
# USING A PLENOPTIC CAMERA FOR REAL-TIME DEPTH ESTIMATION

THESIS

Ryan J. Anderson, 1st Lieutenant, USAF

AFIT-ENY-MS-17-M-002

**DEPARTMENT OF THE AIR FORCE**
**AIR UNIVERSITY**

**AIR FORCE INSTITUTE OF TECHNOLOGY**

**Wright-Patterson Air Force Base, Ohio**

AFIT-ENY-MS-17-M-002

# USING A PLENOPTIC CAMERA FOR REAL-TIME DEPTH ESTIMATION

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Astronautical Engineering

Ryan J. Anderson, BS

1st Lieutenant, USAF

December 2016

AFIT-ENY-MS-17-M-002

# USING A PLENOPTIC CAMERA FOR REAL-TIME DEPTH ESTIMATION

Ryan J. Anderson, BS

1st Lieutenant, USAF

Committee Membership:

Dr. Richard Cobb
Chair

Dr. Michael Hawks
Member

Dr. Matthew Sambora
Member

AFIT-ENY-MS-17-M-002


## Abstract

The plenoptic camera collects samples of the 4D light field, which allows for the collection of imagery and depth information simultaneously. The plenoptic camera differs from stereoscopic systems because the light field is captured by a single lens and sensor rather than two or more. This translates to less size, weight, and power (SWAP) which is ideal for space missions where imagery and depth information is needed such as proximity operations and docking missions. The main objective of this research is to design and evaluate performance of a method to autonomously output the depth of key elements of a scene in real-time. In this research, the depth is estimated using a gradient method and the key elements of the scene are selected using the Hough transform. A major finding of this research is that in order for this to run near real-time, only a small portion of the light field can be analyzed due to size of the data set. This results in the potential to miss important information that the light field has to offer. The average error of the Lytro Illum was ~7% while the First Generation's was ~17% error with a decrease in accuracy as the range increases. The average run time for the Illum and First Generation was approximately five seconds and three seconds respectively using the Hough transform to reduce the size of the light fields. The Hough transform is the most significant portion of the run time, but it still reduced the run time by than it increased it. This work lays the groundwork for using a plenoptic camera to autonomously output the depth information about a scene in real-time by developing a depth estimation method for specific features in light fields and concluding that the Hough transform is a good method for this, especially if multiple features are desired.

## Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Dr. Richard Cobb, for his guidance and support throughout the course of this thesis effort. The insight and experience was very much appreciated. I would, also, like to thank my wife for her support and encouragement throughout the course of this thesis effort.

Ryan J. Anderson

## Table of Contents

## List of Figures

## List of Tables

# USING A PLENOPTIC CAMERA FOR REAL-TIME DEPTH ESTIMATION

## I. Introduction

### 1.1 Background

The ability to rendezvous one satellite with another satellite offers the Air Force with a vast set of capabilities in space. Performing maneuvers near another satellite, known as proximity operations or prox-ops, allows for the collection of data and for interaction with the target satellite in many ways that are not possible with ground-based technology after the satellite has been launched. Numerous mission concepts, such as satellite inspection, maintenance and refueling, and debris removal, are dependent on proximity operations technologies. The ability to accomplish these operations unmanned and autonomously further improves mission capabilities. At a minimum, a satellite performing proximity operations with an object in space, especially an unknown or uncooperative one, must be able to estimate its own relative position with respect to the target and estimate the target's orbit in order to effectively perform proximity operations. In other works, accurate depth information of the target is needed for successful operations. Additionally, knowledge of the target's orientation and dimensions may be required if docking or other physical interaction is part of the mission. In the early years of space exploration, spacecraft rendezvous were performed between two manned spacecraft under the command of an astronaut with very little to no automation [1].

Recently, missions from the National Aeronautics and Space Administration (NASA), the Air Force Research Laboratory (AFRL), the Defense Advanced Research Projects Agency (DARPA), SpaceX, and Orbital Sciences Corporation aim to improve proximity operations technology and demonstrate the capability to rendezvous with numerous spacecraft that use different levels of autonomous operation and sensor data of the target satellite. AFRL's Automated Rendezvous System used an active scanning time-of-flight LiDAR sensor on board XSS-11 (eXperimental Satellite System

11), which used a known model of the spacecraft to estimate the relative positions of the two spacecraft [2]. AFRL more recently launched the Automated Navigation and Guidance Experiment for Local Space (ANGELS) satellite which "hosts an SSA sensor payload to evaluate techniques for detecting, tracking, and characterizing of space objects, as well as, attribution of actions in space" [3]. In a separate DARPA experiment on-board their Orbital Express and Demonstration for Autonomous Rendezvous Technology (DART) spacecraft, DARPA used a system they developed called the Advanced Video Guidance Sensor (AVGS), which illuminates retro-reflectors at a known orientation on a target with a laser to determine relative position [4].

## 1.2 Research Focus

### 1.2.1 Motivation to Use Plenoptic Based Computer Vision for Proximity Operations.

The proximity operations methods listed in Section 1.1 have limitations however. A LiDAR sensor that can be used for range finding requires an additional subsystem with additional hardware, which means more size, weight, and power required. Size, weight, and power each come at a very expensive premium in space. The AVGS system alone weighs approximately 8 kilograms, takes up more than 10 liters of volume, and requires 30 Watts of power during tracking mode [4]. Additionally, the Rendezvous LiDAR Sensor (RLS) and methods based on fiducial markings both require intimate knowledge about the target in order to be effective. Stereoscopic or camera array systems require very precise calibration for relative location between cameras for accurate estimation, which could be compromised during launch and on orbit from effects such as body flex or thermal expansion. This can result in inaccuracies in the data collected by these systems that make it significantly less valuable. Monocular systems require a satellite to take multiple images of a target while keeping track of its own location and the target's or assuming that the target is stationary. These methods may provide less uncertainty for views of cooperative known targets, but severely limit the capabilities of the proximity operations satellite if the mission requires approaching an unknown or uncooperative object.

