WIRELESS INFRARED POWER

An Undergraduate Research Scholars Thesis

by

JOSEPH HARRIS

Submitted to the Undergraduate Research Scholars program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. Sunil Khatri

May 2019

Major: Electrical Engineering

TABLE OF CONTENTS

ABSTRACT

Wireless Infrared Power

Joseph Harris
Department of Electrical Engineering
Texas A&M University

Research Advisor: Dr. Sunil Khatri
Department of Electrical and Computer Engineering
Texas A&M University

We are creating an infrared (IR) beam using LEDs and lenses to power devices wirelessly. All electrical devices we use today are limited by either being hardwired or using batteries to power them. This technology may eliminate the need for devices to be constrained by wires or battery life. There are similar technologies that are currently being brought to market, but they use IR laser technology. The use of IR lasers is more dangerous and costlier. By using LEDs and lenses, we expect to make unit less expensive and safer for everyday use.

Unfortunately with a large LED light source, a simple and inexpensive lensing configuration will not be sufficient to collimate and shrink the light source. A much more complex and expensive optics system is needed to collimate a non-ideal light source. The overall power efficiency of the system was 1.5%. To get the desired one watt of power for charging, the light source would need to have 67 watts of input power. That is much more then the system we designed could handle. Ultimately, using a large LED light source to create a small collimated IR beam would be more expensive than using an IR laser.

DEDICATION

To Ashley, Sam, Jeff, Nathan, and Curtis. Without whom I would not be the person I am today.

ACKNOWLEDGMENTS

I would like to thank my research partner Matthew Skolaut. He was a tremendous help with the work and brought a wealth of knowledge to the project. I would like to think that by working with Matthew over the previous months I have added a lifelong friend as well as someone I can collaborate with in the future.

I would also like to thank Professor Sunil Khatri for giving me the opportunity to work with him on this project. Getting to know him and working with him made this endeavor worthwhile. I learned a great deal from him and have a better understanding of how I want to proceed with my life.

Thank you Runze Li for taking the time to discuss our plans with us, and help us see how our goal can be accomplished.

NOMENCLATURE

IR	Infrared
LED	Light Emitting Diode
FL	Focal Length
W	Watts
mW	Milliwatts
in	Inches
ms	Milliseconds
m	Meters
mm	Millimeter
nm	Nanometers
λ	Wavelength
A	Amperes
mA	Milliamps
V	Volts
Ω	Ohm
°	Degree

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

1.1 Introduction and Literature Review

All electronics need a power delivery system. That power is delivered either through a direct hardwiring to a power source or from a battery that will eventually need to be plugged in and recharged. Both of these two power sources have limitations. A hardwired device has limited mobility since it needs to be connected to a power source. A device that operates on battery power is limited by the amount of time it can operate without being recharged, at which point it will need to be plugged in. Our goal is to eliminate the need for devices to be plugged in to a power source for power or recharging. By creating a transmitter and receiver that work in the infrared spectrum, it is possible to wirelessly power devices.

Two main sources of research have emerged in this field. A company named Wi-Charge is creating a transmitter and receiver system to power devices wirelessly, and The University of Washington published a paper on the topic in 2017. Wi-Charge is using an open-air laser system to deliver infrared power wirelessly [1]. The problem with their system is that they are creating a high-power laser beam that needs to be carefully controlled due to its potential to injure the public. Wi-Charge's system also uses costly custom retroreflectors to create the open-air laser [2]. Additional research into wireless charging has been done by the University of Washington [3]. Like Wi-Charge, the research team used lasers to transmit the power wirelessly. They mostly focused on the ability to transmit power, and not creating a practical device for consumer use. The previous work of The University of Washington and the Wi-Charge has proven that the concept of delivering power wirelessly is possible. Both prior attempts at this technology use infrared lasers to transmit power. The issue with using lasers are the high cost of the lasers and the possibility of harm to the public.

1.2 Hypothesis

The research we are proposing is the creation of an infrared transmitter that does not utilize lasers to transmit the infrared signal. By avoiding the use of lasers, we can create a product that is safer and more cost-effective than the product currently offered by Wi-Charge.

The transmitter we are designing uses an array of infrared LEDs and a series of lenses to produce a light beam. This approach allows us to control the density of power in the light beam and keep it in a range that can operate safely in the public square.

The infrared beam will transmit power from the source to the receiver where it will be converted to electrical power. The power transferred to the receiver can then charge the batteries of devices that are attached.

Previous research has shown that transmitting and collecting infrared power across a room is possible using lasers. I am researching whether it is possible to transmit enough power across the room to recharge devices without using lasers. With this research, we can solve the problem of wireless power transfer with a safer, more cost-effective technology. This will create competition in a burgeoning field and show that the technology is a worthwhile investment.

1.3 Methodology

We are approaching the problem by breaking our device into subsystems and tasks. The first task is to run some ideal condition experiments by converting electrical power to optical and then collecting the optical output and converting it back to electrical power. Running these experiments can quickly test several light sources and receivers and determine which produce the best power conversion results. Once we have those results, we can use the proper components in the future experiments.

The first subsystem is the LED array and collimating lenses. Approaching this subsystem first is necessary for the following systems. Using the LED source, and the observed power efficiency from the power conversion experiments, we can create an LED array that outputs sufficient power to charge a device. We then need to create a series of lenses that will shrink and collimate the LED source to the desired output.

Once the LED source and collimating lenses subsystem is completed, we can run a series of tests to determine the power delivered across a room. This is where we will get our first true results from our research. The results will tell us if we are able to transfer sufficient power across the room to charge a device like the previous research showed using infrared lasers. We will reproduce this step with different sources, if necessary, to reach our desired power output. Once we have obtained sufficient output, we will move onto the receiver detection subsystem and the beam placement subsystem.

The receiver location subsystem will locate the receiver by detecting infrared light reflected from retroreflectors attached to the receiver. The infrared light from the retroreflector will be transmitted from an infrared camera and infrared flashes attached to a single board computer. Using a computer learning program, OpenCV, on the single board computer, the reflected light from the retroreflector will be detected and located in the image. The location of the receiver will then be sent to another program that controls two servo motors with gold mirrors attached. Each motor will control the movement of a mirror that will change the position of the infrared beam. One motor will control the x-axis of the beam position, and the other motor will control the y-axis of the beam position. Combined the two motors will move the infrared beam toward the receiver. Once the motors are in position, a test pulse will be sent to the receiver. If the test beam is delivered to the receiver successfully, the receiver will send a radio transmission back to the transmitter and the charging beam will then proceed to be sent.

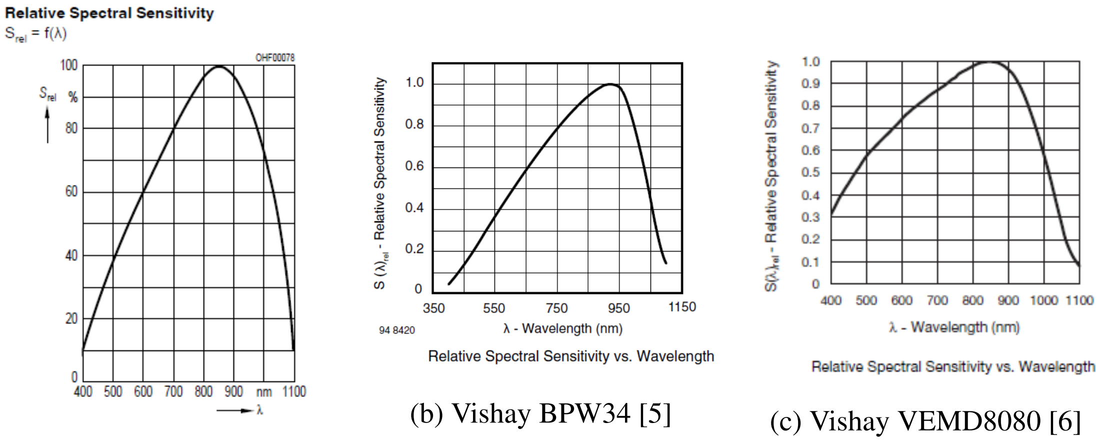
The receiver of this system consists of a solar cell, retroreflector, a low power radio and voltage boosting circuitry. The solar cells convert the infrared signal back to electrical power. The retroreflectors are used to locate the receiver in a room. The low power radio is used to communicate with the transmitter. The voltage boosting circuit is used to convert the power received from the solar cells to a level that can be used to charge the batteries of the device.

By performing the experiments and creating the subsystems listed above, we hope to show that transmitting wireless infrared power is possible without having to use high powered and expensive lasers.

