity variation test employed a MATLAB simulation to calculate minute changes in fiber input intensity depending on the LCD pixel layout. That mapping represented the potential total input intensity on the surface of a single fiber, but prevented inference on the light interaction with the fiber. The single fiber test focused on that interaction's loss, analyzing the coupling losses from the change of medium added with the intensity attenuation stemming from intra-fiber propagation.

Setup and Method

To experimentally determine the relative intensity attenuation for the three fiber diameters, three separate one fiber blocks were built. The single fiber testing block was a 5 x 5 x .5 cm black rectangular prism of Acrylonitrile butadiene styrene (ABS) with a smooth glossy face (L x W) and a textured matte face. At the facial origin was a drilled hole of diameter slightly larger than a fiber, through which an 10 inch fiber was threaded. The fiber was mounted in the hole with epoxy and the smooth face fiber end polished flush with the facial plane. The fiber extruded several inches from the textured facial plane terminating with a polished end. The fiber ends were polished to help increase the coupling efficiency and imitate the fiber state in the fiber plate. Additionally, it supports the inherent experimental assumption of a flush

interface between the LCD and fiber end and the opposite fiber end and power sensor. The LCD screen, which displayed a maximum intensity color screen, was covered by a piece of black construction paper with a 30 mm x 30 mm square aperture to eliminate any errant light from the screen. For the experiment, a fiber block was aligned over the illuminated square aperture and the fiber was placed flush with the front plane of the power meter. This was repeated for each color (R = 255, G = 255, and B = 255) and each fiber size. Since the purpose of this experiment was to analyze the power going in and out of the fiber only, the longer fiber was necessary. The size of the fiber hole drilled in the block, had to be slightly larger than the fiber to allow the epoxy room to surround the fiber and cement its position in the hole. Though most of the light coupled to the fiber core, a small percentage likely coupled with the epoxy. To prevent measuring any light that did not propagate through the fiber core, the fiber was elongated and curved to the side of the LCD. This ensured any non-fiber light dispersed to essentially zero if it somehow reached the power sensor.

Results

There are two distinct but related forms of attenuation in a fiber: gross attenuation and spectral attenuation. The gross attenuation focuses on the overall wide band losses that affect the transmission power. Spectral attenuation is the uneven distribution of losses across the signal spectrum. Depending on the fiber material and geometry, certain wavelengths are attenuated more than others. For this reason and the power sensor's inability to measure broad band power, the experiments were conducted assuming extremely small bandwidth. While the spectrum analyses in Figure 3-4, demonstrate the broadband nature of each LCD color, the spectrum analysis clearly states the intention to approximate each color with one wavelength. For this reason, the experiment's spectral attenuation is incorporated into the gross attenuation and becomes another factor that decreases the transmission efficiency. The transmission efficiency of an optical fiber measures the percent of light transmitted between input and output. It is calculated by the ratio of $\frac{I_{out}}{I_{in}}$, and is affected by the fiber's light coupling efficiency and the propagation attenuation caused by material

absorption and scattering. Unfortunately, due to the nature of the experiment, only the total attenuation can be determined. Instantiating the values contributed by each type of attenuation required other experiments and was not critical at this stage of the project. The output intensities were measured by the experiment and are listed in Table 4.1.

	Fiber Diameter								
	.5 mm			.75 mm			1.0 mm		
Wavelength	I_{in} [nW]	I_{out} [nW]	TE	I_{in} [nW]	I_{out} [nW]	TE	I_{in} [nW]	I_{out} [nW]	TE
584 nm	13.074	12.4	94.85%	30.027	22.5	74.93%	53.382	52.4	98.16%
537 nm	18.863	18.6	98.61%	43.320	31.4	72.48%	77.014	75.9	98.55%
493 nm	17.154	16.6	96.77%	39.396	27.9	70.82%	70.036	69.3	98.95%

Table 4.1: The calculated input intensity, measured output intensity,and calculated Transmission Efficiency for the single fiber intensity attenuation experiment. In this table $I_{in} \equiv I(0)$ and $I_{out} \equiv I(z)$, with the wavelength for each intensity determined by the table.

The input intensity can be calculated using the data from the absolute intensity versus RGB experiment in Figure 3-5. As stated in the current experiment's method, the LCD input image was set at full intensity for the chosen color and illuminated an area larger than the fiber input. Also, the fiber block was assumed flush with the LCD screen. Therefore, the fiber input was analogous to the power sensor in the absolute intensity versus RGB experiment. The maximum intensity at each wavelength per cm^2, multiplied by the average fiber core area in cm^2, calculated the input intensity for the fiber block: $I(0,\lambda) = I_{Max}(\lambda) \cdot \pi \left(\frac{D_{Core}}{2}\right)^2$. For this calculation the maximum intensity is assumed to be constant over the fiber core. The calculated input intensities for each fiber diameter are displayed in Table 4.1. With both input and output intensity determined, the transmission efficiency for each fiber and color can be calculated with a simple fraction. The values are listed in Table 4.1 with the input and output intensity.

The first positive sign was the relatively tight value grouping for fiber efficiency. Over a small length like 10 inches, the spectral attenuation should not have a drastic effect. Though the efficiencies for the 1 mm fiber were unexpectedly high, their values were not cause for concern. The intensities all measured in the nanoWatts range and the power sensor has a resolution of .1 nanoWatts and uncertainty of ±3% for this

spectral range. Therefore, the measurements could have up to ±4% error, however, that is still rather negligible for the worst case and should not affect the data. The .75 mm data showed cause for concern, since it was around 20% lower than the other two average efficiencies. Such a substantial difference is not the result of resolution error, but more likely a problem with the fiber itself. The attenuation affected all three wavelengths equally, again pointing to a fiber problem. It is unlikely that the fiber end polishing is too blame, since the other two fibers were not polished significantly more than the .75 mm. Another suggestion was related to the positional variation of the input light intensity to the fiber, but after examining Figure 4-3, the most it would account for is about 8%. Therefore, it is believed to be anomaly of the fiber, and would be debunked by testing another fiber with the same diameter. For this specific case, it is reasonable to extrapolate the results from the other two diameters and assume a similar transmission efficiency for the .75 mm diameter. Despite these positive test results, the best way to test the viability is comparison to an outside source. To facilitate the comparison, the transmission efficiency must be transformed to the standard attenuation in decibels per meter. An optical fiber's attenuation coefficient is expressed by the equation:

$$\alpha = -\frac{10}{z}\log\left(\frac{P(z)}{P(0)}\right)[dB/m] = -\frac{10}{z}\log\left(\frac{I(z)}{I(0)}\right)[dB/m] \tag{4.1}$$

where $P(0)$ and $I(0)$ are the power and intensity at the fiber input before coupling losses and $P(z)$ and $I(z)$ are the post coupling loss power and intensity for the fiber output that is a distance z from the input. Table 4.2 shows the attenuation coefficients for each fiber and wavelength. Normally this measurement does not take into account the coupling loss, but for this experiment it does.

According to the fiber vendor, the 1 mm fiber has a maximum attenuation of .2 dB/m and an average of .18 dB/m with 650 nm non-collimated light. This value does not include the coupling loss, therefore, will be much less than most values measured during this experiment. Based on the comparison of with the manufacturer, the loss measurements are of reasonable value. Most of the discrepancy can be resolved

	Attenuation Coeff. [dB/m] by Fiber Diameter		
Wavelength	.5 mm	.75 mm	1.0 mm
650 nm (Mfr)	-	-	.180
Red 584 nm	.905	4.934	.317
Green 537 nm	.240	5.502	.249
Blue 493 nm	.561	5.900	.181

Table 4.2: The calculated attenuation coefficient for each fiber and color. The first row is the average attenuation coefficient as measured by the fiber manufacturer (Mfr).

by blaming the coupling loss for the above average values. The end polishing can always be improved to reduce this error, but not eliminate it. The small sample set and power sensor error may also account for a small portion of the difference, as the experiment tested only one fiber per diameter. Taking all these caveats into consideration, the resulting measurements still present a decent image of the baseline fiber function, confirming the plausibility of use in this fiber plate and new iterations. The percent intensity lost through transmission is a manageable value even with the limited input intensity value. In fact, these values can be considered some of the worst case scenarios for transmission loss, as future iterations will only improve and reduce the losses.

4.3 Effects of Crosstalk in the Fiber Plate

Purpose

While the single fiber test was an excellent indicator for individual fiber performance, it cannot be considered representative of the fiber plate function. The addition of numerous fibers and smaller dimensions breeds new problems inherent to the fiber plate but absent from the single fiber block. The light leakage discussed in Section 4.2 is amplified by the fiber plate's compact nature, increasing the probability non-negligible amounts of errant light could couple to fibers or epoxy boundaries. Light from fibers also could couple to the output layer epoxy boundary. Of course, whenever there is a high concentration of optical fibers in a compact space, the existence of fiber crosstalk is a concern. In most systems, including the HENN, inadvertent cross-fiber

light coupling and light leakage coupling induce a fundamental system failure. Therefore, the fiber plate tests were compulsory to establish a baseline influence of these effects on fiber plate performance. The performance analysis consisted of two parts: single node illumination and intra-node fiber deviations. The single node component examined the output image created by the illuminated input nodes. This served two purposes: to determine whether the nearest neighbor model was maintained or if light leakage/crosstalk illuminated other output nodes and to calculate the average output intensity and attenuation for each fiber, when an entire node was illuminated. The intra-node portion attempted to assess the affects of differing fiber lengths and radial bending on fiber transmission efficiency within a single node. This aided in characterizing the attenuation fluctuation between the different connections. The gathered data for both sections would also aid the formulation of guidelines for future fiber plate designs.