A plenoptic, or light field, camera offers a solution to many of these problems. Plenoptic cameras offer a passive form of depth estimation and requires significantly less SWAP than active systems such as LiDAR. While plenoptic cameras do require calibration, the more rigid structure of the micolens array and main lens, as opposed to multiple cameras, means that it is less likely to be compromised during launch or on orbit from effects such as body flex of thermal expansion. Also, only a single image is required to gather range information, so there is no need to keep track of its own position, the position of the target or make assumptions about the target's movement. Section 2.1.2 provides the theory of light field cameras but a brief overview is provided here. Light field cameras use a single camera to capture an image set called a light field, which a collection of subimages that each show a slightly different perspective of a scene. An example light field is shown in Figure 1. [5].

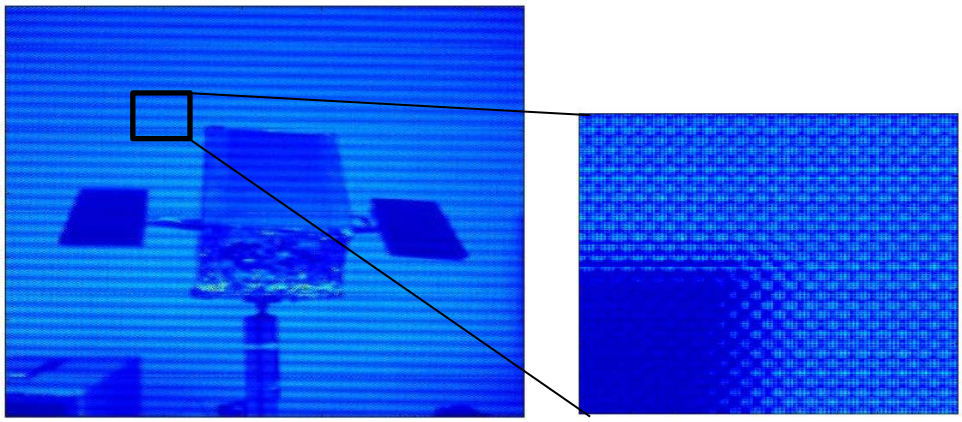


**Figure 1: Example light field (left) and zoomed in to show lenslets (right)**

It does this by placing an array of micro lenses between the main lens and the sensor to separate the light into micro images. Each of these micro images has a slightly different view and focus, and the light field as a whole holds a great deal of information, including depth information, that can be extracted in post processing.

Designing a plenoptic image-based computer vision (CV) method to perform the required tasks that only uses a single camera and a CPU with very specialized software would significantly minimize the additional subsystem requirements on the spacecraft. Most spacecraft already have star trackers, which use dedicated CPUs to perform stellar rectification with a catalog for attitude determination. Suddenly, with a dual purpose image processing subsystem that performs attitude determination and CV-based proximity operations navigation, the required SWAP to perform a proximity operations mission may be much smaller than previously imagined.

Once within range to acquire resolved imagery, it may also be possible to construct a model of the target spacecraft using plenoptic image processing and multiple view geometry methods [6]. This type of method removes the requirement for *a priori* knowledge of the shape and structure of the target spacecraft in question. This method further expands the potential mission areas into operations near uncooperative targets, such as sequential rendezvous for removal of space debris, or operations with a damaged or unresponsive spacecraft for information gathering or repair purposes.

### 1.2.2 Proximity Operations with a CubeSat.

CubeSats are becoming more popular for many organizations as alternatives to much larger spacecraft. CubeSats allow for flexible launch options with a smaller and lighter platform, which makes CubeSats a viable and affordable space platform for universities. Additionally, advances in technology are allowing CubeSat designers to fit an incredible amount of mission capability into very small volumes.

One popular mission concept for proximity operations satellites is the inspector satellite. An inspector satellite can gather valuable data from space about its target satellite, potentially providing the right information to extend the target satellite's time on-orbit and decrease its cost per year by determining if it needs to be repaired. To make the inspector satellite a practical idea, it needs to be cheaper than replacing the original satellite a few years early. The flexible and affordable launch options that CubeSats provide offer a significant motivation behind implementing a plenoptic image-based CV proximity operations method on a CubeSat when developing satellite inspector missions.

Plenoptic imaging systems are ideally suited for prox ops because they allow for spatial information to be extracted from a single image through image processing. This spatial information includes depth and position information that can be used to create a 3D model of the scene. However, accuracy of the spatial information decreases as the distance from the target increases relative to the camera size. As the distance from the target continues to increase, eventually no additional information is provided, so the relatively close distances of prox ops are ideal [7].

### 1.3 Goals

As discussed in Section 1.2.2, CubeSats are becoming increasingly adopted for space operations but there is a shortage of hardware systems that are suitable for flight in some areas such as proximity operations hardware. Therefore, the goal of this research is to determine if light field cameras are suitable for use on spacecraft as real-time ranging devices for proximity operations and docking operations by answering the following investigative questions:

- What is the availability of plenoptic camera (light field) commercial hardware?
- What accuracy of range estimation (depth) is attainable from light field cameras?
- What is the speed of computation of range estimation, and is it suitable for real-time range estimation?
- Is custom processing needed, and if so, in what specific areas?