2. SUB-SYSTEMS AND EXPERIMENTS

2.1 Power Conversion Experiments

Our research started by determining the components we would use in our device. We started by reviewing our infrared (IR) light sources. There are two common wavelengths for IR light emitting diodes (LEDs), 850nm and 940nm. To decide which wavelength LED to use, we looked at the photodiodes we were considering for the receiver. Originally, we considered three different photodiodes. One of the specifications on the datasheets for the photodiodes is the relative spectral sensitivity, the efficiency of detecting light dependent upon the wavelength. In Figure 2.1 below you can see that two of the three spectral sensitivities peak at 850nm and the third peaks at 900nm. For this reason, we chose to use the 850nm LEDs. We then had to narrow down which 850nmn LED we would use. Because of their inexpensive cost, we bought two different types and chose to run experiments to determine which LED best suited our desired outcome.



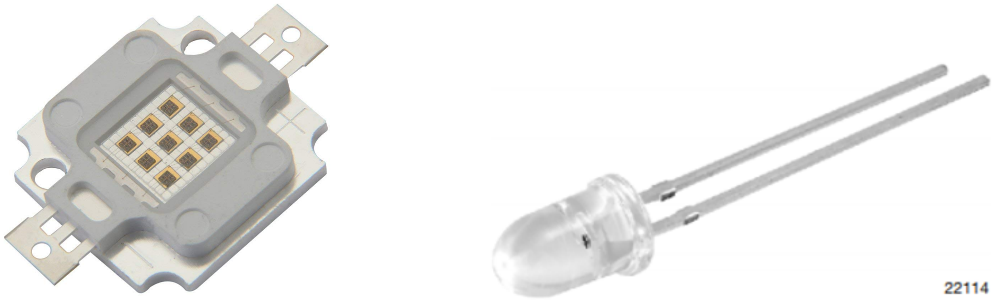
(a) Osram SFH206K [4]

(b) Vishay BPW34 [5]

(c) Vishay VEMD8080 [6]

Figure 2.1: Photo Diodes Spectral Sensitivity Plots

We narrowed the choice down to two 850nm LEDs. The first LED is a high-speed low-power model made by Vishay, see Figure 2.2b. The other is a high-power model made by Chanzon, see Figure 2.2a.



(b) Vishay VSLY5850 [8]

(a) Chanzon 10W [7]

Figure 2.2: 850nm Infrared LEDs

To determine which LED would work best in our system we setup an experiment to measure the power efficiency of the components. We calculated the efficiency, Equation 2.1, of the components by measuring the power provided to the LED and then measuring the power output from the photodiode.

$$\eta = \frac{Power from Photodiode}{Power into LED} \tag{2.1}$$

Several different configurations were measured in the initial set of tests. The tests were conducted for ideal results. There was virtually no distance between the LED and the receiver. The primary purpose of these tests was to determine which LED to proceed with, and to determine whether it was possible to transfer power with a high enough efficiency to make this method of power delivery feasible. We wanted to produce 1 watt of power from the receiver, so we were hoping to get close to 5% efficiency. With that level of efficiency, we could use a 20W LED source and get the 1 watt of power output we desired.

As you can see in Table 2.1 and Table 2.2 the test did not produce the desired efficiency. We did observe that when the photodiodes were in parallel, the larger area produced more output power. This made us rethink our use of photodiodes as receivers, and look for a component with a larger surface area.

Table 2.1: Power Tests Using High Power LED & Photodiode Receivers

Receiver Configuration	Power In [W]	Power Out [mW]	Efficiency [%]
1 photodiode	4.05	13.44	0.33%
2 photodiodes in series	4.05	14.3	0.35%
2 photodiodes in parallel	4.05	26.65	0.66%
4 photodiodes in parallel	4.05	44.5	1.10%

Table 2.2: Power Tests Using High Speed LED & Photodiode Receivers

Receiver Configuration	Power In [W]	Power Out [mW]	Efficiency [%]
1 photodiode	1.8	5.9	0.33%
2 photodiodes in series	1.8	3.5	0.19%
2 photodiodes in parallel	1.8	6.84	0.38%
4 photodiodes in parallel	1.8	8.28	0.46%

This first set of experiments lead us to look for a different component for the receiver. We ultimately decided to use a solar cell. We originally tried to use a large 56mm x 74mm solar panel. The four individual solar cells were wired in series, so if they were lit unevenly the unlit solar cells acted like resistors and drew power from the circuit. We disconnected the panels from each other and used a single solar cell in the circuit. We ran the experiment using a single solar cell, 56mm x 17mm in size, in the circuit instead of the photodiode. You can see in Table 2.3 that this test produced a result that was better than our desired outcome of 5% efficiency.

Table 2.3: Power Tests Using Solar Cell as the Receiver

LED Configuration	Power In [W]	Power Out [mW]	Efficiency [%]
High Power	4.14	182.66	4.41%
High Speed	1.24	23.03	1.86%
High Speed	0.512	39.1	7.64%

Another thing we observed from these experiments was the lowering of the efficiency due to heat production. You can see in Table 2.3 that when we lowered the current to the high-speed LED, voltage in both cases is 1.6V, the efficiency increased fourfold. This is due to a loss of power from heat production when the current was at 0.7A. By decreasing the current to 0.32A, we lost less power to the production of heat and increased our efficiency.

2.2 Beam Forming Sub-System

The goal of the beam forming sub-system is to create a narrow-collimated beam of infrared light that can be directed around the room toward the receiver. We planned to accomplish this by creating an array of high-speed LEDs and then having the light produced from the LEDs proceed through a series of lenses. The first lens the light reaches is the aspheric lens. By placing the LED array at the focal point of the lens, f_0, the output of the lens is then collimated. The size of the collimated light beam is given by the relationship in Figure 2.3 and Equation 2.2 [9]. Using this equation, you can take the initial size of the LED array, 34mm, the focal length of the lens, 26mm, and the degree of divergence, 3°, and calculate the final size of the collimated beam.

$$f_0 = \frac{r}{tan\theta_0} \tag{2.2}$$

$$34mm + 2[26 * tan(3^o)] = 36.73mm$$

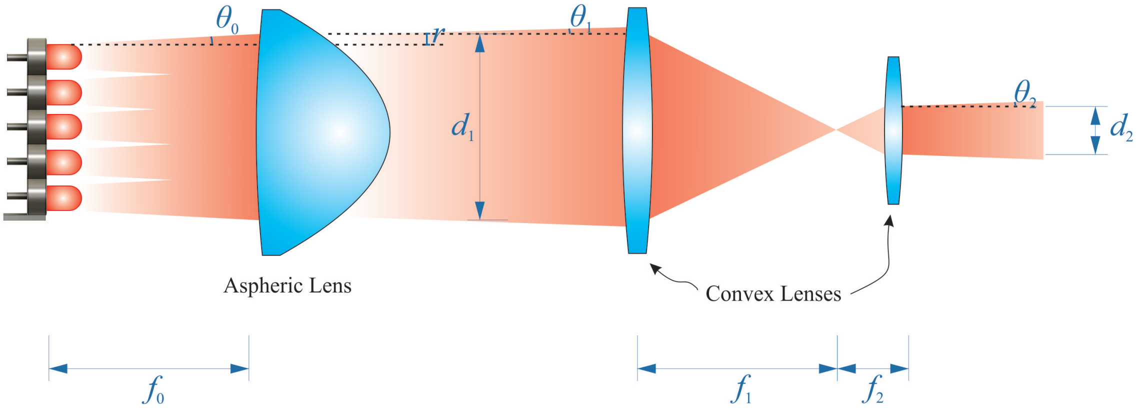


Figure 2.3: Lens Configuration

The collimated light then passes through a series of convex lenses. The pair of convex lenses shrinks or increases the diameter of the light beam depending on the ratio of the lens's focal lengths. Since we are shrinking the beam, f_2 needs to be much smaller than f_1. You can see this from Equation 2.3 [9]. Given the collimated beam size of 36.73mm and the focal lengths of f_1 and f_2 are 250mm and 30mm respectively, we can calculate the final beam size.

$$d_2 = d_1 \frac{f_2}{f_1} \tag{2.3}$$

$$36.73mm * \frac{30}{250} = 4.41mm$$

If you had a light source with a divergent beam, then you would need to use Equation 2.4 [9] in the final beam size calculations to obtain the degree of divergence. Since we are collimating our light source first our initial degree of divergence θ_1 is approximately zero. Therefore, when you plug zero into Equation 2.4 for θ_1 the result is zero.

$$\theta_2 = \theta_1 \frac{f_1}{f_2} \tag{2.4}$$

$$0 * \frac{250}{30} = 0$$

This creates a beam approximately half a centimeter in size to be projected across the room. With zero degrees of divergence, the beam should not increase in size as it is projected onto objects further and further away. This ensures that all of the light focused into the beam will land on the solar cells of the receiver, and therefore collecting all the transmitted power.