Setup and Method

The experimental setup required the fiber plate, two pieces of black construction paper, black electrical tape, the LCD screen, and power sensor. The LCD screen was entirely illuminated with a maximal RGB color, depending on the wave length being tested. Obviously, because the fiber plate was a prefabricated structure, the only testable fiber diameter was .5 mm. In preparation for the experiment, all nodes in the first layer except four pseudo-randomly selected inner nodes were covered with black electrical tape that blocked any input light. A piece of black construction paper with a 30 mm x 30 mm square aperture was taped onto the LCD screen to eliminate errant light. The fiber plate test nodes were aligned within the aperture and then a second smaller piece of black construction paper with a circular aperture was placed on top of the second fiber plate layer. To conduct the experiment, the 2.2 mm aperture (slightly larger than D_{Node}) in the second piece of construction paper was aligned over one of the input nodes nearest neighbor nodes in layer two. The power sensor origin was then placed flush with the fiber plate and paper. The aperture and power sensor were repositioned to measure individually each of the nearest neighbor

for all four input nodes. Additionally, to ascertain light leakage and cross-talk, the experiment recorded the intensities for the next nearest neighbors. This provided a larger picture of the system reliability and efficiency. Since the input fibers will never be individually illuminated during neural network operation, the intra-node fiber deviations were analyzed from the same data.

Results

To preface the following experimental results, the purpose of this experiment was not to comprehensively characterize the fiber plate as that would require a uniform light source and many more experiments. The main idea was to construct a partial example of normal fiber plate operation within the HENN system using an LCD. For this reason, the following results can be classified more as reference for future fiber plate designs and as evidence for the feasibility of an LCD working with a fiber plate. When the experimentally modified fiber plate was placed onto the illuminated LCD, the first action was visually documenting the output image generated by the four illuminated input nodes. The resulting pattern was captured for reference and is displayed by Figure 4-5. Proper fiber plate operation dictated that an illuminated input node would yield one illuminated fiber in each of the nine nearest neighbor output nodes. Four non-adjacent input nodes should then translate to four separate groups of nine output nodes with one illuminated fiber. However, as evidenced by Figure 4-5, the upper left node group contained two fibers in the central node and zero in the bottom central node. While slightly unexpected, this was the consequence of incorrectly connecting the optical fibers and, after a quick examination of the connections, was deemed an ineffectual influence on the rest of the system. Ignoring the botched connection, a visual analysis was sufficient to confirm the actual output matched the expected output. There were no other unexpected fiber illuminations or irregular patterns. Obviously, visual confirmation would not be considered solid evidence, so the intensity for relevant nodes was measured as well.

The sixteen next nearest neighbor nodes were measured to assess if light leakage and cross talk were illuminating these fibers. The experimental result showed little to

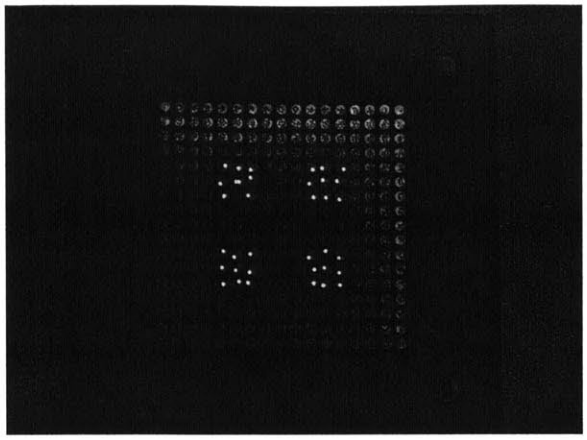

Figure 4-5: Fiber plate output when only four separate nodes are fully illuminated. Note the discovered wiring error in the upper left node output. The fiber meant for the bottom nearest neighbor was connected incorrectly, resulting in two fibers connected to the central nearest neighbor. The green dots are the illuminated fibers and the yellowish light is from a lamp used to light the room for the photograph. The green dots are the only fibers with visually perceptible lighting.

zero crosstalk. The measured intensity for every single next nearest neighbor for all four input nodes was zero. Due to the resolution constraints of the power sensor, it cannot be asserted as indisputable fact, but, since all sixty four next nearest neighbor nodes had an intensity value small enough not to be rounded to 100 picoWatts, it was reasonable to assume as a negligible amount or zero. Typically, fibers with a quality cladding have minimal to no light leakage, but the chosen fiber was plastic instead of silica glass. This result was strong confirmation of the fiber cladding quality and the fiber choice, despite its plastic composition. It is critical to note that this was evidence for no cross-talk or leakage when the magnitude of light was equivalent to one node. While it is sensible to assume it will hold true for adjacent multi-node illumination, the summation of the errant light might no longer be negligible. That being said, it would not be impossible to allow a variance of ± 100 pW in the weight training algorithm, but variation in the input intensity is extremely difficult to incorporate. It would have to be hard coded into the training algorithm, meaning it would require extensive testing to further analyze and positively confirm the variation and its value. The measured non-zero values for each node and each color are listed in Table 4-6.

After searching for trends in Table 4-7, the inconsistency for fiber values within

Red Output Intensity [nW]

8.5	8.6	6.4	10.4	11.8	11.2
11.8	15.0	11.4	6.4	11.6	10.5
10.9	0	10.0	11.6	10.6	11.0
8.9	9.2	7.9	8.5	8.2	6.1
8.8	4.6	9.0	8.5	4.6	9.4
8.5	7.9	10.2	9.8	9.3	7.2

Green Output Intensity [nW]

12.0	12.2	9.1	14.3	16.8	16.0
16.5	22.0	15.2	9.1	16.5	15.0
16.0	0	14.5	17.1	15.1	15.7
13.1	13.1	11.7	12.1	11.5	8.5
13.0	6.2	12.8	11.9	6.5	13.0
12.1	11.1	14.7	14.1	13.4	10.0

Blue Output Intensity [nW]

10.7	10.8	7.9	13.0	15.2	14.4
15.7	19.5	15.1	8.0	15.2	13.4
15.0	0	13.1	15.3	13.8	14.1
11.5	12.2	10.2	10.7	10.4	7.4
11.7	5.3	11.3	10.7	5.6	11.7
10.7	9.8	12.6	12.6	11.9	8.9

Figure 4-6: The measured output intensities for each of the nine nearest neighbors when four unique layer one nodes were illuminated with each color. Upper left table under each color corresponds to the upper left node in Figure 4-5.

nodes was obvious. For better perspective, the resulting intensities were used to calculate the transmission efficiency for each output node, with the assumption that each output node has an input of one illuminated fiber. The calculated values are in Table 4-7.

Red Transmission Efficiency

.65	.66	.49	.80	.90	.86
.90	1.15	.87	.49	.89	.80
.83	0	.76	.89	.81	.84
.68	.70	.60	.65	.63	.47
.67	.35	.69	.65	.35	.72
.65	.60	.78	.75	.71	.55

Green Transmission Efficiency

.64	.65	.48	.76	.89	.85
.87	1.17	.81	.48	.87	.80
.85	0	.77	.91	.80	.83
.69	.69	.62	.64	.61	.45
.69	.33	.68	.63	.34	.69
.64	.59	.78	.75	.71	.53

Blue Transmission Efficiency

.62	.63	.46	.76	.89	.84
.92	1.14	.88	.47	.89	.78
.87	0	.76	.89	.80	.82
.67	.71	.59	.62	.61	.43
.68	.31	.66	.62	.33	.68
.62	.57	.73	.73	.69	.52

Figure 4-7: The calculated transmission efficiencies for each of the nine nearest neighbors when four unique layer one nodes were illuminated with each color. Upper left table under each color corresponds to the upper left node in Figure 4-5.

The transmission efficiencies painted a much clearer picture than the absolute intensity. First off, the as was proven by the single fiber loss experiment, the transmission efficiency of each fiber was a similar value for all three wavelengths. The agreement in results lent credence to these fiber late measurements. Excluding the upper left node group, each node group has an average transmission efficiency range with the exception of one or two fibers. However, there was no consistency between different node groups for the same color. Furthermore, when comparing these efficiencies to the ones measured in the single fiber loss test, in Table 4.1, there was a significant drop in the transmission. The average values from the single fiber test were above 90% for a .5 mm fiber, which put the upper right node group about 10-15% below average and both bottom groups about 30-35% below average. These

were astounding differences that could not be accounted for by traditional error. The difference immediately inspired a return to the sub par average for the .75 mm fiber. However, where that was one fiber, this was twenty seven different fibers in three separate groupings. Knowing that a majority of the fiber loss stemmed from the light-fiber coupling, the fiber plate was examined for sufficient fiber polishing, and that was where the problem was discovered. The fiber plates had been properly polish, but under a microscope, it was obvious that many of the fiber ends had large indentations or chunks of fiber missing. After examining the fibers on the input and output, it was determined that the amount of extra loss correlated to the quality of fiber end. The upper central fiber in the upper right node group had slight indentations on both ends and a transmittance of 0.90, compared to the central fibers of the bottom nodes, which both had at least a fourth of the fiber end missing on one or more ends and a transmittance 0.30. While the exact fiber end indention data was not formally recorded, it can be assumed as the cause for these extremely sub par transmittances. The sole remedy for this difficulty is redesigning the fiber plate. In addition, the construction methods employed must be outlined so as to avoid ruining future iterations, something to be discussed in depth in Section 6.2.