### 1.4 Research approach

This research was conducted using a combination of testing on both synthetic and lab images, processed using commercial software and custom routines. This thesis analyzes the processing time required and the resulting performance of each and provides comparisons.

### 1.5 Assumptions/Limitations

The research was conducted in this thesis under the following assumptions and limitations. The hardware used in this research was limited to the two currently available Lytro light field cameras, which

will be describe in greater detail in Section 2.2. These are assumed to have comparable results to the other commercially available light field cameras for the purpose of this research. All experiments conducted during this research were limited to laboratory conditions. The results collected from these experiments were assumed to be representative of real-world scenarios. The effects of exoatmoshperic lighting conditions were not considered.

Light fields come with a multitude of different file types between the different commercially available light field cameras and light field data sets available online, so for the course of this research, it was assumed that all source images used for the method developed were already in a format that was readable to Matlab because in real applications, all images used for a system will be in a single format and the system will be designed to that format. Although the implementation of a light field camera on a satellite for proximity operations is the end goal, in this phase of the research relative motion has not been and does not need to be accounted for at this point in the research. Relative motion does not need to be taken into account when using light field cameras because range is extracted from a single image as will be discussed in chapters II and III, and a discussion of why it may be useful to take into account in the future work section of chapter V. The speed of calculation of this method is limited by the performance of the computer hardware used in this research and may not be representative of hardware that is suitable for use in space. It is assumed that hardware with similar specifications will result in similar results in future testing.

### 1.6 Summary

This thesis is divided into five chapters. This chapter provided a background on this research, the research focus and motivation behind the research, the goals, the approach used, and the assumptions and limitation that went into this research. Chapter II provides a literature review that examines information that is relevant to this thesis to include: fundamentals of optics, plenoptic theory, the Hough transform, current hardware available, software packages, and related research in non plenoptic and plenoptic depth estimation. Chapter III describes the hardware used, the method of depth estimation, and the experimental

setup. Chapter IV presents the results of the analysis of the accuracy of depth estimation and the speed of computation. Chapter V summarizes the conclusions and recommendations that resulted from the experiment and discusses recommendations for future applications and research.

# II. Literature Review

## Chapter Overview

This chapter provides a summary of the fundamentals of optics, the relevant theory used in the design of the developed light field depth estimation method, current hardware available for light field cameras, software used, and related research. First, Section 2.1 goes into the fundamentals of optics, plenoptic theory, and the Hough transform. Next, Section 2.2 goes into the current light field camera hardware available and the specifications of that hardware. Then, Section 2.3 gives an overview of the software used for this research. Lastly, Section 2.4 discusses related research in depth estimation.

## 2.1 Theory

### 2.1.1 The Pinhole Camera Model

The geometric relationship between a 3D scene and a 2D image is defined by the pinhole camera model. The pinhole camera is an ideal camera with no lens and an aperture that is characterized as a single point. This is shown in Figure 2, where the point C, is the camera aperture and light from the 3D scene is projected onto the image plane, the $(x, y)$ plane, as a 2D image that is a distance $f$, the camera's focal length, from the camera center. Each point visible to the image plane from the camera's center is mapped onto the image plane where the line that originates at the point and travels through the camera center intersects with the image plane. Figure 2 uses similar triangles to show that the 3D point at $(X, Y, Z)$ is mapped as a 2D image to the image plane at the coordinates $(\frac{fX}{Z}, \frac{fY}{Z})$ [6]. From this basis, the theory of the plenoptic function and plenoptic cameras will be expanded upon.