2.3 Receiver Locating Sub-System

The locating algorithm will incorporate the use of aruco markers to find and position the receiver in the room. We are using a Raspberry Pi with an infrared camera to run the locating algorithm, see Figure 2.4. The program takes a picture and locates the aruco marker in the picture using OpenCV. Using OpenCV, an open source computer vision software library, gives us the coordinates of the corners of the aruco marker in the picture. This information is sent to the gimbal mirror setup to position the mirrors. An aruco marker can also determine the orientation of the receiver, see Figure 2.5b. Using the orientation of the receiver the location of the solar cells that need to be targeted by the IR beam will be known.

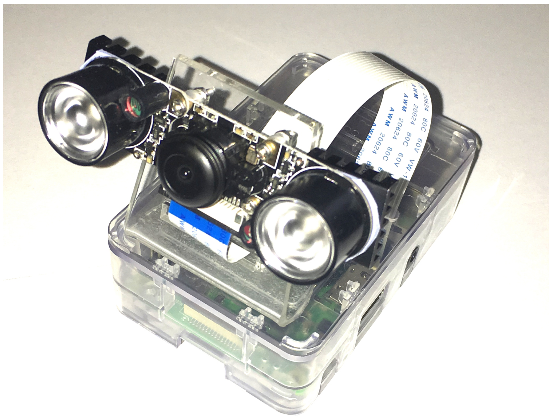


Figure 2.4: Locating Camera

The IR camera on the Raspberry Pi also has a visible light filter over the lens. We are working to determine the best filter. The first option is a 780nm wavelength filter for a camera. That filter lets a fair amount of light through, and although you can see the image clearly it is tinted purple. The other option for a filter is the film inside an old 3.5in floppy disk. The magnetic material, which was originally used to store data, can be used as a filter that only allows infrared light through. The images from this filter makes everything look black except for objects reflecting infrared light. We are working to determine if this filter will work well with the aruco marker at a distance.

The aruco markers are made from infrared retroreflectors with black vinyl applied on top to create the marker pattern, see Figure 2.5a. The IR retroreflector is illuminated by the IR flashes on the camera and light up bright white in the images. We are trying various sizes for the markers ranging from 1in to 2in in width. The idea is to maximize the distance of recognition while limiting the obtrusiveness of the marker.

retroreflector
23

(a) Aruco Marker [10]

(b) Aruco Marker with Directional Axis [10]

Figure 2.5: Aruco Markers

2.4 Beam Movement Sub-System

A gimbal mirror arrangement places the beam in the room. A gimbal mirror setup is a pair of mirrors attached to two motors that move independently of one another and control either vertical or horizontal movement, see Figure 2.6. The servo motors are controlled by the Raspberry Pi. The program controlling the servos receives the location of the aruco marker in the form of X-Y pixel coordinates. A mapping is then done to match the pixel location in the image to the angle of the servo motors to direct the IR beam to that point.

The target solar cell size is 39mm x 57mm and the beam size is 4.41mm so there is some room for error in the mapping.

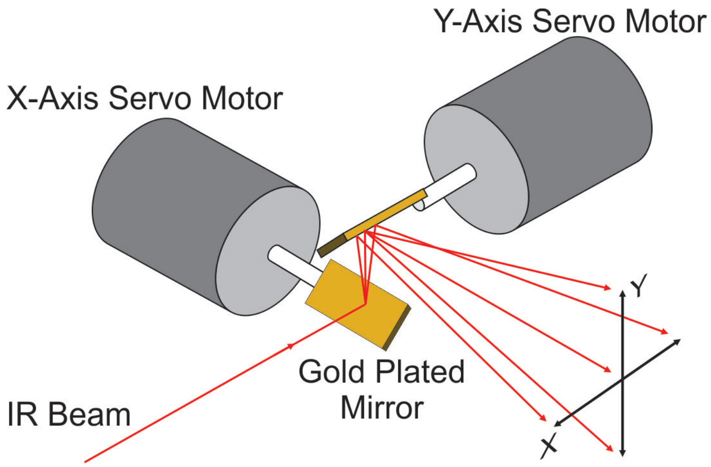


Figure 2.6: Gimbal Mirror Configuration

2.5 Receiver Sub-System

Because this is a new method of delivering power, no devices currently have the necessary components integrated into them. This requires us to create a device that will bridge the new power delivery system to existing devices. To do this we conceived a phone case, see Figure 2.7, that will house the receiver components and connect to the phone to deliver the transmitted power. The receiver consists of a phone case, solar cells, the aruco marker, and a boost converter. The solar cells are the target of the IR beam, and convert the optical power of the beam to electrical power. The aruco marker is used for locating the receiver in the room, see Section 2.3. The boost converter takes the power from the solar cell and converts it to 5 volts, a USB standard [11].

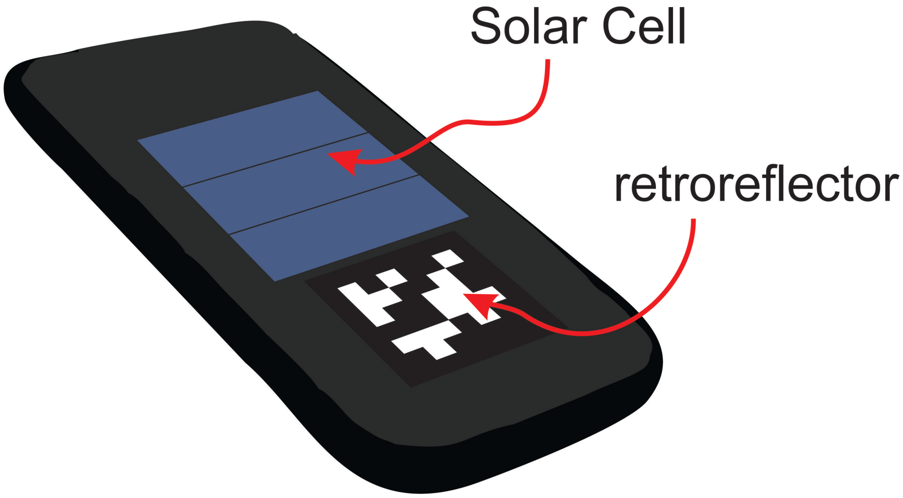


Figure 2.7: Receiver Case

3. RESULTS

3.1 Lensing Results

Using the focal lengths of the lenses, see Table 3.1, and the configuration in Figure 2.3 we arranged the lenses and LED light source in our system, see Figure 3.1. Using this configuration we hoped to get a collimated and reduced light source at the far end of the system.

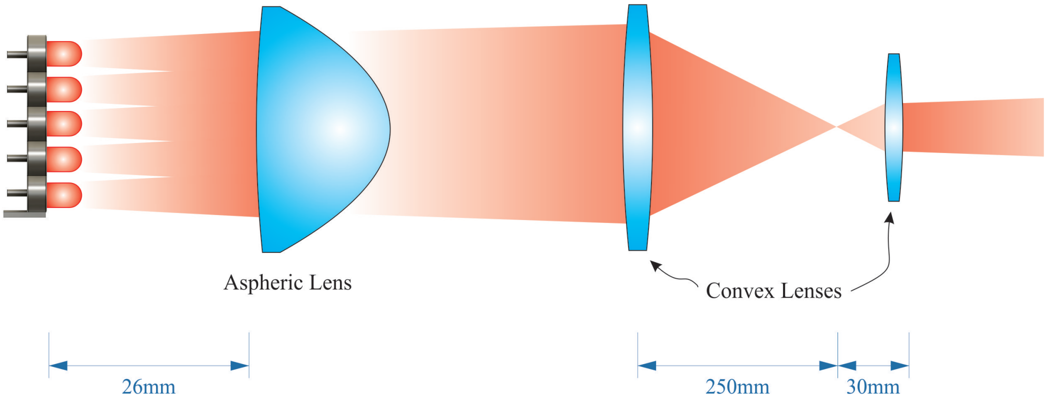


Figure 3.1: Lens Configuration

Table 3.1: Lens Specifications

Lens Type	Diameter [mm]	Focal Length [mm]	Back Focal Length [mm]
Aspheric	50	40	25.92
Convex	50	250	246.74
Convex	30	30	26.13

The results of the lensing array were disappointing and unexpected in light of our initial calculations. When we put our LED light source at the focal length of the aspheric lens, we observed a large divergence, see Table 3.2 and Equation 3.1, when we were expecting to observe little to none. This divergence disrupts the operation of the remaining lenses. From Equation 2.4, any divergence coming out of the aspheric lens will be multiplied by the ratio of the focal lengths of the first lens over the second lens. Since we are shrinking the beam size, that ratio is larger than one and increases the angle of divergence even further.