These results are significant for more than just the fiber plate, they have intrinsic implications for the LCD weighting system as well. The successful measurement of the intensity values in Table 4-6 and correct output pattern proved that light can travel from the backlight, through the LCD, and into the fibers for proper illumination. While not groundbreaking, it is evidence supporting the feasibility of an LCD as a HENN system input and as a weighting system. It was also proved by the single fiber test, but this confirmed it for the fiber plate. It was not guaranteed because the input light intensity from the LCD was on the order of nanoWatts and if the losses were too high, then the output may not surpass the illumination threshold in the bistable optical device circuit. Even with the dismal efficiencies in Table 4-7, the higher values support the possibility of minimal loss when using the LCD and fiber plate.

Chapter 5

Creating the Weighting Image for Testing

The reason for characterizing and testing the LCD is to determine its viability to display weights for the HENN input and the hidden layers as well. As discussed in Section 2.5, the modified HENN training algorithm and system allows for the LCD with backlight to function as the input light source and the hidden layer weighting system.

While the other steps are rather simple to deal with, the transition from the weighting matrix to an image displayed on the LCD is rather complicated. The final weighting value matrix is 49 x 49 element matrix and the output image from the weight generation program, shown in Figure 5-1, is a 800 x 600 pixel JPEG. This chapter explains the creation and implementation of the program to generate the weighting image and also how the fiber plate, LCD, and weighting image all interface with each other.

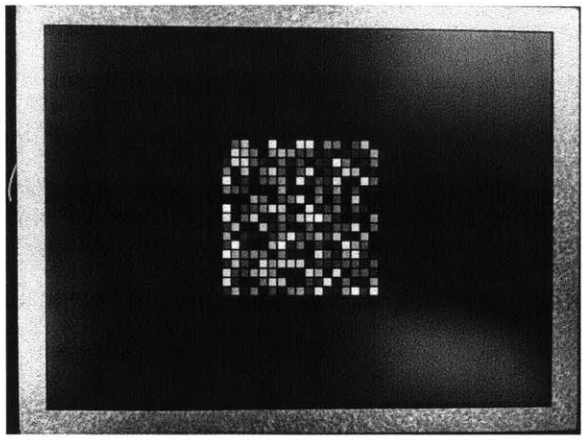

Figure 5-1: Photograph of the output weighting image generated by the weighting image program. The entire active LCD screen is illuminated black except for an area slightly larger than the fiber plate active area. The non-black image is the weighting image generated to modulate the intensity input of the fiber plate. In this image, the weighting is done by node rather than by fiber, so each individual square covers the area of one 3 x 3 fiber group. The black space in between each square is the buffer area where the intensity value is zero. The current weighting image displayed is just an example and has random weights varying from 0 to 255. For the per fiber weighting scheme, the same type of pattern emerges, just split each square in this image into a group of 3 x 3 equal squares with black buffer pixels between them.

5.1 Developing the Weighting Image Program

The final weighting image product is a 800 x 600 pixel image that perfectly aligns a unique weighting mask with each fiber in the first layer of the fiber plate. However, the sole input is a weight value matrix provided by the microcontroller, with dimensions equal to the number of fibers in a single row of the node array. Therefore, a program was developed to bridge the gap between the training algorithm matrix and the physical implementation of the weights in the network to prove the viability of an LCD based weighting system. The problem was broken down into four different steps:

1. Mapping the training weights to an array of physical pixels
2. Positioning and sizing the weighting filter to fit the LCD screen
3. Changing the array to an RGB pixel model
4. Creating a JPEG file of the pixel array for proper display

Numerous different languages were researched, but in the end MATLAB was chosen due to familiarity and ease of matrix manipulation and image processing. MATLAB has especially powerful image processing capabilities and can easily represent an image as an array of values. It was extremely useful when trying to construct the weighting image pixel by pixel. A separate MATLAB function was generated for each problem in the list above to allow for function reuse and adaptability. A large majority of the earlier tests utilized the LCD screen image and the generation of those images employed at least one if not two of the steps above, again pointing to the importance of modularity.

Mapping the Training Weights

As proven by the Absolute Intensity experiment in Section 3.4, varying the pixel color is an effective method for modulating the intensity of light passing through it. The obvious result was mapping the training weights to different pixel colors. However, before discussing that a more fundamental difficulty required a solution. The pixels were to be used as modulators for light traveling between the light source (backlight) and the fiber plate. As per the artificial neuron model and HENN project, each fiber possessed a unique weight and all light coupling into that specific fiber was mandated to be uniformly modulated. Satisfying this prerequisite meant that the weighting coverage for each fiber had to be a slightly larger than the fiber itself. The challenge was to implement this coverage for each fiber with an image made of pixels with dimensions disparate to the fiber plate dimensions. The difference between the current values of W_{pixel}, W_{dead} and $L_{Internode}$, $L_{Interfiber}$, D_{Fiber} created an offset nightmare that forced extra pixels to be added in some places but not others. This would not have been a major problem if the program was only created for these current fiber plate specifications, but the fiber plate will inevitably be redesigned and this program must be able to properly function with new dimension values. The exact problem and remedy will be explained after setting up the design method.

For this function, the pixel dead space plus pixel dimension was considered one d-pixel and the dimensions assumed equal. The fiber plate dimensions were then

mapped to a d-pixel value. The initial test program was mapping unique weights to each node, which worked perfectly fine. The node length and the distance between the edges of two nodes were floored to the nearest d-pixel integer and then additional d-pixels were added where needed to achieve the d-pixel length equivalent to the LCD active area dimensions in millimeters. It is important to note the inter-node length is the distance between the origins of two nodes, so the distance between the edges of two nodes is the between node length. Once the length issues had been resolved, the program generated the image one d-pixel row at a time. Assuming the input weight matrix was comprised of values from 0 to 255, the program created a row matrix with node values corresponding to the specific value in the weight matrix and the between node values as zero. Take a 3 x 3 node array with Node_Pixel_Len = 4 and Between_Node_Len = 3 for example, if the first row of its weight matrix is $\begin{bmatrix} 122 & 54 & 239 \end{bmatrix}$, then the first weight image row would be

$$\begin{bmatrix} 122 & 122 & 122 & 122 & 0 & 0 & 0 & 54 & 54 & 54 & 54 & 0 & 0 & 0 & 239 & 239 & 239 \end{bmatrix}$$

However, to assure the full coverage of each node, a buffer coefficient was included. This coefficient was multiplied with the between node distance to determine the length of a buffer zone that would replace some of the between node values with values equal to those of the closest node. This essentially expanded the dimensions of each node's weighting image and while it reduced the between node zero gap, it manufactured an offset allowance for positioning the fiber plate on the LCD. Depending on the buffer coefficient, the plate can have the same weighting even when accidentally shifted by several tenths of a millimeter. Figure 5-2 illustrates the discussed mapping. Returning to the earlier 3 x 3 node example, with a buffer coefficient of $\frac{1}{3}$,the new weight image row would be

$$\begin{bmatrix} 122 & 122 & 122 & 122 & 122 & 122 & 0 & 54 & 54 & 54 & 54 & 54 & 54 & 0 & 239 & 239 & 239 & 239 & 239 & 239 \end{bmatrix}$$

The buffer zone length is equal to one-third of Between_Node_Len or one and, there-

fore, one value of the Between_Node_Len zero gap is altered to reflect the adjacent node weight. Additionally, a value was added on the exterior of the matrix because the inner Node_Pixel_Len expands so the outer lengths should as well. Coincidentally, since this buffer replacement concept also applies for the weight image columns as well, the new example row matrix is actually the second weight image row. The buffer coefficient altered weight image row matrix was then replicated in the column dimension one node length to create a matrix that correctly aligned with the first node row on the fiber plate. This process repeated for each node row, covering the entire node array. For reference, the code for this program is located in Appendix Section A.3.

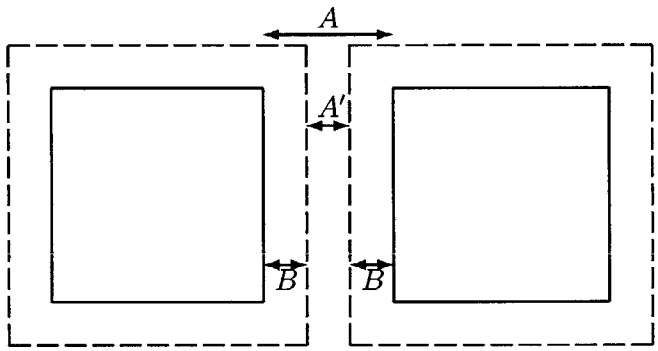

Figure 5-2: Two weighting squares following the mapping described in the previous paragraph including buffer coefficients. This generic figure applies when the square is sized for an entire node or for a single fiber, the concept remains unaltered, though the distances may change. In this figure, the solid lines represent the original size of the weighting squares and the dashed lines are the new size with the buffer expansion. The distance between the original squares was 18 units and is represented by A. After, applying the buffer coefficient of $\frac{1}{3}$, A shrinks to A'. B, the expansion distance added by both squares is calculated by $B = \frac{1}{3} \cdot A$. Therefore, after the inclusion of the buffer mechanism, the new distance between weighting squares is $A' = A - 2 \cdot B$. This situation applies to the distance between any two adjacent weight squares for both axes. For context, these two squares represent any two axially adjacent squares in the left image of Figure 5-4 weighting image.