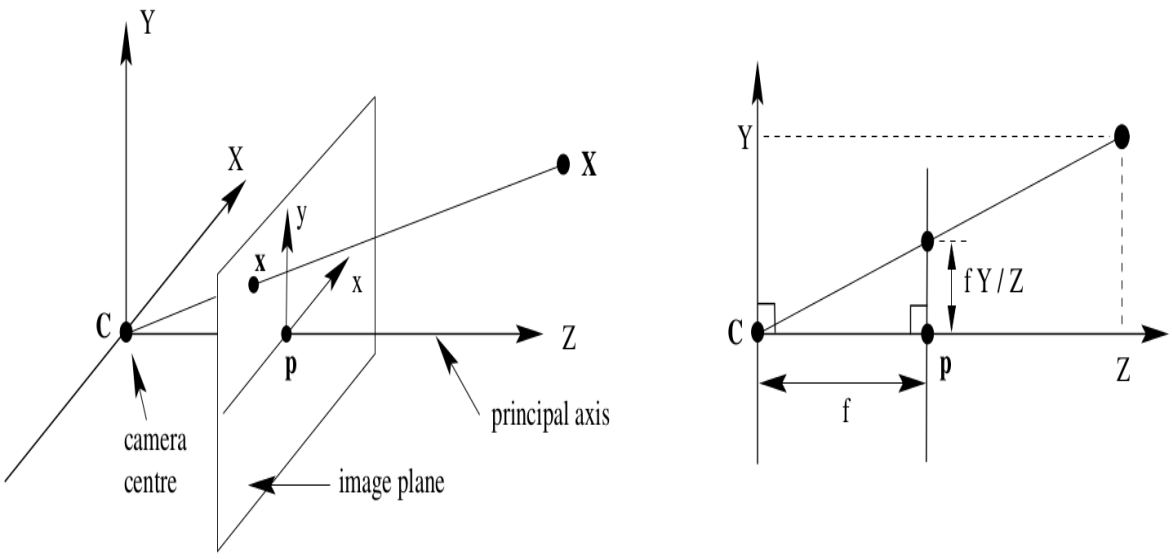



**Figure 2: The pinhole camera model [6]**

2.1.2 Plenoptic Theory

The plenoptic function describes the information about the world that is contained in the light in a region of space. The plenoptic function consists of 7 dimensions that are as follows: time $(t)$, viewing position $(x, y)$, wavelength $(\lambda)$, and distribution across lens $(V_x, V_y, V_z)$, $P(x, y, \lambda, t, V_x, V_y, V_Z)$. This describes a color holographic movie that would allow the "reconstruction of every possible view, at every moment, from every position, at every wavelength, within the bounds of the space-time-wavelength region under consideration" [5]. In a limited plenoptic function $P(x, y, \lambda, V_x, V_Y)$, which describes the Lytro plenoptic camera, where the images are observed in a distribution across the lens, wavelength, and viewing position in two dimensions. While this does not allow for a full reconstruction[1] of the space-time-wavelength region under consideration, it does allow for more reconstruction than a conventional camera.

When Adelson and Wang first designed the plenoptic camera, they observed that its main feature was its ability to achieve stereo ranging using a single image. Figure 3 shows a plenoptic camera where

[1] Reconstruction refers to the process of recreating the 3D properties of an object from a 2D images

the sensor array is covered with an array of tiny pinhole cameras [8]. In 3(a), the object is in focus and in 3(b) and 3(c), the object is near and far respectively, resulting in the image being displaced left or right depending on the object's range. If the pinhole array is replaced by a lenticular array, the light-gathering efficiency of the lens can be improved and aliasing artifacts can be reduced. Figure 4 shows a more complete plenoptic optical system where a field lens places the main lens at optical infinity from the lenticular array so that the image in the center of each micro image originates from the center of the main lens [8].