Table 3.2: LEDs Through Aspheric Lens Divergence Measurements

Image Size [mm]	Distance from Source [mm]
80	100
135	190

$$\Theta = 2 * tan^{-1}\left(\frac{(s_2 - s_1)/2}{d_2 - d_1}\right) \tag{3.1}$$

Using the measured image sizes, *s*, and the distances from the source, *d*, see Table 3.2, the angle of divergence can be calculated using Equation 3.1.

$$\Theta = 2 * tan^{-1}\left(\frac{(135 - 80)/2}{190 - 100}\right) = 33.98^o$$

After observing these results we did further research to determine the cause of the divergence. We discovered an article by Edwin Diaz and Matthias Knobl that uses Equation 3.2 to describe the divergence of a non-ideal light source [12]. In the equation d_s is the diameter of the light source, *fl* is the focal length of the aspheric lens, and that gives you the full divergence in radians. By multiplying Equation 3.2 by 180/π, see Equation 3.3, the full divergence of the beam is given in degrees.

$$\Theta = \frac{d_s}{fl} \tag{3.2}$$

$$\Theta = \frac{d_s}{fl} * \frac{180}{\pi} \tag{3.3}$$

Using Equation 3.3 we calculated the divergence. It was not exactly what we observed and measured, but it was within a reasonable range. The slight inconsistencies could be the result of measurement errors or optical aberrations with the lens and a non-ideal light source.

$$\Theta = \frac{34}{40} * \frac{180}{\pi} = 48.7^o$$

We wanted to see what the divergence of the aspheric lens would be if we used a nearly ideal point source. We were trying to determine whether we could overcome our divergence issues by changing our light source to a laser. We used an ordinary green laser pointer, with a light source diameter of 3mm, and measured the images after the light passed through the aspheric lens, see Table 3.3.

Table 3.3: Laser Through Aspheric Lens Divergence Measurements

Image Size [mm]	Distance from Source [mm]
11	156
71	660

Using the measurements from Table 3.3 and Equation 3.1, we calculated the angle of divergence of the laser pointer through the aspheric lens. Then using Equation 3.3, we calculated what the expected divergence should be from a non-ideal source. Again, the calculated divergence is not exactly what we observed but close enough that our results are reasonable.

$$\Theta = 2 * tan^{-1}\left(\frac{(71-11)/2}{660-156}\right) = 6.81^o$$

$$\Theta = \frac{3}{40} * \frac{180}{\pi} = 4.3^o$$

We observed and measured the divergence of the laser beam without any lensing. The measurements of the observed beam are in Table 3.4. Using Equation 3.1 we can calculate the divergence.

Table 3.4: Laser Divergence Measurements

Image Size [mm]	Distance from Source [mm]
3	0
4	2300

$$\Theta = 2 * tan^{-1}\left(\frac{(4-3)/2}{2300-0}\right) = 0.0249^o$$

Working backwards from that result, we can calculate the focal length of a lens needed to create that divergence. We want 0.0249° as our final divergence, then we need to use Equation 2.4 and solve for the incoming half divergence.

$$\theta_2 = \frac{0.0249^o}{2} = \theta_1 * \frac{250}{30} \implies \theta_1 = \frac{0.0249^o}{2} * \frac{30}{250} = 0.00149^o$$

Then using Equation 3.3, we can solve for the focal length of the lens needed to achieve this divergence.

$$\Theta = 0.00149^o{*}2 = \frac{34}{fl}{*}\frac{180}{\pi} \implies fl = \frac{1}{0.00149^o * 2}{*}34{*}\frac{180}{\pi} = 6.517x10^5mm = 651.7m$$

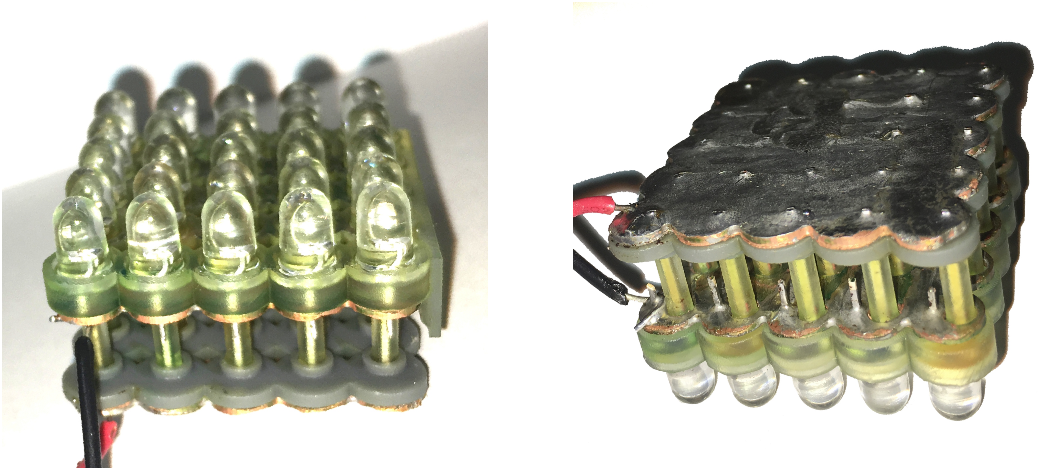
The result of this calculation shows how drastic of a difference between source diameter and focal length is required to come close to approximating our LED light source as an ideal point source. The simple method of using a single lens to collimate our non-ideal light source is not practical. A much more advanced and complicated optical configuration will be needed to obtain the degree of divergence required.

3.2 Power Results

Starting where we left off in chapter 2.1, we built an array of twenty five infrared LEDs. With our previous observation of 7.64% efficiency, see Table 2.3, we hoped to receive 1.146 watts of electrical power out of the solar cells, see calculation below.

$$OutputPower = efficiency * LEDs * \frac{Power}{LED} = 0.0764 * 25 * 0.6 = 1.146W$$

We 3D printed a holder for the LEDs to create an evenly spaced 5x5 array, see Figure 3.2. To try and dissipate heat away from the LEDs, we made two levels of copper plates and soldered a contact from each LED to the plates. Since the LEDs we used are encapsulated in a plastic casing, the contact leads were the best place to try and remove heat from the LEDs. The design of the LED holder allowed air to move over each copper plate, and if the plates radiating into the air was not enough to keep the LEDs cool, we could incorporate a fan to pull the hot air away from the copper plates.



(a) Front View of LEDs

(b) Back View of LEDs

Figure 3.2: LEDs in 3D Printed Holder

We bought a boost converter to increase the voltage from the solar panels to a level we could use to charge a device using a USB cable. The boost converters specified input voltage range was 0.9V to 5V. Testing the boost converter with the variable power supply, we found it would turn on at just under 0.6V. This voltage is out of the specified input range of the converter, but is close to the output voltage of the solar cells. We decided to test and see if the parallel configuration would work with the boost converter. If it did not, we could reconfigure the solar cells into a series connection and increase the voltage to a range that would be in the specified range of the boost converter.

Since we could not form a collimated beam as we planned, see Chapter 3.1, we proceeded to perform the tests as before. We laid the LED array directly on the solar panels to achieve optimal results, see Figure 3.3. The initial tests were done with two solar cells in parallel since that is how we observed our best efficiency results in the initial tests, see Table 2.1 and Table 2.2.

The power test with the solar cells in parallel yielded some unexpected results. We measured the short circuit current and the open circuit voltage to calculate the ideal power

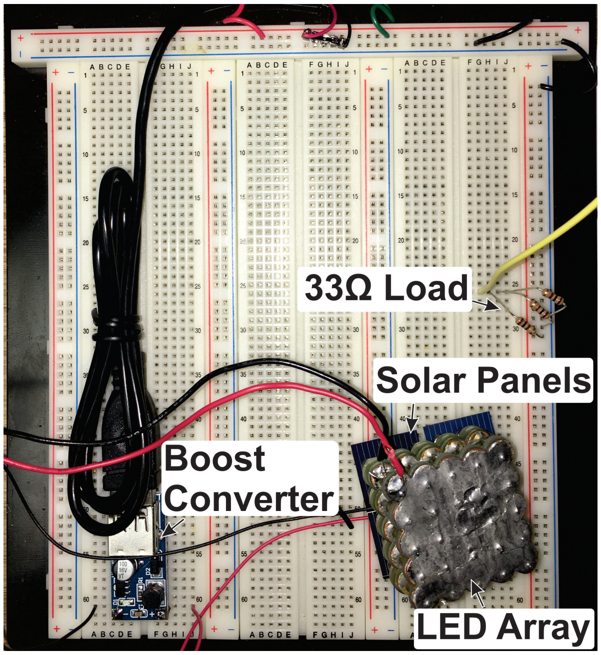


Figure 3.3: Test Setup

from the solar cells. As you can see in Figure 3.4, and in Table 4.1 in Appendix A, the output power saturates after 2 watts of input power. This is due to the current saturating at around 270mA with a voltage of 0.58 volts. This is a problem since we need to supply 15W of power to the LEDs to obtain the 1W of power we require for charging. We decided to change the configuration of the solar cells and see if we could get better results with the panels in series.