This program worked perfectly well and the resulting un-sized image is shown on the left in Figure 5-4, but the HENN project required per fiber weighting. The same design concept may be applied to the fiber based version, the only difference being the smaller dimensions. And it was precisely the smaller dimensions that rendered this design concept impractical. The layer one nodal length for the current fiber plate is

approximately 2 mm and the between node distance is 1 mm. Assuming the average d-pixel dimension is about .2021 microns, the previous measurements translate to 9.9 and 4.9 d-pixels, which work fine with the current design. On the other hand, the fiber diameter is .5 mm, which translates to a workable value of 2.5 d-pixels. But the inter-fiber distance, which is .75 mm, is measured similarly to the inter-node distance, so the desired value is the between fiber distance, .25 mm. This implies there is only one d-pixel between the two fiber weighting squares. While the buffer concept can be removed from the model, it does not remedy the larger problems. With only one d-pixel separation, the movement allowance is little to none and due to the offset between d-pixels and fiber plate dimensions, attempting full coverage of a single fiber also poses problems. The fiber should require only 3 d-pixels for full width

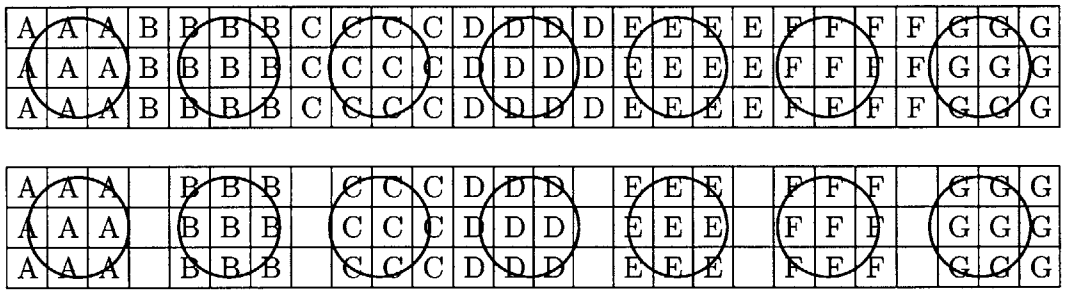

Figure 5-3: A scaled representation of one possible alignment for fiber plate and groups of three d-pixel rows. Everything is properly scaled, using values from experimental measurements and known lengths. The circles represent fibers and the squares are pixels plus dead space or d-pixels. The letters represent the arbitrary weight assigned to that pixel for fiber weighting coverage and blanks are just a weight of zero. The bottom group is the results from assigning 3 x 3 pixels of weight to each fiber. The second level is the results of assigning 4 x 3 to each pixel (A group has one value that is cut off on the left). And the third group is just for reference.

coverage, but the offset forces that to be four. Figure 5-3 is a scaled representation of the alignment problems faced by the current fiber plate. Clearly, neither assignment works properly due to the fiber spacing and pixel spacing offset. Allocating three d-pixels to each fiber fails to adequately cover bottom D and bottom G. Even if the coverage is expanded to four and an extra d-pixel is concatenated to the front of the top row, it still fails. It seemingly works at first but finally it fails when the top F weight covers some of the top G fiber. Clearly, the evidence points toward a mixture

of three and four d-pixel allocations for a fiber dependent on its position. While it may not be arduous to ascertain the pattern for these distances, it seems unnecessary and impractical, especially if the pattern changes with new dimensions. Therefore, to properly utilize the weighting image it is necessary to re-evaluate the fiber plate dimensions, something discussed in Section 6.2.

Returning to the properly functioning code, once the process has been repeated for each node row, the result is a matrix slightly larger than the active area of the fiber plate and having as many weighting squares as the number of fibers or nodes. The pseudo-image matrix is ready to move to the next step.

Positioning, Sizing, and Transitioning to RGB Pixel Model

Once the pseudo-image matrix has been mapped to d-pixels, it is slightly larger than the active area of the fiber plate. The active LCD area is 16.2 cm by 12.1 cm compared to the active fiber plate area of 4.85 cm^2. The smaller output from the weight image creator must be scaled to fit the larger LCD d-pixel resolution of 800 x 600 pixels to properly display the image. Before the scaling, the weight image must be optimally positioned for fiber plate alignment. The exact alignment will be discussed in the Section 6.2, when redesigning the fiber plate, but the program allows free placement anywhere within the 800 x 600 pixel display. After setting the weight image location, the program sizes the image to fit a 800 x 600 profile. A black 800 x 600 d-pixel image is created and the weight image is overlaid at the set position. An example of the weight image with an origin at the center of the rectangle is illustrated on the right in Figure 5-4. For reference, the code for this program is located in Appendix Section A.4.

Once properly sized, the image matrix must be converted to an RGB color pixel model. The initial image after sizing is an 800 x 600 x 1 element matrix or black and white intensity image. Knowing that the end product is a JPEG, the program shifts the matrix properties to be compatible with the JPEG image file format. The RGB pixel model satisfies JPEG compatibility requirements, allows for pixel by pixel manipulation, and provides a wide range of different colors. The RGB color model

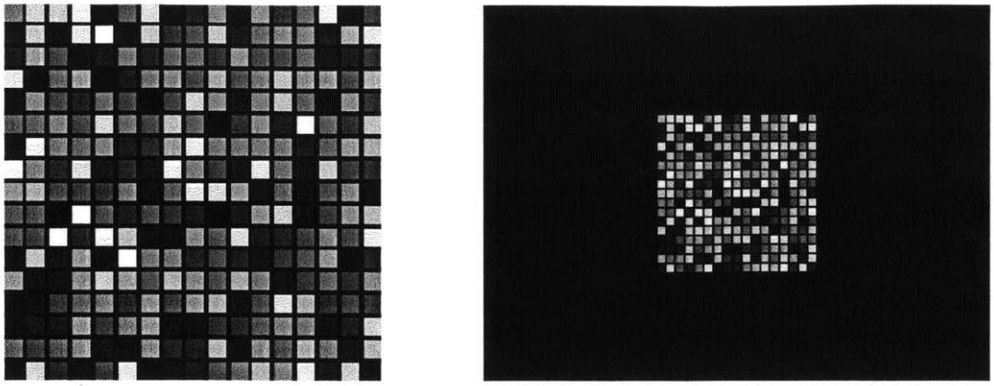

Figure 5-4: Left: Scaled representation of a randomly chosen un-sized weighting image for uniform node weights. Right: Scaled representation of same weighting image for uniform node weights padded with zero pixels to fit the 800 x 600 pixel JPEG picture profile. The different shades of gray represent the 256 different possible weight values.

describes the pixel with a 3 x 1 matrix of eight bit unsigned integers from 0 to 255. Each element in the pixel matrix represents a color, typically $\begin{bmatrix} R & G & B \end{bmatrix}$. This implies 256 different shades of red, blue, and green, a format recognized in most consumer LCD technology. After converting every matrix element to an unsigned eight bit integer, the program reads an input to determine the color. Currently, there are four different selections for coloring the image: white (or gray scale), red, green, blue. The latter three are achieved by zeroing the other two values in each pixel matrix, but the gray scale occurs when the R, G, and B values are equal for a single pixel. The program can easily be expanded to choose more colors, but the current setup sufficed for the intensity testing images and the test weight images. For reference, the code for this program is located in Appendix Section A.5. After coloring, the image is ready for format conversion and display.

Creating the JPEG

The last step in creating the weighting image is converting the colored 800 x 600 x 3 array of unsigned eight bit integers into a JPEG image. The chosen LCD screen is from a digital picture frame and strongly suggests displaying only JPEG formatted images. The JPEG image file format is meant for the compression of photographs and other images with smooth variations of tone and color. While this characteristic is

beneficial for the transitions between two extremely similar weights, sharply contrasting images like the weighting image can create noticeable image artifacts from the compression. To prevent these artifacts from affecting the weights, the compression must be prevented. Additionally, the file format syntax and structure are complicated, requiring initialization, a baseline discrete cosine transform, Huffman tables, and quantization tables. Rather than use the code to create all that from scratch, the program relies on a few MATLAB functions and a blank 800 x 600 pixel prefabricated JPEG image. The program makes a copy of the blank JPEG in a specific file location dictated by user input. Then it replaces the blank picture data with the weighting image data on the copy, formats it, and saves it as an uncompressed lossless JPEG. The resulting JPEG weighting image is compatible with the picture frame LCD and it has no compression artifacts. For reference, the code for this program is located in Appendix Section A.6. After the weighting image JPEG has been created, the last step is to display it on the LCD. The current method for accomplishing this is to save the weighting image on a flash drive and then insert the flash drive into the LCD picture frame's USB slot. After insertion, the LCD screen immediately displays the image and switches after an adjustable amount of time, if there are multiple images on the flash drive.

Chapter 6

Conclusions

This thesis has outlined the purpose of the prototype for a neural network optical and electrical hybrid, that can be built as a proof of concept and encourage more research on this project. It then focused specifically on developing and testing an LCD-based weighting image and compatible fiber plate. Finally, it discussed a program that could build the weighting image on the LCD screen and what that meant. Based on the original goals, this thesis succeeded in what it attempted to do. It has established a guide for adapting an LCD as a weighting system for the fiber plate in the HENN.

Reexamining the experimental results, the LCD and backlight in the Aluratek ADPF08SF can act as a system input weighting system for the HENN architecture. The intensity versus RGB experiment proved the output light intensity of individual pixels can be modulated by controlling the RGB value. The output range generated by this experiment was a nonlinear relationship that would be sufficient for any photodetector. The position-based variation light intensity test established that if the fiber plate is properly designed and all weighting is equal, the LCD can ensure uniform light intensity enters the fibers. It also determined that the variation in input intensity, due to dead space and pixel alignment with the fiber, is small enough to guarantee a consistent input value even when the fiber plate is slight misaligned. This was critical for the LCD to properly apply the training algorithm weights. The single fiber loss and LCD fiber plate tests verified that at the minimum LCD attenuation, even with fiber loss, the fiber plate output light intensity is a non-negligible and mea-

surable value with a range that can easily be detected by a photoelement. At a bare minimum, the weighting system needed to efficiently couple light into the fiber plate and then have the measured output be a non-negligible value that was of the proper pattern relative to the input. These requirements were most definitely proven by the experiments. The last requirement is to be able to map 49 x 49 matrix on a weighting image the active size of the fiber array on the weight image onto the LCD screen. The weighting image creation program clearly proved this was possible by mapping a weighting image to the LCD. Therefore all the requirements have been met.