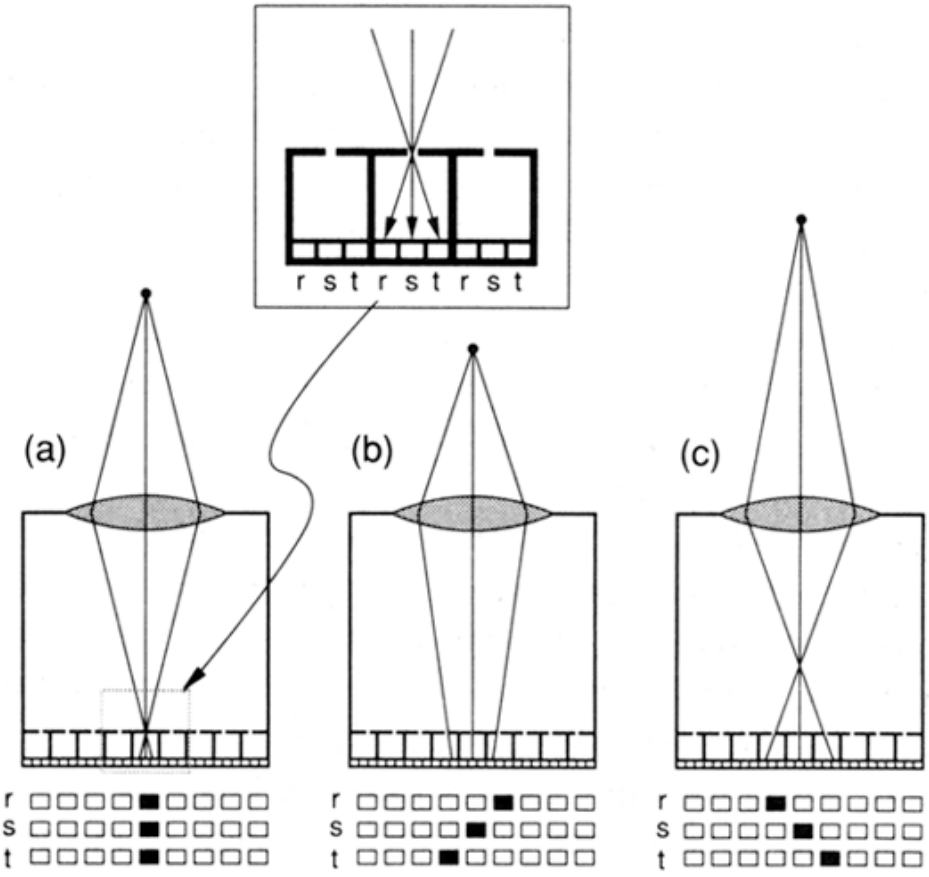



**Figure 3: Array of pinhole cameras placed over sensor array [8]**

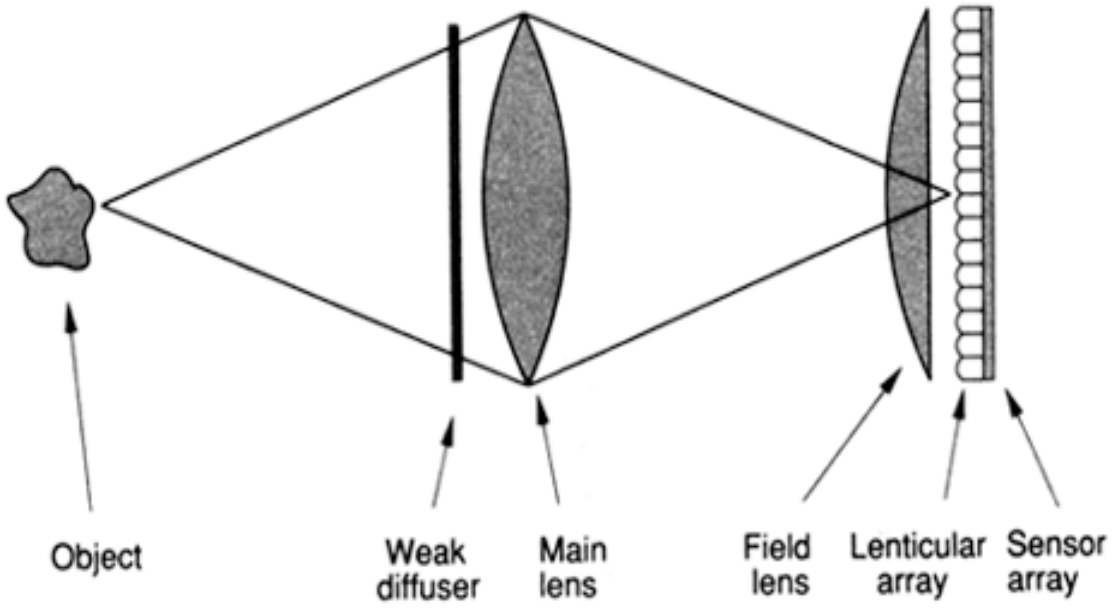



**Figure 4: Plenoptic camera with lenticular array [11]**

As already mentioned in Chapter I, the set of data that comes from a plenoptic image is a known as a 4D light field or light field [8]. It is based on a geometric optic formulation of light describing its propagation through space and is basically the distribution of radiance in the 2D plane.

By setting some of the variables contained in the light field as constant, we can gain some information about the scene. The coordinates $u$ and $v$ are used to specify the location of the subaperture, while the coordinates $s$ and $t$ specify the pixel location within each subaperture. From this, if $u$ and $v$ are fixed at some value, we expect to get an image that is similar to an image taken with a conventional camera with comparable specifications. Ng calls this 2D light field slice a 'subaperture image'. Figure 5 shows that this subaperture image will produce an image with less blur. This is due to the higher f/#, a consequence of the smaller aperture size of the lenslet in the array that the subaperture image is associated with, which results in greater depth of field and more of the image being in focus. Since more of the image is in focus, the final image will have less blur over a greater depth. This means that a light field camera can be used for a wider range of image depths than a conventional camera at the same focus. The top image shows an image from a conventional camera of two point sources that are not in focus while

the bottom image shows a subaperture image that was obtained by fixing $u$ and $v$ in the light field shown in the middle image [9].