The next experiment we did was with a pair of solar cells in series. In the past with photodiodes, see Table 2.1 and Table 2.2, the series connection configuration increased the voltage but decreased our current and overall efficiency. This time we took readings at increasing input current levels and graphed the results. We were able to get more current at a higher voltage with the panels in series, see Figure 3.5, and Table 4.2 in Appendix A, than we were with the panels in parallel. Heat began to build due to the high current flowing through the LEDs. We were unable to sufficiently decrease the temperature of the

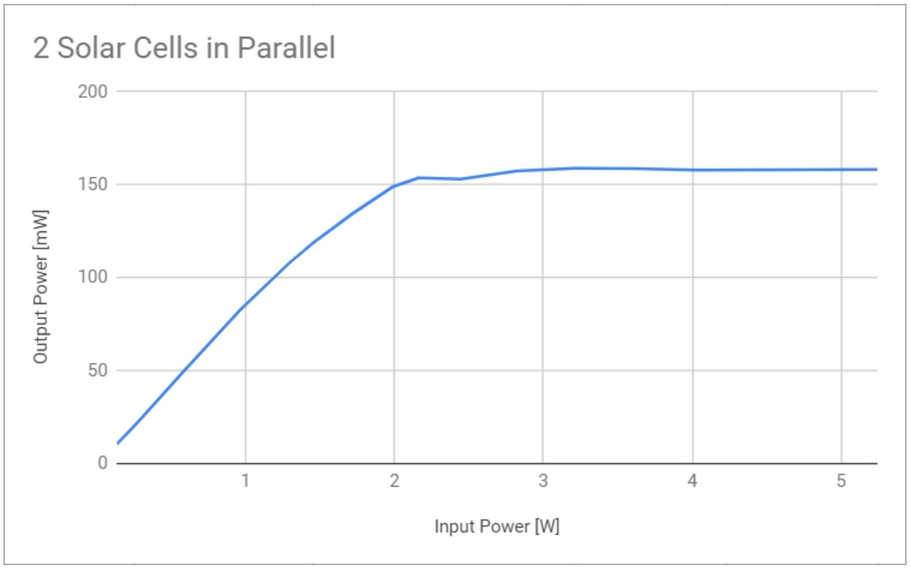


Figure 3.4: Power from 2 Solar Cells in Parallel

LEDs and as a result we were not able to reach the input power goal of 15W.

Connecting the solar cells in series increased the voltage to 1.2V which would allow the boost converter to turn on. We connected the boost converter to the solar panels and added a 33Ω load to simulate charging a battery, see Figure 3.3. We then repeated the previous experiment, starting at zero and increasing the input current 0.25A and recording the output power, see Figure 3.5, and Table 4.3 in Appendix A. Once the boost converter received enough power to turn on, around 3W, the output power increased linearly. The output power also decreased significantly when a load was applied to the system. We reached a maximum of 0.15W of output power, which is significantly less then the 1W we were hoping to achieve.

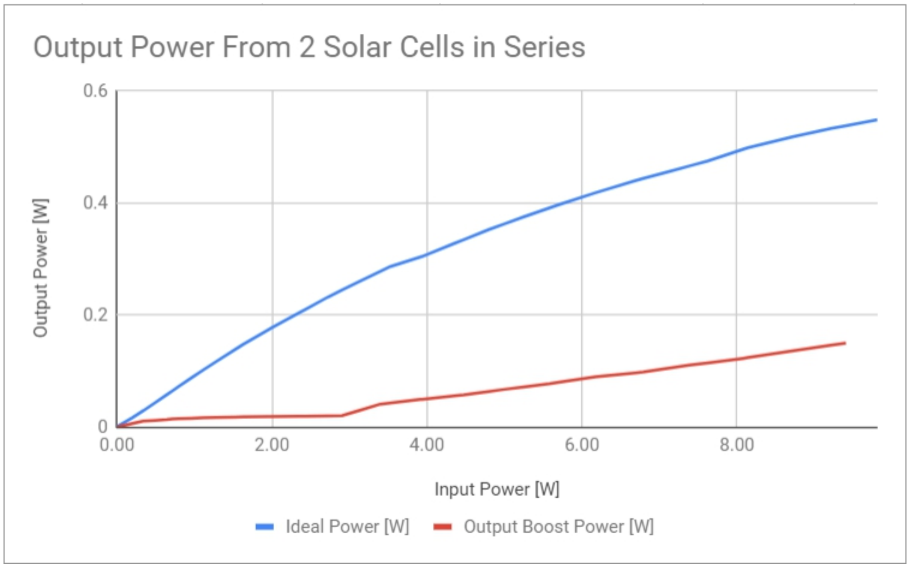


Figure 3.5: Power from 2 Solar Cells in Series

3.3 Locating Results

The first version of the locating algorithm we designed was a blob detector, see Appendix B. The blob detector applied a series of filters to the original image, see Figure 3.6, to extract the brightest spots in the image. Since the targets were IR retroreflectors, they would reflect the light from the IR camera's infrared flashes and show up as bright white spots in the images, see Figure 3.6. A threshold is applied to the image so only the brightest spots are kept. You can see in Figure 3.7 that the three uniform shaped blobs are the retroreflectors, and the remaining irregular shaped blobs are other bright parts of the image that made it through the thresholding.

The next step is to count the number of pixels in a blob. Since a retroreflector is located on the back of a dark case, see Figure 2.7, the blob will be isolated from other bright spots. By increasing the size of the blobs by a small amount small groups of bright spots can be combined into a larger bright spot. Then by counting the number of pixels in a blob and limiting the number of pixels in a blob considered a target, the large irregular shaped blobs