Clearly, the concept of an LCD as a weighting system for the fiber plate is more than feasible. However, there is one caveat that prevents this statement from encompassing the HENN project 100%. All the experiments were conducted with the backlight as the lighting source. While this is perfectly acceptable for the input layer of the HENN it does not exactly replicate the hidden layer weighting. The hidden layer weighting systems would use a non-uniform backlight such as a LED array or something similar. The effects of a non uniform light source are unknown at this junction and therefore prevent this from being considered a solved problem. The revised statement is that an LCD as a weighting system for the HENN is a feasible concept for the system input always. The feasibility of its use for hidden layer neurons is reasonable assuming that the coupling and fiber attenuation are relatively similar for non-uniform light and that the weight generation program does not apply.

6.1 Recommendations for Optimization

There are three main recommendations from the experimental data that can be made for using this system.

1. Operate the network with white light

2. Resize the fiber plate

3. Expand the current test to encompass non-uniform light

The first recommendation will be briefly explained while the last two while be discussed in the next two sections.

The HENN should be operated with white light as opposed to a specific color, because there is no clear advantage to picking one over the other. The exceedingly high transmittance rate for the fiber loss test supports a theory of minimal spectral attenuation in the fiber. Looking at the relative intensities for each color, it is clear that green is the strongest, but white is even better despite the fact is a broad band signal as opposed to a signal wavelength. There is no hard evidence to favor one light wavelength over the other, therefore the best option is to employ white light. It only makes sense to choose the one with the largest average intensity, as its noise immunity is higher.

The next section will step through the process of designing a fiber plate using the selected LCD or a similar one. The guidelines are not guaranteed to apply for other LCD types, but they should provide an excellent framework to begin design. Additionally, the previous experiments can easily be adapted for other LCDs and systems, so they can also provide direction in adapting an LCD.

6.2 Redesigning the Fiber Plate

The three previous chapters have been characterizing the LCD weighting system, gathering functionality data, and developing a weighting image program. The subcontext for these actions was the inspiration of enhancing the complementary nature of each HENN subsystem. One subsystem in particular proved to be in dire need of an upgrade, the fiber plate. The current fiber plate predated the advent of the HENN project and, therefore, was designed and built without any of the guidance or knowledge gained from the project's evolution. As a couple of the previous experiments have indicated, the best course of action for building a solid HENN system is the redesign and refabrication of the fiber plate. Specifically, for the LCD to function as a viable efficient weighting system in the HENN, it must be compatible with the fiber plate spacing. This is not the case for the current fiber plate, therefore, the fiber

plate should be redesigned. This section will discuss the benefits of redesign and the effects of the necessary changes based on the previous test results. The goal is to design a fiber plate ideally suited for the neural network functionality, while minimizing problems created by dismantlement/reconstruction, movement, LCD specifications, system improvement, and system mobility. The new design discussion will start from the basics and rebuild the fiber plate piece by piece. Obviously this is not the penultimate authority on fiber plates, proper design etiquette dictates building a small prototype to characterize before constructing the actual plates. The prototype plate results may suggest designs conflicting with those listed below, so, ultimately, what works for the HENN may not be applicable for all projects. However, these new design guidelines will save an enormous amount of time when trying to develop an LCD weighting system and a fiber plate. Any project will have a significant head start compared to those starting from scratch.

Returning to the experimental results, the one common theme amongst them is that while the LCD can function as a weighting system, it's integration with the current fiber plate is inefficient. For a stronger case, the fiber plate must be redesigned to ensure that the LCD meets the requirements of the HENN weighting image system. Even if a specific LCD is perfect as a weighting system, larger offset with the fiber plate will increase inefficiency because more power will be lost from scattering and the network capacity will decrease. The more variation the network has to account for in the training algorithm, the less versatile the network becomes. The HENN requires each subsystem to integrate and cooperate with the others, so subsystem incongruity disables it as well. As stated in most of the experimental results, the current fiber plate fundamentally conflicts with the selected LCD dimensions. Therefore, the prudent route is redesigning the fiber plate. Not because the LCD was selected for this thesis and displayed compatibility, whereas the fiber plate was preexisting, but because, in general, the most advantageous and efficient method bases the fiber plate on the LCD. The reason is practicality, proceeding this way is cheaper and less time consuming, which is especially pertinent for a tight budget prototype like the HENN. Unless the research group intends to design and build an LCD, which will be extremely time

consuming, the LCD will be a commercial product. Obviously, there are companies that can custom manufacture an LCD system based on a fiber plate, but that is financially unwise on a small budget. The only remaining option is to search the commercial sector for a reasonably priced LCD that meets the project constraints, which is exactly what this thesis did. Many products will contain LCDs that cannot be separated from the backlight, others will lack resolution, while others still will not allow for simple interfacing with the controls. Once a suitable LCD is finally found, it is extremely unlikely that the specs will sync with the preexisting fiber plate, forcing the abandonment of the preexisting fiber plate. On the other hand, the fiber plate is always custom built and simple to design and build. It uses cheap materials such as PVC and plastic optical fibers, which can be bought in bulk. For these reasons stated above, the logical decision is purchasing the desired LCD and then designing the fiber plate.

Fiber Selection

The first step is selecting the fiber for the interconnections. The most logical action is increasing the size to a 1.0 mm diameter fiber, $D_{Fiber} = 1.0\,mm$. With the current LCD screen resolution, the original .5 mm fiber area covered just over four pixels. The small number of pixels and the dead space between them, meant that the input light intensity of a fiber would vary based on position. This was proven by the positional input light intensity experiment in Section 4.1. As the fiber diameter increased, the total variation decreased, which means the 1.0 mm fiber is more stable than the .5 mm. For the HENN to function properly, the light output by each fiber plate node must be able to vary around a certain threshold value. If the input intensity is too small or the attenuation too large, the output light intensity range will be skewed from the threshold value and inhibit correct intensity measurement. The 1.0 mm fiber minimizes the possibility of that scenario, as the single fiber loss data in Table 4.1 affirms the 1.0 mm transmits more total intensity more efficiently than the .5 mm. The thicker 1.0 mm fiber generally withstands strain better as well, but may also increase the difficulty of "wiring" the interconnects. Now that the fiber size has

been substantiated, the fiber materials can be examined. The same type of plastic fibers used for the original fiber plate are best for the new iteration. The reason for choosing a plastic fiber over a glass fiber is twofold. For the HENN project's LCD, BOD, and fiber plate, the plastic fiber is a cheaper more durable alternative. The transition to glass fibers will be made once the HENN prototype is completed and the group begins work on the smaller-dimensioned COIN. The fiber plate experiment in Section 4.3, supports proceeding with the current material. It did not show any measurable signs of crosstalk or non-negligible light leakage and as stated before, has a very high transmittance.

Fiber Plate Input Fiber and Node Spacing

The spacing between the fibers and nodes is the most crucial and largest difference between the original and redesigned fiber plate. The inter-fiber $L_{Interfiber}$ and inter-node $L_{Internode}$ distances should be multiples of the unit d-pixel (pixel plus dead space) width or height. The training algorithm selected for the HENN, which is discussed further in Simpkins [1], depends on an absolute intensity value. Therefore, ensuring the minimization of variation between fiber input light intensities is critical. Forcing the inter-node and inter-fiber distances to be d-pixel multiples ensures uniform input light intensity. The experimental results from the positional intensity variation test in Section 4.1 substantiate this claim. The intensity value in Figure 4-4 is mapped over an area of four d-pixels and clearly displays a periodic nature. For the input light intensities to match, each fiber must be positioned at the same intensity origin, forcing a distribution of one fiber per unit d-pixel. However, the fiber diameter is much larger than one d-pixel, so the distribution is spread out more, but the distance between two fibers must be a multiple of d-pixel dimension for the intensities to match. This also applies for the inter-node distance as well, however, it must be larger than the inter-fiber distance.

Additionally, the inter-fiber distance should be at least the fiber diameter plus three d-pixels and the inter-node distance should be no less than twice the inter-fiber distance plus the fiber diameter plus five d-pixels. As discussed in the weight

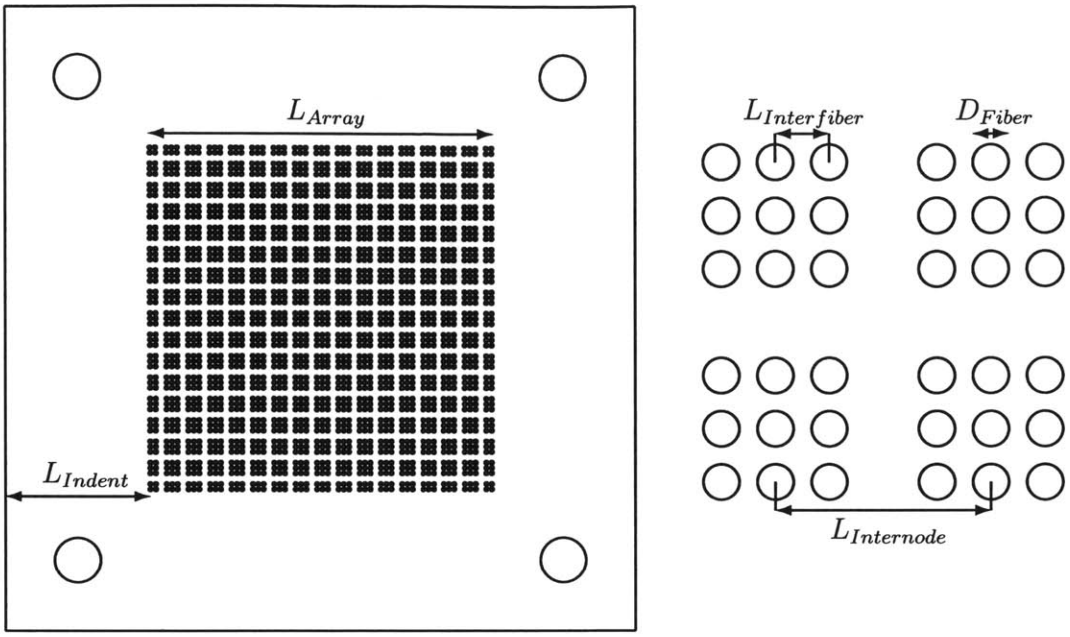

Figure 6-1: Left: Scaled 17 x 17 node array fiber plate first layer. Right: Zoomed in 2 x 2 node array to illustrate sizing.