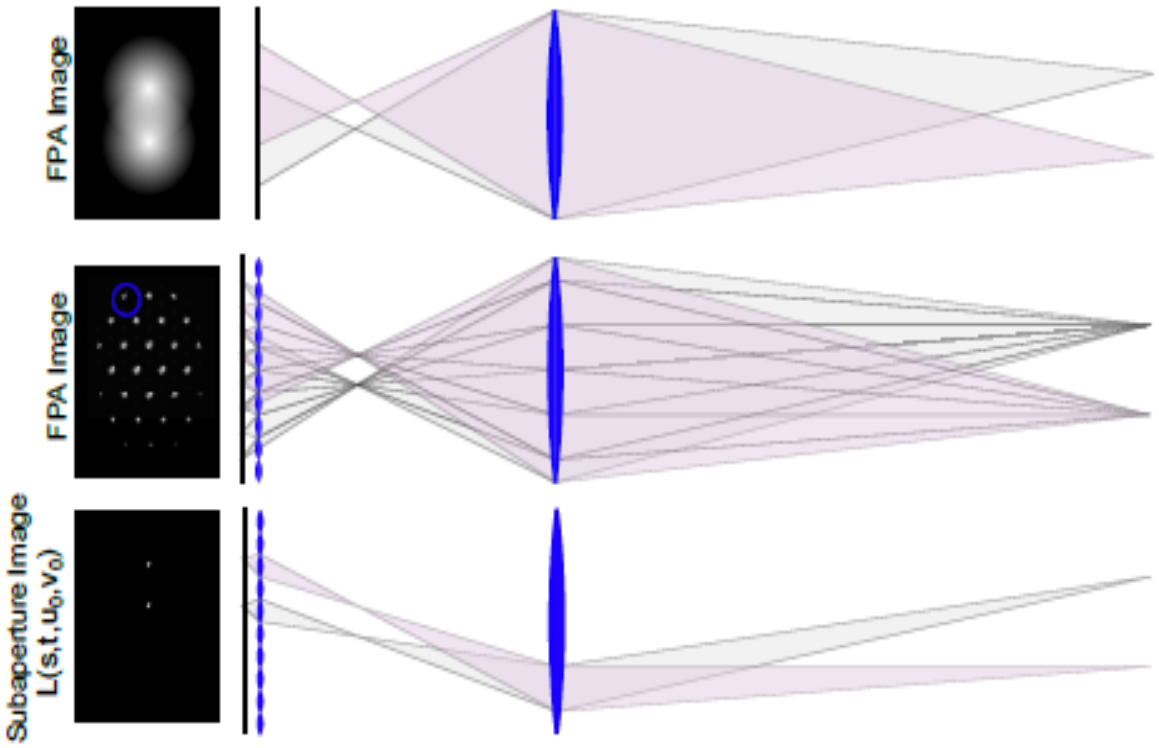



**Figure 5: Subaperture image formulation [7]**

By fixing only one coordinate, the 3D subspace of the light field is obtained, as shown in Figure 6 where the $v$ coordinate is fixed. The representation of the subspace is created by stacking all the subaperture images that have the same value of $v$ on top of each other to create an image cube. As expected from what has been shown previously, the top surface of the cube is similar to an image taken by a conventional camera. The front face of the structure, shown in Figure 6(b), is a 2D plane that is formed by fixing another coordinate, $t$, in addition to $v$, and this subspace is known as an epipolar plane image or EPI [10]. The sloped lines visible in the resulting image are a consequence of the difference in the apparent location of objects when viewed through different apertures of the camera, or as $u$ varies,

under the parallax effect. The slope of each of these lines, *m*, is given by Equation 1 and is related to the distance from the camera to the point that this line represents

$$m = \frac{\frac{ds}{\Delta s}}{\frac{du}{\Delta u}} \quad [1]$$

where $\frac{ds}{\Delta s}$ is the change in the *s* direction per pixel and $\frac{du}{\Delta u}$ is the change in the *u* direction per pixel.

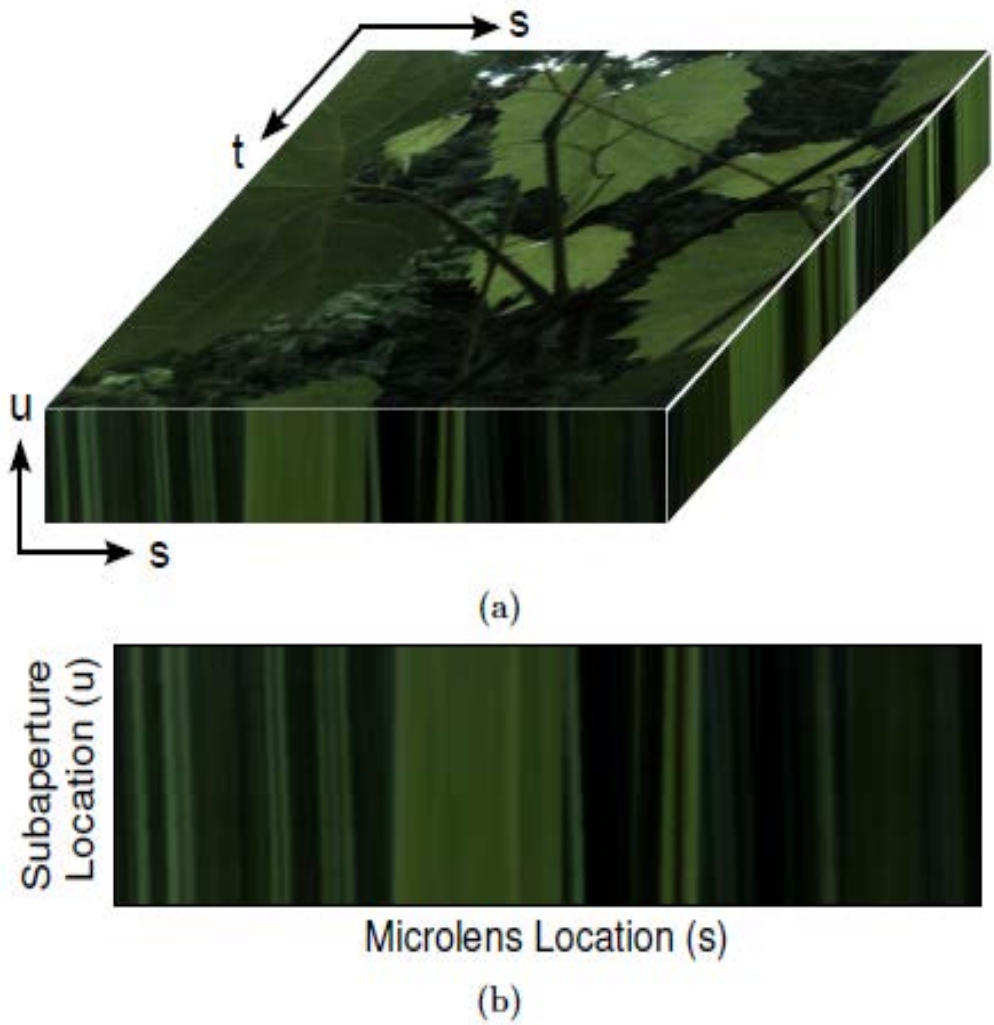



**Figure 6: (a) 3D subspace of a light field, and (b) an epipolar plane image (EPI) of the subspace [14]**

The generation of an EPI representation from a sampled light field is simpler compared to camera arrays, where the projective transformations of the views of the individual cameras have to be rectified and unified into one epipolar coordinate system requiring a precise calibration of all cameras. Due to the optical properties of the micro lenses, with the image plane of the main lens defining the epipolar coordinate system, these projective transformations are reduced to simple translations, which are given by an offset. Hence, one simply has to rearrange the viewpoint-dependent rendered views into the EPI representation. However, the necessarily small depth of field of the micro lenses causes other problems. For most algorithms, the EPI structure can only be effectively evaluated in areas with high-frequency textures, which is only possible for parts of a scene which are in focus. Additionally, plenoptic cameras suffer from imaging artifacts in out-of-focus areas. Hence, in order to generate EPIs which can be used to analyze the entire scene at once, we have to generate the EPIs from "all-in-focus" (i.e. full depth of field) views [11].