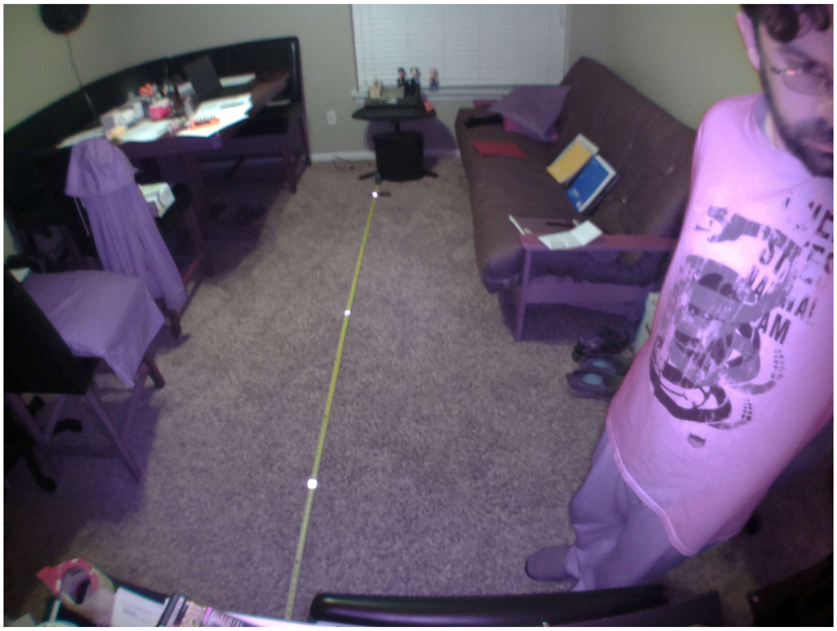


Figure 3.6: Original Image

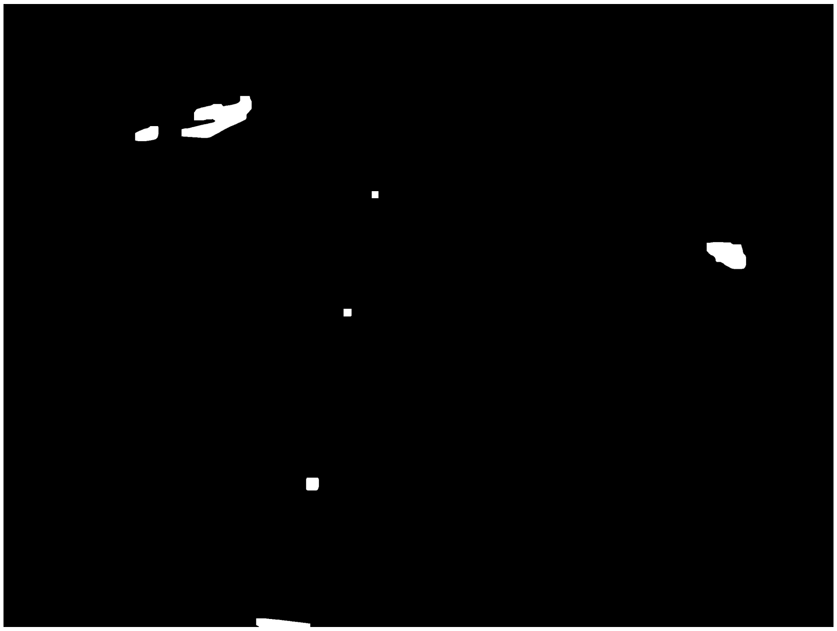


Figure 3.7: Blob Detection - Threshold Image

can be removed from consideration. The result is the detection of the three uniform blobs in the image, see Figure 3.8, and labeling them as receiver locations. The program then displays and saves the X and Y coordinates of the centers of each blob, see Figure 3.9.

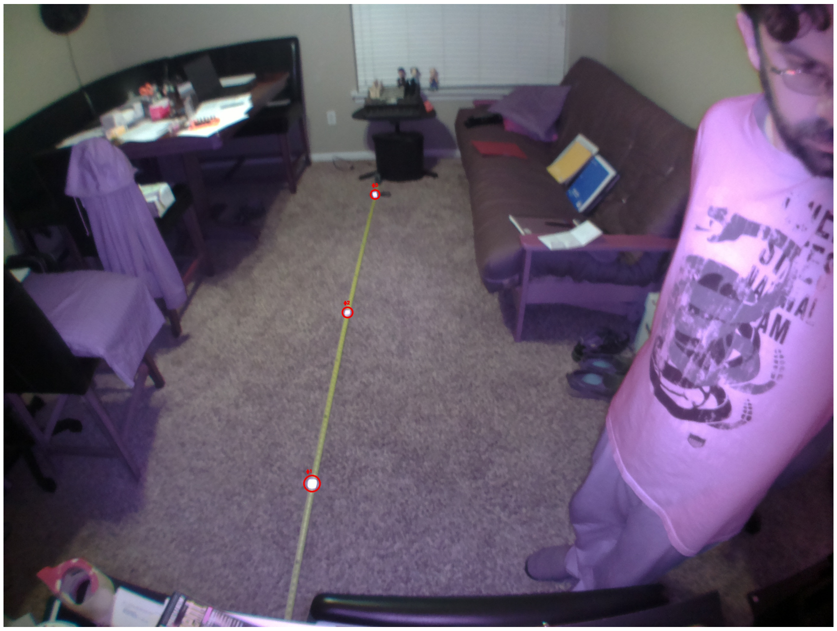
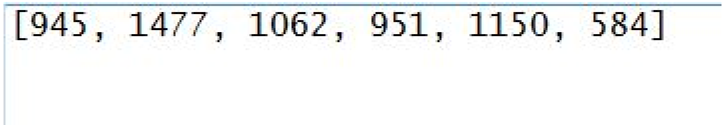


Figure 3.8: Blob Detection - Final Output Image



```
[945, 1477, 1062, 951, 1150, 584]
```

Figure 3.9: Blob Detection - Final Output Centers

The blob method worked rather well. It was able to detect receivers up to ten feet away, 3.05m. Although it did give a lot of false positive detections, see Figure 3.10. We did our best to adjust to the settings to keep the false detections to a minimum, but they were never completely eliminated.

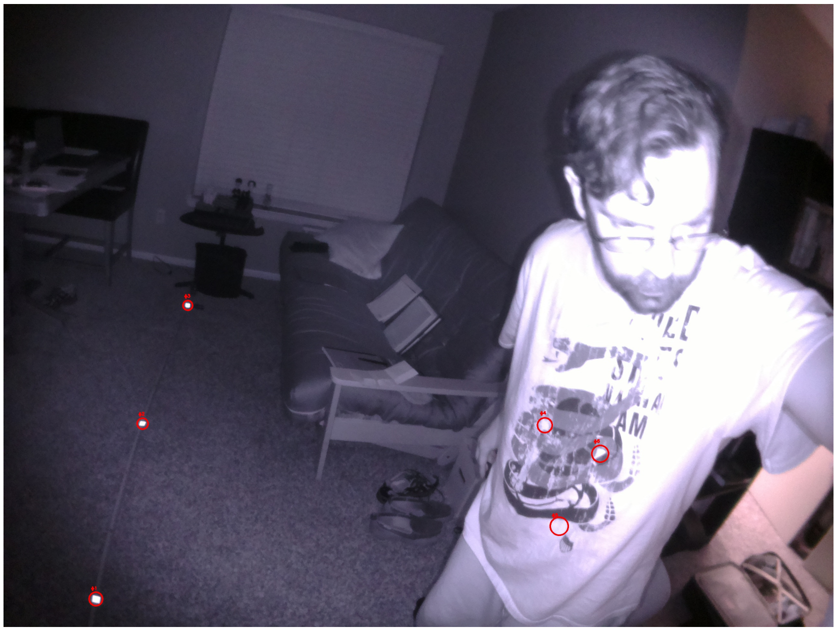


Figure 3.10: Blob Detection - Erroneous Detection

To attempt to eliminate false positive detection of receivers, we started to use aruco markers, see Chapter 2.3 and Appendix C. Aruco detection is more robust than blob detection, and less susceptible to erroneous detection of receivers. The challenge was to get the aruco marker detected at the distance we required without making the target too large. We limited our aruco marker size to two inches, so it would fit on the back of the phone case with the solar cells, see Figure 2.7. At that size, we were able to detect the aruco marker up to 10 feet away, see Figure 3.11, just like with blob detection. The program for aruco detection outputs the corners of the detection box, see Figure 3.12. We can send that information to Beam Movement Sub-System, see Chapter 2.4, to direct the beam to the receiver.

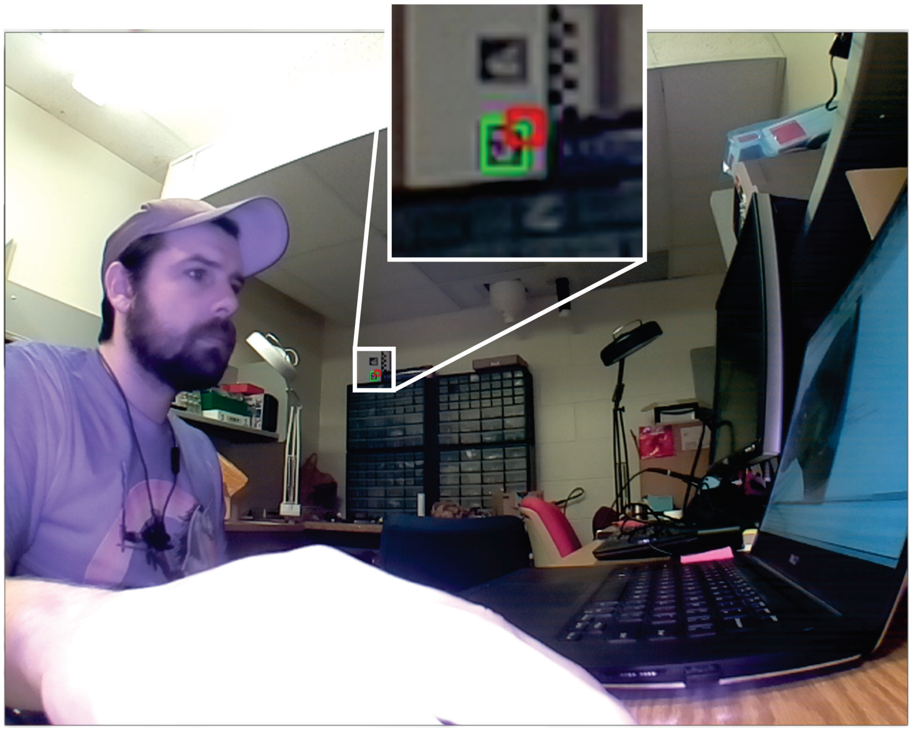
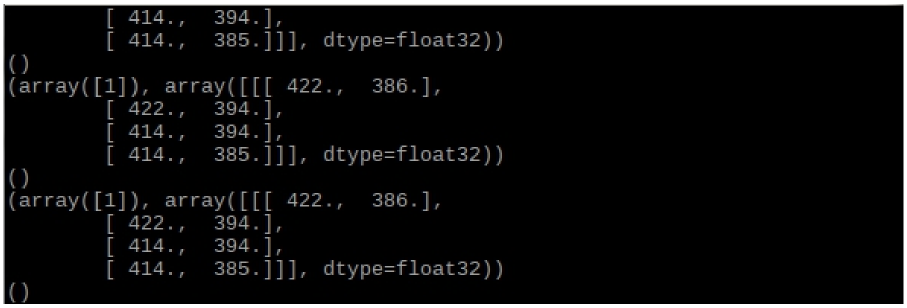