mapping portion of Chapter 5, in order for the weighting program to properly create a per fiber mapping with a buffer coefficient, it is necessary to have a minimum of three d-pixels between each fiber weight square. This is why the inter-fiber distance should account for that d-pixel distance. Forcing theses distances to the nearest d-pixel multiple greater than the aforementioned values exponentially decreases the difficulty of creating a weighting program for the fiber plate. The inter-fiber and inter-node distances could be directly mapped to a pixel value and not have to deal with adding pixels to compensate for offset.

Fiber Plate Output Fiber and Node Spacing

This dimension depends directly on the fiber diameter and inter-node distance. The node diameter D_{Node} must be large enough to fit all nine 1.0 mm fibers, therefore since the previous node diameter was 2.0 mm, twice the fiber diameter should be a node diameter of 4.0 mm. It can probably be sized slightly smaller or larger, but it should not be greater than the width of a layer one node. The positioning for

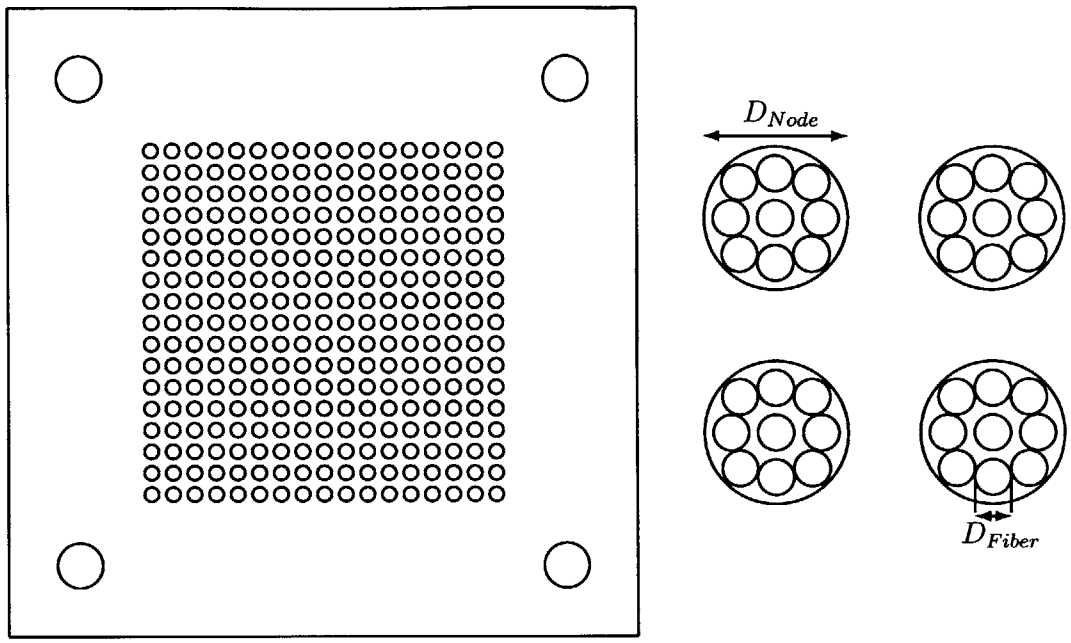

Figure 6-2: Left: Scaled 17 x 17 node array fiber plate second layer. Right: Zoomed in 2 x 2 node array to illustrate sizing and ideal positioning of fibers.

each layer two node should be the origin of the center fiber in the corresponding first later node. That is the adjacent layer two node origins should be separated by the inter-node distance.

Design for Repeatability

Designing the fiber plate for repeatability returns to the training algorithm's dependence on absolute light intensity inputs and, therefore, the desire to maximize that intensity as well. The repeatability, is the ability for the fiber plate to always have the correct input intensity value, even if the system has been taken apart and then reassembled. The fiber plate must be positioned so that it can reliably and effortlessly return to this set value. The best method is to secure the plate on the corner of the LCD. The metal casing will prevent movement two direction and the fiber plate can be secured with vertically applied pressure to limit the other two direction. The last step for repeatability, is to guarantee the secured alignment position generates the maximum possible intensity. This is accomplished by offsetting or indenting the fiber

plate active node array a certain distance, L_{Indent} into the plate. From the upper left corner of the LCD active area, it is 1 cm to the top edge and 1 cm to the left edge, so the indent must account for that. The following values are be estimates of actual values and should not be taken as the actual measurements, their purpose is to explain how the calculation works rather than find it exactly. According to Figure 4-4, one maximum is around (725,670), so using the pixel mask in Figure 4-1 as a model for the LCD screen edge, the position of one maximum relative to a pixel may be determined. The maximum is about 125 microns right of the pixel edge and 70 microns below the pixel top, placing it about vertically centered on the green blue sub-pixel border. So the indentation value is different for horizontal $L_{HIndent}$ and vertical $L_{VIndent}$, 1 cm plus 125 microns plus as many d-pixels as desired and $L_{HIndent}$ minus 55 microns, respectively. In order for these measurements to be accurate it is necessary to refine the masks to one matrix element equal .1 or .01 microns for the positional intensity variation test. Then run the test again and determine the exact coordinates of the maxima for the repeatability. This design assumes that when placing the fiber plate against the metal fitting, the metal will give a negligible amount. This cannot necessarily be said for the fiber plate, depending on its material composition. If it is a plastic, though, and the normal alignment is assumed to be the maximum input light intensity, Figure 4-4 illustrates a 0.5-1% intensity reduction resulting from 50 microns of movement on either axis. A miniscule variation such as this, (for example 0.7 nW is 1% of maximum input light intensity for a 1.0 mm fiber), could be integrated into the weight training without to much difficulty.

Plate Material, Size, and Number of Nodes

None of the experiments test characteristics of the plate material, size, or number of nodes, so it can be considered personal preference. The material for the original fiber plate was PVC and it had no significant problems. The possibility of give, which was discussed in the previous paragraph, could inspire switching materials, but something less malleable, such as steel, would make the plate too heavy and subject the LCD screen to superfluous pressure. The size is determined by the twice the

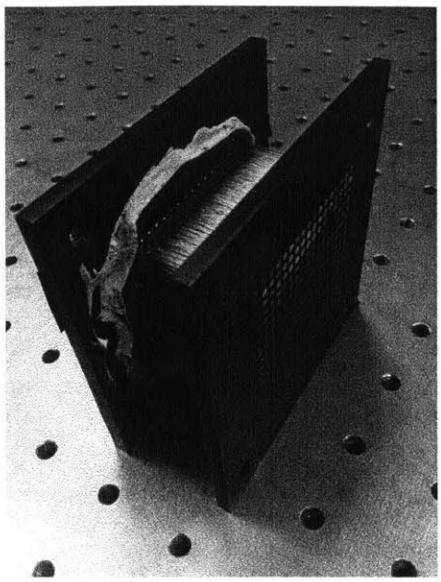

Figure 6-3: Angle view of the current fiber plate constructed by Bill Herrington.

indentation length and the dimension of the fiber plate active array sans one fiber diameter. Finally, the number of nodes is personal preference, however, wiring the fiber plate grows exponentially more difficult as the node number rises. Increasing the node number improves the neural network capabilities and may be beneficial for practical testing, but the current value of 17 x 17 is suitable for the HENN.

Building the Interconnects

One of the last steps, building the interconnection network incorrectly can permanently cripple a fiber plate's abilities. The first tip is to leave at least a centimeter of extra fiber, protruding from each end of the plate. Otherwise, the fiber ends may not fully thread the connection or be too short to cut. After threading the fibers build a small putty dam around the active area on the inner for both sides. Elevate and fasten the fiber plate with the desired side down, then fill the bottom plate damn with thirty minute epoxy. The thirty minute dry time allows epoxy to seep into the fiber holes of the bottom plate. Once completely dry, vertically flip the plate and repeat the procedure for the other layer. With both sides dry, the fiber plate can be taken a shop for milling. The extra protruding wire must be cut down to sev-

eral millimeters above the plate. This prevents the creation of ripping indentations that greatly reduced the transmittance. The exact distance to cut above the plate is unknown but the ripping likely also correlates to how quickly the machine cuts the fibers. Once the fibers have been cut, the ends require polishing until flush with the fiber plate. Starting with 600 grain, moving to 1000 grain and then possibly the polishing compound is the most efficient approach. Once the polishing is complete, the fiber plate should be done. Anything else regarding the fiber plate design is rather self-explanatory, so the new fiber plate dimension are in Table 6.1.