Adelson and Wang developed an algorithm to determine from the parallax shift resulting from viewing an object through different subapertures of the main lens [8]. This shift manifests itself as a sloping of lines when neighboring subapertures are stacked on top of each other, shown in Figure 6 (b) [7]. As a result of this occurrence, they included the gradient of the image into their algorithm in order to estimate the slope. Each of these lines maps to a point on the image and its slope is related to the point's range. One method to estimate the slope is to shear the light field by different degrees of shearing and search for vertical lines that have low variance at each of the degrees of shearing. This results in a matrix of $N_s$ by $N_m$ variance values, where $N_s$ is the width of the slice and $N_m$ is the number of slopes used for shearing the EPI [7]. Another method to estimate depth is via refocusing. The ability to produce refocused imagery is one of the most striking capabilities latent in the light field captured by the plenoptic camera. When properly focused on an object within a scene, an image will be characterized by sharp edges, steep gradients, and comparatively large amounts of energy in high spatial frequencies. Thus, a natural approach to range finding is to search for refocused images containing these characteristics [11].

In the spatial domain, this means producing a stack of refocused images and determining in which frame the image gradient reaches a maximum at each pixel. Indeed, refocusing involves the same shearing operation used there to identify the slope in an EPI. The depth-through-refocusing technique first performs this summation, and then looks for the high spatial gradients that are made possible when accurate refocusing minimizes the effective point spread function[2] (PSF) of an object point. When the photo-consistency (variance) is low, which results when the samples associated with a single object point are gathered together under a single image pixel, or in focus, rather than spread across neighboring pixels, or out of focus, where they would reduce spatial image gradients. In order to consider the effects of defocus on imagery, it is necessary to know the defocus-induced point spread function. Figure 7 demonstrates the formation of the PSF for a two dimensional slice of the light field. A defocused point is represented by a sloped line within the light field, and the image formation operation projects the line down into one dimension. The projection of the line is then equal to the point spread function for an object at the distance giving a line of that slope. The projection operation consists of counting up the number of *u* samples associated with each s sample. This same approach can be utilized for the case of the 2D PSF generated from the 4D light field, as illustrated in Figure 8. The light field slope dictates the range of (*u*, *v*) samples over which each (*s*, *t*) sample is spread [7].

[2] The point spread function is the impulse response of a focused optical system.