Figure 3.11: ArUco Detection



```
        [ 414.,  394.],
        [ 414.,  385.]]], dtype=float32))
()
(array([1]), array([[[ 422.,  386.],
        [ 422.,  394.],
        [ 414.,  394.],
        [ 414.,  385.]]], dtype=float32))
()
(array([1]), array([[[ 422.,  386.],
        [ 422.,  394.],
        [ 414.,  394.],
        [ 414.,  385.]]], dtype=float32))
()
```

Figure 3.12: Aruco Corner Output

4. SUMMARY AND CONCLUSIONS

4.1 Challenges

The lensing portion of this project ended up being much more complicated than we originally expected. Working with non-ideal light sources was the biggest challenge. There were no simple equations that described how the light would interact with the lenses. All of the equations that we had would breakdown since our light source was so large. We spoke with professionals in the optics field, thank you Runze Li for helping us, and we discovered that a much more complex optics system would be needed to work with a large light source. Because this would need to be designed and specially created for this project, the cost would be considerable. That answered our question as to whether we could accomplish creating a small collimated light source from LEDs for less cost than a laser. In short, the answer is no we could not. Another consideration is that a more complicated optics system would lead to a loss of power every time our beam passed through a lens. With a complicated optics system, approximately 15 lenses like a camera zoom lens system, we would be losing five times the power than with a 3-lens system we were proposing.

The other challenge with this project was the power conversion. Our initial tests of the solar panels showed about a 7 percent power efficiency. This was done by measuring the short circuit current and the open circuit voltage of the solar cells and calculating the power from those measurements. Once a 33Ω load was applied only 12 percent of that ideal power is usable for charging. That means we end up with about a 1.5 percent power efficiency overall. Our source input would need to be approximately 67 watts to get 1 watt out the other end to use for charging. We would need to increase the LED array from 25 LEDs to 112 LEDs. That only complicates our first problem more, of having too large of a light source.

4.2 Further Study

We are going to continue to work on this system and try and get a working prototype made. The original idea of using LEDs as a light source will be abandoned. We will proceed with using an IR laser diode as our light source. Using the laser diode will eliminate the entire optics component of this project. We will have to be careful not to infringe on the patents of Wi-Charge, but they are creating an open air laser between the transmitter and receiver and we will be using a laser diode to create the laser [2]. We may need some optics to expand the size of the laser beam, but that is a known and solvable problem. That is the opposite of the optics problem we were trying to solve, and since we are using a laser, we can use the ideal light source equations and models.

To solve the power conversion problem we are going to use several smaller photodiodes connected in series to increase the output voltage. We will then use a buck converter to stepdown the voltage to 5 volts. Again, this is the opposite of what we tried to do initially. Since we were operating the boost converter at the very edge of its operation range, we were not getting optimal output. Using the buck converter to step down the voltage should give us a better efficiency than the boost converter did.

We will finish integrating the locating and beam placement software and hardware. The motors we are using give us angular position feedback. The plan is to construct the permanent positioning of the camera and the mirrors. By moving the mirrors by hand, we can align the beam output to the corners of the image provided by the camera. Since the motors give us angular feedback, once we moved the mirror into position by hand, we can record the mirror angles that cover the entire image. Once those angles are recorded, we can perform a mapping of the image and relate the angle to the pixel position in the image. This is how we will integrate the locating and beam placement hardware and software.

REFERENCES

[1] A. Ortal, “Directional Light Transmitter and Receiver.” U.S. Patent 9,705,606, 2017.

[2] A. Ortal and R. Paschotta, “Wireless Laser System for Power Transmission Utilizing a Gain Medium Between Retroreflectors.” U.S. Patent 8,525,097, 2013.

[3] V. Iyer, “Charging a Smartphone Across a Room Using Lasers,” *Proceedings of the ACM on Interactive, Mobile, and Ubiquitous Technology*, vol. 1, no. 4, 2017.

[4] Osram Opto Semiconductors, “Silicon PIN Photodiode, SFH 206 K.” SFH 206 K Datasheet, Apr 2016.

[5] Vishay Semiconductors, “Silicon PIN Photodiode, BPW34.” BPW34 Datasheet, Jul 2008. [Revised: September 2008].

[6] Vishay Semiconductors, “Silicon PIN Photodiode, VEMD8080.” VEMD8080 Datasheet, Feb 2017. [Revised: April 2018].

[7] Chanzon, “Chanzon High Power LED Chip 10W Infrared.” [Online], 2019. [Accessed: 23-Oct-2018].

[8] Vishay Semiconductors, “High Speed Infrared Emitting Diode, 850 nm.” VSLY5850 Datasheet, Mar 2013. [Revised: February 2017].

[9] Newport.com, “Focusing and Collimating.” [Online], 2019. [Accessed: 20-Nov-2018].

[10] OpenCV.org, “Detection of ArUco Markers.” [Online], 2017. [Accessed: 20-Jan-2019].

[11] USB-IF, “USB Power Delivery.” [Online], 2012. [Accessed: 20-Oct-2018].

[12] E. Diaz, M. Knobl, “Prototyping illumination systems with stock optical components,” *Photonik International*, pp. 24–27, March 2012.

APPENDIX A

Table 4.1: Two Solar Cells in Parallel

Input Voltage (V)	Input Current (A)	Input Power (W)	Output Voltage (V)	Output Current (mA)	Output Power (W)	Power Efficiency (%)
1.34	0.1	0.13	0.50	20.84	0.010	7.85
1.37	0.2	0.27	0.53	41.69	0.022	8.13
1.42	0.4	0.56	0.55	86.57	0.048	8.52
1.48	0.65	0.96	0.57	144.47	0.082	8.59
1.52	0.85	1.29	0.57	187.28	0.107	8.36
1.53	0.95	1.45	0.57	206.3	0.118	8.17
1.56	1.1	1.71	0.57	232.76	0.134	7.83
1.59	1.25	1.98	0.57	257.7	0.148	7.49
1.6	1.35	2.16	0.58	265.04	0.153	7.12
1.63	1.5	2.44	0.57	264.8	0.153	6.26
1.66	1.7	2.82	0.57	271.9	0.157	5.58
1.69	1.9	3.21	0.57	275.3	0.158	4.95
1.72	2.1	3.61	0.57	276.03	0.158	4.39
1.75	2.3	4.02	0.57	276.6	0.157	3.92
1.84	2.85	5.24	0.57	277.5	0.158	3.02

Table 4.2: Two Solar Cells in Series

Input Voltage (V)	Input Current (A)	Input Power (W)	Output Voltage (V)	Output Current (mA)	Output Power (W)	Power Efficiency (%)
1.23	0	0	0.61	0.60	0.000	0
1.35	0.15	0.20	1.03	16.02	0.016	8.19
1.38	0.25	0.35	1.08	27.07	0.029	8.5
1.41	0.35	0.49	1.11	38.84	0.043	8.75
1.43	0.45	0.64	1.13	50.79	0.057	8.93
1.45	0.55	0.80	1.14	63.11	0.072	9.06
1.47	0.65	0.96	1.15	75.3	0.087	9.11
1.51	0.85	1.28	1.17	99.64	0.116	9.11
1.55	1.05	1.63	1.18	124.4	0.147	9.04
1.59	1.25	1.99	1.18	148.0	0.176	8.86
1.62	1.45	2.35	1.19	170.9	0.203	8.67
1.65	1.65	2.72	1.19	194.1	0.231	8.51
1.69	1.85	3.13	1.19	217.0	0.259	8.30
1.72	2.05	3.53	1.19	239.0	0.286	8.12
1.75	2.25	3.94	1.19	254.3	0.304	7.73
1.78	2.45	4.36	1.19	275.2	0.328	7.52
1.81	2.65	4.80	1.19	295.9	0.352	7.34
1.84	2.85	5.24	1.18	315.3	0.375	7.15
1.87	3.05	5.70	1.18	334.5	0.397	6.96
1.90	3.25	6.18	1.18	352.9	0.418	6.77
1.94	3.45	6.69	1.18	371.8	0.440	6.57
1.96	3.65	7.15	1.17	387.8	0.457	6.39
1.98	3.85	7.62	1.17	404.1	0.474	6.23
2.01	4.05	8.14	1.17	424.2	0.498	6.12
2.05	4.25	8.71	1.17	441.6	0.518	5.94
2.07	4.45	9.21	1.17	455.6	0.533	5.78
2.11	4.65	9.81	1.16	470.2	0.548	5.59
2.13	4.85	10.3	1.16	483.3	0.561	5.43
2.16	5.05	10.9	1.15	495.6	0.574	5.26
2.19	5.25	11.5	1.15	506.1	0.585	5.08
2.22	5.45	12.1	1.15	521.5	0.601	4.97