Measurement	Value
Dimensions	$2 \cdot L_{HIndent} + L_{Array} - D_{Fiber}$ x $2 \cdot L_{VIndent} + L_{Array} - D_{Fiber}$
Nodes	17 x 17
d-pixel	202 x 202 μm
D_{Fiber}	1.0 mm
Min $L_{Interfiber}$	Closest multiple of d-pixel $> (D_{Fiber} + 3 \cdot$ d-pixel$)$
Min $L_{Internode}$	Closest multiple of d-pixel $> (2 \cdot L_{Interfiber} + D_{Fiber} + 5 \cdot$ d-pixel$)$
L_{Array}	$(\text{Nodes} - 1) \cdot L_{Internode} + D_{Fiber}$
Min $L_{HIndent}$	$\sim 10.125\,\text{mm} + X \cdot$ d-pixel
Min $L_{VIndent}$	$\sim 10.070\,\text{mm} + X \cdot$ d-pixel
D_{Node}	4 mm

Table 6.1: Dimensions and guidelines for redesigned machined fiber plate.

6.3 BOD-based Light Tests

As discussed earlier in this paper, the experiments in this thesis were conducted with a uniform light source. Logically, the next step is to expand the current tests to consider non-uniform light sources, such as the BOD output light source. The first consideration is confining the light, unlike a backlight the non-uniform light source will likely be directed in more than one direction or have parts be on or off like an LED array. Therefore, the non-uniform light must be confined. For example, if it is an LED array, then the solution is to simply machine a black piece of plastic that places each LED in a cell. The coupling must also be redone to account for the affect of non-uniform light on the fiber. However, this does not address all the

difficulties. If an LED array is chosen then the uniformity and intensity depend on LED viewing angle, which is where a large majority of the optical power exists. Depending on the distance of the LED from the LCD, the illumination of pixels may or may not be uniform, more likely the latter. The output intensity problem can be solved by purchasing a more powerful LED, but that also translates to more power consumption and heat generation, which are also factors to consider. There are several approaches to solving the LED uniformity problem: using frosted LEDs, placing extremely narrow field LEDs further away from the source, and placing wide field LEDs closer to the source. However, this is something to be investigated by other group members. The factor of LED wavelength is also an unknown, but redoing the intensity wavelength profile measurement would shed light on the issue. This would also help determine the filter absorption, since the LED spectrum and the measured spectrum could be compared. Another consideration is fiber coupling efficiency versus frequency, thought the current results from the single fiber loss test point to negligible spectral attenuation.

Once the testing obstacles are completed, the expanded test results will provide significant conclusions. The first expansion is testing the hidden layer neurons with the planned HENN architecture, to decide whether they are plausible candidates for LCD weighting systems. The second expansion examines the LCD as a spatial light modulator for the hidden neurons in the planned HENN architecture. The backlight provides half of the testing for a spatial light modulator, but it is cannot be considered an SLM without testing the non-uniform BOD source .as well Using the tests of the LCD with backlight as a basis, adding the results of the non-uniform BOD LED array light would absolutely confirm the use of an LCD as a spatial light modulator within the HENN, assuming the new data agrees with the current.

6.4 Future Work

With the fiber plate redesigned, the non-uniform light/SLM discussed and the LCD as a weighting system for the fiber plate confirmed, the only thing left to discuss

is the future direction of the project and the discussed subsystems. The following several concepts build upon current problems in the system or possible enhancements for better performance.

LCD Controller Interfacing

The current method for displaying weights on the LCD screen is via USB. While acceptable for the testing and characterization of a device, if and when the HENN is constructed with more than one LCD, each one would need to be manually loaded with a USB. For multiple iterations through the HENN the weighting images would need to be cycled via buttons on each LCD. It is highly inefficient and would retard the calculation process. This is exactly why one of the qualifications for the chosen LCD was ease of interfacing with the controller. The future goal is a computer or master microcontroller interface that can synchronously control each LCD to cycle the images. Furthermore, the system could store the images and upload a new image to each LCD for new iterations.

Expanding Network Function with Multiple Colors

Rather than stay with white light, a possible future is exploiting the existence of all three colors in the network. Most of the data gathered by this thesis is separated into three colors, so working with one color would be simple. The interesting segment is what the differing colors represent. There are two main options that will be discussed: Using the colors for three different trainings and using one color for positive values and another for negative values. Since the system was broken up into three colors, most of the data already exists. Only a few modifications would be required to implement this. First, the BOD would need three different detectors for each node, one for each wavelength, then the regular LED could be replaced by a four pin tricolor LED to save space. The major problem with this system is the crosstalk registered by the unintended photodetectors. If they register a high enough value, the more than one switch will close, possibly shorting the circuit or nullifying the value depending on the circuit. Also, while single tricolor LED saves space, it presents a geometry problem

for aligning the three photodetectors. However, if these problems were solved, the neural network would hypothetically have three different sets of weights to analyze the same picture or the system could be trained for three unique image groupings. Both of which vastly improve the abilities of the network.

The second option is using only two different colors, but simultaneously. For example, the blue could represent positive weights and the red negative. The system would be built similar to the tricolor, but with only two. Because the light to light interaction is very weak, both signals could propagate through the system. The only difference is there would be an optical element that separated the colors at the output. The positive and negative weighting scheme is so enticing because true neural networks have negative values, however, it is impossible to implement a negative intensity, therefore, one of the two positive intensities has to be dubbed negative for functional purposes. The addition of a negative or dampening weight removes one advantage software neural networks hold over their hardware counterparts.

Appendix A

MATLAB Code Appendix

A.1 Absolute Intensity versus RGB

../LCD_Weighting/intensity_vs_RGB.m

```
function [] = intensity_vs_RGB()

%repeat_colors()

data = xlsread('intensityRGB_data.xlsx');

max_b = max(data(:,2));
max_g = max(data(:,4));
max_r = max(data(:,6));

max_val = max([max_b max_g max_r]);

figure();
plot(data(:,1),data(:,2)/max_val,'-ks','LineWidth',1.75,'MarkerSize',3.5
    ); grid on; hold on;
plot(data(:,1),data(:,4)/max_val,'--k^','LineWidth',1.75,'MarkerSize',4)
    ;
```

```
plot(data(:,1),data(:,6)/max_val,':ko','LineWidth',1.75,'MarkerSize',3.5
    ); hold off;
%plot(data(:,1),data(:,2),'-k','LineWidth',2,'MarkerSize',3); grid on;
    hold on;
%plot(data(:,1),data(:,4),'--k','LineWidth',2,'MarkerSize',3);
%plot(data(:,1),data(:,6),':k','LineWidth',2,'MarkerSize',3); hold off;

legend('Blue (486 nm)','Green (537 nm)', 'Red (584 nm)');
title('Relative Intensity of RGB Values');
xlabel('(R,G,B) Value');
ylabel('Rel. Intensity');
%ylim([0 10.5]);
xlim([0 260]);

    function [] = color_maker(color,number)
        image = number*ones(600,800);

        colored_image = image_color(image,color);

        image_saver(colored_image,[color,num2str(number),'.jpg'],'\
            Test_Images\Intensity_vs_RGB\');
    end

    function [] = repeat_colors()
        for i = [0:8:255 255]
            color_maker('red',i);
            color_maker('green',i);
            color_maker('blue',i);
        end
    end
end
```

A.2 Positional Intensity Variation

```
function [intensity,max_intensity,startpt,endpt,h] = intensity_variation
    ()

% VARIABLES
% pixel sizing

W_pix = 183;
W_micro = 61;
H_pix = 139;
W_dead = 18;
H_dead = 63;

% pixel intensity to color rating
blue = .91;
green = 1;
red = .69;
dead = 0;

%R_fiber = 243; % cladding + core radius = 250
R_fiber = 368; % cladding + core radius = 375
%R_fiber = 490; % cladding + core radius = 500

% determine start point coefficients
switch R_fiber
    case 490
        a = 3; b = 2.5; rep = 8;
    case 368
        a = 3; b = 2.5; rep = 8;
    case 243
        a = 2; b = 1.5; rep = 6;
end

[m_h,m_w,pix_array] = LED_Mask(rep);

% find the path of circle origin
```

```
%[H,W]
startpt = [a*H_pix + b*H_dead , a*W_pix + b*W_dead];
endpt = [(a+2)*H_pix + (b+2)*H_dead + 1, (a+2)*W_pix + (b+2)*W_dead +
    1];

intensity = zeros(endpt(1)-startpt(1)+1,endpt(2)-startpt(2)+1);
c_mask = Fiber_Mask(m_h,m_w,startpt(1),startpt(2));

[h_max,w_max] = size(intensity);


for w = 0:w_max-1
    for h = 0:h_max-1
        temip = circshift(c_mask,[h w]).*pix_array;
        intensity(h+1,w+1) = sum(temp(:));
    end
end

max_intensity = max(intensity(:));

figure(3)
h = surf(startpt(2):endpt(2),startpt(1):endpt(1),intensity/max_intensity
    );
colorbar;
grid on;




%-----------------------------------------------------------------
% LED Mask Matrix
%-----------------------------------------------------------------
    function [mask_height,mask_width,pixel_array] = LED_Mask(rep)
        % create one pixel array then replicate

        color_array = [red green blue dead];
```

```
        width_array = [W_micro W_micro W_micro W_dead];
        pixel = [];

        for k = 1:4
            % create height matrix then rep
            color = color_array(k);
            width = width_array(k);
            H = [color * ones(H_pix,width) ; zeros(H_dead,width)];
            pixel = [pixel H];

        end

        pixel_array = repmat(pixel,[rep,rep,1]);

        size(pixel_array);

        pixel_array = pixel_array(1:end-H_dead,1:end-W_dead);

        iptsetpref('ImshowAxesVisible','on');
        figure(1), imshow(pixel_array)
        title('Pixel Mask Coordinate Mapping');
        ylabel('Y Position');
        xlabel('X Position');

        [mask_height mask_width] = size(pixel_array);

    end

%-----------------------------------------------------------------
% Circular Fiber Mask Matrix
%-----------------------------------------------------------------

    function [Circle_mask] = Fiber_Mask(mask_height,mask_width,rr_origin
        ,cc_origin)

        [rr cc] = meshgrid(1:mask_width,1:mask_height);
```

```
        Circle_mask = sqrt((rr-rr_origin).^2+(cc-cc_origin).^2)<=R_fiber
            ;

        % figure(2),imshow(Circle_mask)

    end


end
```