Table 4.3: Two Solar Cells in Series with Boost Converter and 33Ω Load

Input Voltage (V)	Input Current (A)	Input Power (W)	Output Voltage (V)	Output Current (mA)	Output Power (W)	Power Efficiency (%)
0	0	0	0	0	0	0
1.36	0.25	0.34	0.564	18.3	0.010	3.03
1.43	0.50	0.72	0.660	21.5	0.014	1.98
1.48	0.75	1.11	0.712	23.2	0.016	1.48
1.53	1.00	1.53	0.740	23.9	0.017	1.15
1.57	1.25	1.96	0.759	24.4	0.018	0.94
1.62	1.50	2.43	0.772	24.7	0.019	0.78
1.66	1.75	2.19	0.794	25.3	0.020	0.69
1.70	2.00	3.40	1.135	35.9	0.040	1.19
1.74	2.25	3.92	1.254	39.1	0.049	1.25
1.79	2.50	4.48	1.353	42.2	0.057	1.27
1.82	2.75	5.01	1.466	45.8	0.067	1.34
1.86	3.00	5.58	1.578	48.8	0.077	1.38
1.90	3.25	6.18	1.698	52.8	0.089	1.45
1.94	3.50	6.79	1.781	55.0	0.097	1.55
1.97	3.75	7.39	1.892	58.3	0.110	1.49
2.02	4.00	8.08	1.985	61.7	0.122	1.51
2.05	4.25	8.71	2.100	64.5	0.135	1.55
2.09	4.50	9.41	2.205	67.9	0.149	1.59

APPENDIX B

```
#import necessary packages#
from imutils import contours
from skimage import measure
import numpy as np
import argparse
import imutils
import cv2
import json

#construct the argument parse and parse the arguments#
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True, help="path to
    the image file")
args = vars(ap.parse_args())

#load the image, convert it to grayscale and blur it#
image = cv2.imread(args["image"])
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blurred = cv2.GaussianBlur(gray, (17,17), 0)

#save and export image for debug #
cv2.imwrite("db_grey_blurred.jpeg", blurred)

#threshold the image to reveal light regions
in the blurred image#
thresh = cv2.threshold(blurred, 240, 255, cv2.THRESH_BINARY)[1]

#save and export image for debug #
cv2.imwrite("db_thresh1.jpeg", thresh)

#perform a series of erosions and dilations to remove any small
    blobs of noise from the threshold image#
thresh = cv2.erode(thresh, None, iterations=1)
#save and export image for debug #
cv2.imwrite("db_erode1.jpeg", thresh)

#create convex hulls of blobs#
k = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(30,30))
thresh = cv2.morphologyEx(thresh, cv2.MORPH_CLOSE, k)
#save and export image for debug #
```

```
cv2.imwrite("db_close.jpeg", thresh)

thresh = cv2.dilate(thresh, None, iterations=10)
#save and export image for debug #
cv2.imwrite("db_dilate1.jpeg", thresh)

#perform a connected component analysis on the threshold image,
    then initialize a mask to store only the "large" components#
labels = measure.label(thresh, neighbors=8, background=0)
mask = np.zeros(thresh.shape, dtype="uint8")

#loop over the unique components#
for label in np.unique(labels):
    #if this is a background label ignore it#
    if label == 0:
        continue
    #otherwise, construct the label mask and count the number
        of pixels#
    labelMask = np.zeros(thresh.shape, dtype="uint8")
    labelMask[labels == label] = 255
    numPixels = cv2.countNonZero(labelMask)

    #if the number of pixels in the component is sufficiently
        small then add it to our mask of blobs#
    if numPixels < 2100:
        mask = cv2.add(mask, labelMask)

#find the contours in the mask, then sort them
    from left to right#
cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL,
    cv2.CHAIN_APPROX_SIMPLE)
cnts = cnts[0] if imutils.is_cv2() else cnts[1]
cnts = contours.sort_contours(cnts)[0]

a = []

#loop over the contours#
for (i, c) in enumerate(cnts):
    #draw the bright spot on the image#
    (x, y, w, h) = cv2.boundingRect(c)
    #code for saving x-y coordinates#
    a.append(x)
    a.append(y)
    ((cX, cY), radius) = cv2.minEnclosingCircle(c)
    cv2.circle(image, (int(cX), int(cY)), int(radius),
        (0, 0, 255), 3)
    cv2.putText(image, "No{}".format(i+1), (x, y-15),
        cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)

#output centers as [x1, y1, x2, y2,...]#
with open('centers.txt', 'w') as f:
    f.write(json.dumps(a))

#display centers#
print(a)

#save and export image file#
cv2.imwrite("output.jpeg", image)

#scale image to fit on screen#
image = cv2.resize(image, None, fx=0.5, fy=0.5)

#show the output image#
cv2.imshow("Image", image)
cv2.waitKey(0)
```

APPENDIX C

```
import numpy as np
import cv2
import cv2.aruco as aruco

cap = cv2.VideoCapture(0)

# FILE_STORAGE_READ#
cv_file = cv2.FileStorage("calib_images/test.yaml",
    cv2.FILE_STORAGE_READ)

# Note : we also have to specify the type to retrieve otherwise
    we only get a#
# FileNode object back instead of a matrix#
mtx = cv_file.getNode("camera_matrix").mat()
dist = cv_file.getNode("dist_coeff").mat()

print("camera_matrix : ", mtx.tolist())
print("dist_matrix : ", dist.tolist())

cv_file.release()

aruco_dict = aruco.Dictionary_get(aruco.DICT_4X4_50)
parameters = aruco.DetectorParameters_create()

#PARAMETERS FOR BETTER ARUCO DETECTION#
#default = 0.05 - increase for higher distorted images#
parameters.polygonalApproxAccuracyRate = 0.1

#default = 0.03 - min marker size relative to image size#
parameters.minMarkerPerimeterRate = 0.001

#default = 4 - increase to consider more contours in image#
parameters.maxMarkerPerimeterRate = 4

#default = 5 - Std Dev of pixels when Thresholding#
parameters.minOtsuStdDev = 4

#default = 4 - increase to improve bit extraction process#
parameters.perspectiveRemovePixelPerCell = 6

#default = 0.13 - increase to reduce the number of
```

```
    considered pixels#
parameters.perspectiveRemoveIgnoredMarginPerCell = 0.4

#default = 0.35 - increase to correct more pixels in
    border area#
parameters.maxErroneousBitsInBorderRate = 0.4

#default = 0.6 - increase the number of corrected bits in cell#
parameters.errorCorrectionRate = 0.6


while True:
    ret, frame = cap.read()
    # operations on the frame come here#
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # lists of ids and the corners belonging to each id#
    corners, ids, rejectedImgPoints = aruco.detectMarkers(gray,
        aruco_dict, parameters=parameters)

    # font for displaying text (below)#
    font = cv2.FONT_HERSHEY_SIMPLEX

    if np.all(ids is not None):
        for i, c in zip(ids, corners):
            print(i, c)
            print()
            #Estimate pose of each marker and return the values#
            #rvet and tvec---different from camera coefficients#
            rvec, tvec, _ = aruco.estimatePoseSingleMarkers(c,
                0.01, mtx, dist)
            aruco.drawAxis(frame, mtx, dist, rvec[0], tvec[0],
                0.01)  #Draw Axis#

        #Draw A square around the markers#
        aruco.drawDetectedMarkers(frame, corners)

        cv2.putText(frame, "Id: " + str(ids), (0, 64), font, 1,
            (0, 255, 0), 2, cv2.LINE_AA)

        # Display the resulting frame#
    cv2.imshow(’frame’, frame)
    if cv2.waitKey(1) & 0xFF == ord(’q’):
        break

```

```python
# When everything done, release the capture#
cap.release()
cv2.destroyAllWindows()
```