A.3 Weight Image Creation

../LCD_Weighting/image_maker.m

```
%image maker function
function [WEIGHT_MATRIX] = image_maker(weights, node_num, fiber_width_mm
    , intra_fiber_dist_mm,...
                                        intra_node_dist_mm,
                                            display_height_cm,
                                            display_width_cm,...
                                        height_in_pixels, width_in_pixels
                                            , node_buffer_coeff)


% exceptions from variable inputs
if node_buffer_coeff >= .4
    error('The Node Buffer Coefficient is too large. It must be smaller
        than .4');
end

% determine pixels per distance
pixels_per_width_cm = round(width_in_pixels / display_width_cm);
pixels_per_height_cm = round(height_in_pixels / display_height_cm);

```

```
% change to mm from cm
pixels_per_dist_mm = pixels_per_height_cm / 10;

%-----------------------------------------------------------------
% find height and width of node array (assuming they are the same)
    % find exact height of node
node_height_mm = 2*intra_fiber_dist_mm + fiber_width_mm;
    % find that height in pixels
node_height_pixels = ceil(node_height_mm * pixels_per_dist_mm);
    % Rename with final variable
NODE_PIXEL_LEN = node_height_pixels;


% find the intra node distance in pixels
dist_between_nodes_mm = intra_node_dist_mm - 2 * (.5 * fiber_width_mm +
    intra_fiber_dist_mm);
intra_node_pixels = dist_between_nodes_mm * pixels_per_dist_mm;

% find the average length of the node array in pixels using the length
% of the node array in mm
node_array_exact_height_pixels = round(((node_num * node_height_mm) + ((
    node_num - 1) * dist_between_nodes_mm)) * pixels_per_dist_mm);

% Calculate 2 buffer lengths
INTRA_NODE_PIXEL_LEN = floor(intra_node_pixels);

% Buffer that will serve as error for weighted values
NODE_BUFFER_LEN = round(INTRA_NODE_PIXEL_LEN * node_buffer_coeff);

% Buffer that is dead zone between node
INTRA_NODE_BUFFER = INTRA_NODE_PIXEL_LEN - 2*NODE_BUFFER_LEN;

%-----------------------------------------------------------------
% Generate weight matrix
%-----------------------------------------------------------------

```

```
WEIGHT_MATRIX = [];
% build weight matrix
intra_node_decimal = intra_node_pixels - floor(intra_node_pixels);
% for the rows
count_k = 0;
for k = 1:node_num
    count_k = count_k + intra_node_decimal;
    if count_k > 1
        bal_pixel_k = 1;
        count_k = count_k - 1;
    else
        bal_pixel_k = 0;
    end
    if k == node_num
        buffer_k = 0;
    else
        buffer_k = INTRA_NODE_BUFFER + bal_pixel_k;
    end
    WEIGHT_MATRIX_Y = [];
    % for the columns
    count_j = 0;
    for j = 1:node_num
        count_j = count_j + intra_node_decimal;
        if count_j > 1
            bal_pixel_j = 1;
            count_j = count_j - 1;
        else
            bal_pixel_j = 0;
        end
        if j == node_num
            buffer_j = [];
        else
            buffer_j = zeros(INTRA_NODE_BUFFER + bal_pixel_j,1);
        end
        WEIGHT_MATRIX_Y = [WEIGHT_MATRIX_Y ; weights(j,k)*ones(2*
            NODE_BUFFER_LEN + NODE_PIXEL_LEN,1); buffer_j];
```

```
    end
    W_M_Y_LEN = length(WEIGHT_MATRIX_Y);
    WEIGHT_MATRIX = [WEIGHT_MATRIX repmat(WEIGHT_MATRIX_Y, 1, 2*
        NODE_BUFFER_LEN + NODE_PIXEL_LEN) repmat(zeros(W_M_Y_LEN,1), 1,
        buffer_k)];
end

if abs(length(WEIGHT_MATRIX(1,:)) - (node_array_exact_height_pixels + 2*
    NODE_BUFFER_LEN)) > 10
    error('Weight_Matrix mismatched to actual height');
end
```

A.4 Weight Image Sizing

../LCD_Weighting/image_sizer.m

```
%image_sizer
function [whole_image] = image_sizer(made_image,height_in_pixels,
    width_in_pixels,varargin)

[r_y,c_x] = size(made_image);

whole_image = zeros(height_in_pixels,width_in_pixels);

if isempty(varargin)

    offset_y = round((height_in_pixels - r_y)/2);
    offset_x = round((width_in_pixels - c_x)/2);

    whole_image(offset_y+1:offset_y+r_y, offset_x+1:offset_x+c_x) =
        made_image;

elseif length(varargin) == 1
    center_y = varargin{1}(1);
```

```
    center_x = varargin{1}(2);

    offset_y = center_y - floor(r_y/2);
    offset_x = center_x - floor(c_x/2);

    whole_image(offset_y+1:offset_y+r_y, offset_x+1:offset_x+c_x) =
        made_image;

else
    error('Incorrect number of arguments for image_sizer');
end
```

A.5 Weight Image Color

../LCD_Weighting/image_color.m

```
% image color
function [colored_image] = image_color(sized_image,color)

h2_int = uint8(round(sized_image));

switch color
    case 'white'
        colored_image = repmat(h2_int,[1 1 3]);

    case 'red'
        colored_image = repmat(h2_int,[1 1 3]);
        colored_image(:,:,2) = zeros(size(h2_int));
        colored_image(:,:,3) = colored_image(:,:,2);

    case 'green'
        colored_image = repmat(h2_int,[1 1 3]);
        colored_image(:,:,1) = zeros(size(h2_int));
        colored_image(:,:,3) = colored_image(:,:,1);
```

```

    case 'blue'
        colored_image = repmat(h2_int,[1 1 3]);
        colored_image(:,:,1) = zeros(size(h2_int));
        colored_image(:,:,2) = colored_image(:,:,1);
end
```

A.6 Weight Image Saving

../LCD_Weighting/image_saver.m

```
function save_file = image_saver(colored_sized_image,
    output_image_file_name,output_image_location_path)

%figure(7)
%imshow(colored_sized_image)

output_image_path = [pwd, output_image_location_path,
    output_image_file_name];

copyfile('Untitled.jpg',output_image_path);

data = colored_sized_image;

imwrite(data,output_image_path,'JPEG','Quality',100);

save_file = output_image_file_name;
```

A.7 Weight Image Main Method

../LCD_Weighting/WEIGHT_IMAGE_MAIN.m

```
function filename = WEIGHT_IMAGE_MAIN(weight_file, node_num,
    fiber_width_mm, intra_fiber_dist_mm, ...
                                        intra_node_dist_mm,
                                            display_height_cm,
                                            display_width_cm, ...
                                        height_in_pixels, width_in_pixels,
                                             node_buffer_coeff, ...
                                        output_image_file_name,varargin)
% Variables
%----------------
% weights
% max_weight = 5;
% node_num =
% fiber_width_mm =
% intra_fiber_dist_mm   % from center of one fiber to the center of
    adjacent fiber
% intra_node_dist_mm
% display_height_cm
% display_width_cm
% height_in_pixels
% width_in_pixels
% node_buffer_coeff

close all;
addpath('Input_Weight_File');
addpath('Output_Image_File');

if ischar(weight_file)
    weights = load('weights.mat', '-ASCII');
else
    weights = weight_file;
end

if isempty(varargin)
    output_image_location_path = '\Output_Image_File\';
```

```
elseif length(varargin) == 1
    output_image_location_path = varargin{1};
else
    error('Varargin length too long');
end

image_1 = image_maker(weights,node_num,fiber_width_mm,
    intra_fiber_dist_mm,...
                        intra_node_dist_mm,display_height_cm,
                            display_width_cm,...
                        height_in_pixels,width_in_pixels,node_buffer_coeff
                            );

sized_image = image_sizer(image_1,height_in_pixels,width_in_pixels);

colored_image = image_color(sized_image,'white');

filename = image_saver(colored_image,output_image_file_name,
    output_image_location_path);
```

Bibliography

[1] T. Simpkins, *Design, Modeling, and Simulation of a Compact Optoelectronic Neural Coprocessor.* PhD thesis, Massachusetts Institute of Technology, 2005.

[2] E. Krandel, J. Schwartz, and T. Jessell, *Principles of Neural Science.* McGraw-Hill, 2000.

[3] R. Beale and T. Jackson, *Neural Computing: An Introduction.* Institute of Physics Publishing, 1990.

[4] R. Lippmann, "An Introduction to Computing with Neural Nets," *ASSP Magazine, IEEE*, vol. 4, no. 2, pp. 4–22, 1987.

[5] C. Bishop, *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[6] T. Simpkins, J. M. Perkins, C. Fonstad, and C. Warde, "Compact Optoelectronic Integrated Neural Coprocessor: Architecture, Components, and Training." To be submitted to IEEE Transactions on Neural Networks, 2009.

[7] D. Griffiths, *Introduction to Electrodynamics.* Prentice-Hall, 1999.