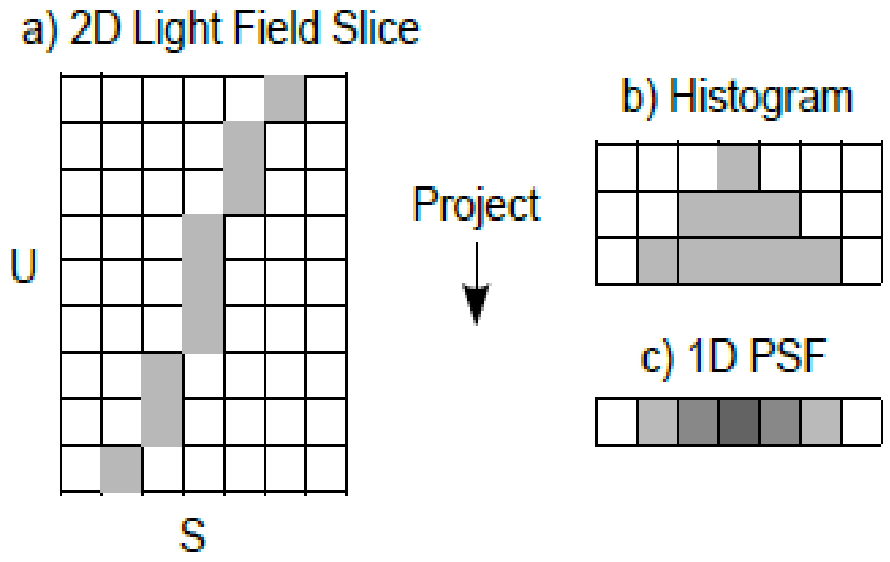



**Figure 7: 1D point spread function [9]**

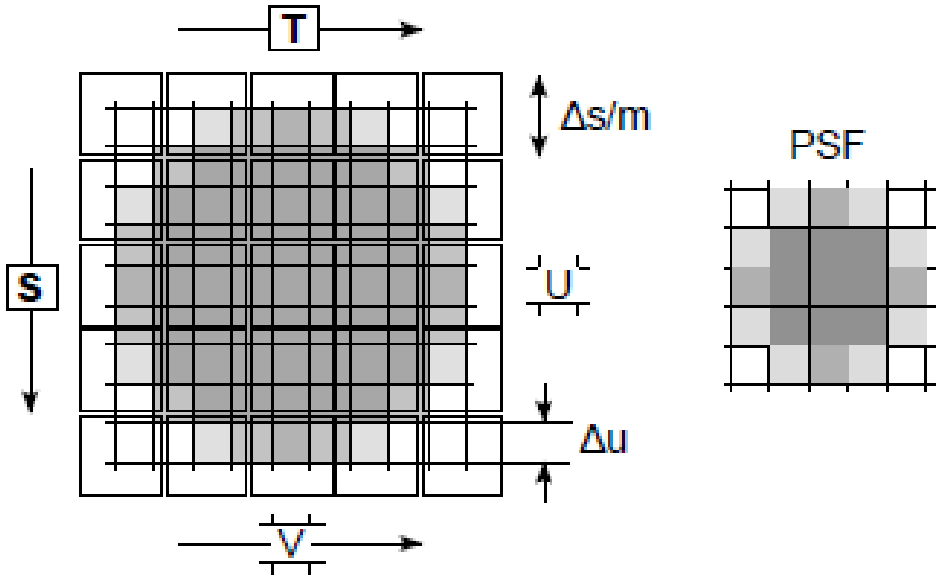



**Figure 8: 2D point spread function [9]**

2.1.3 The Hough Transform

The Hough transform, named after Paul Hough who patented the method in 1962, detects shapes in an image by finding accumulated points in a parameter space. A prominent shape will have many

values clustered around the parameter values that correspond to that particular shape. The easiest shape to detect is the line, which is what is used in this method. This is because satellites in general have a lot of edges and it is less computationally intensive, and therefore faster, to detect lines rather than to detect other shapes. The Hough transform uses point/line duality to detect lines. A line in the image can be defined by Equation 2 as

$$y = kx + c \quad [2]$$

where k is the slope of the line and $c$ is the value where the line crosses the $y$ axis. Rearranging equation 2, every line passing through a single point in the image can be defined by Equation 3

$$c = y - kx \quad [3]$$

This leads to point/line duality shown in Figure 9 where point P corresponds to line $p$ and line L corresponds to point $l$.

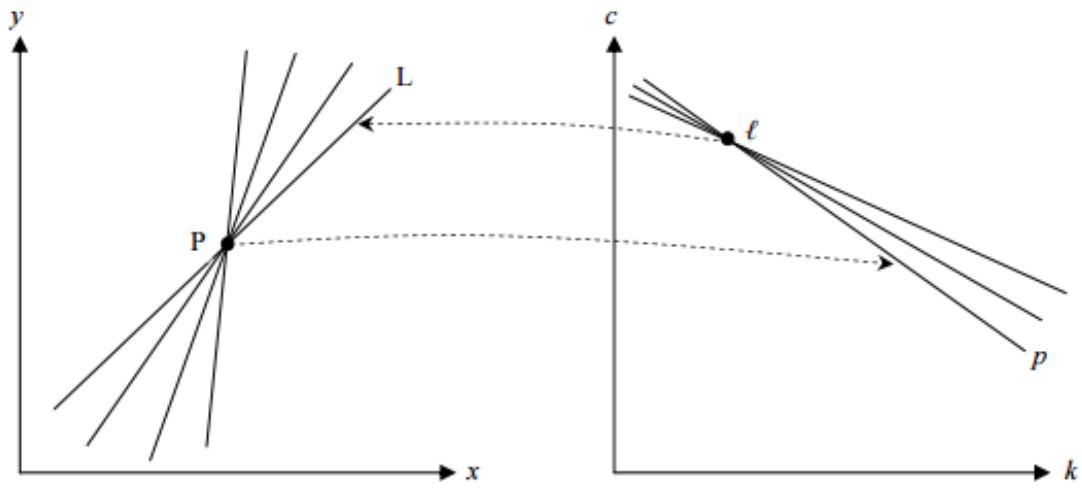



**Figure 9: Point/line duality [12]**

There is a singularity however, because when line L is vertical the value of $k$ would be infinity. To remove the singularity, the line is defined in polar coordinates shown by Equation 4,

$$\rho = xcos(\theta) + ysin(\theta) \quad [4]$$

where $\rho$ is the distance from a line going through the origin that is parallel to line L and $\theta$ is the angle of inclination to the normal. This results in point P corresponding to a curve, as shown in Figure 10, instead of a line [12].

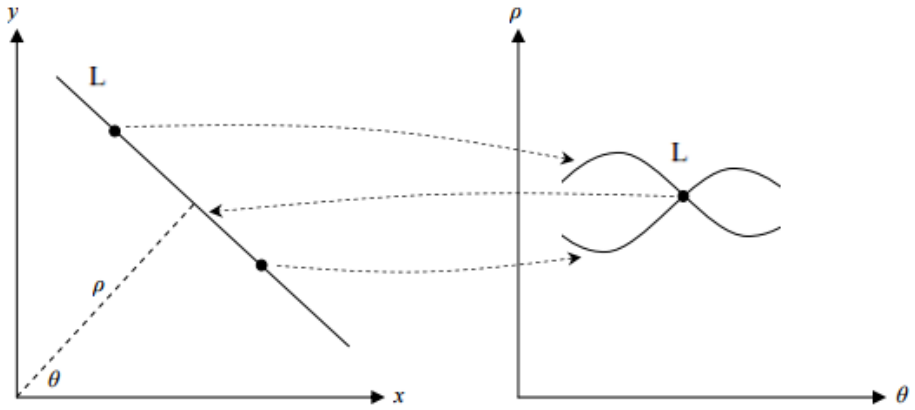



**Figure 10: Point/line duality in polar coordinates [12]**

When the curve of each point along line L is plotted on the rho/theta map, they all intersect at a single point that has become line L. The height, or number of curves that intersect, at this point corresponds to the length of the line, the higher the point is, the longer the line is. Figure 11 shows an image, the top image, and its corresponding rho/theta map, the bottom image. In this rho/theta map the height of each point is indicated by the brightness of each point. As shown, the brightest areas of the rho/theta map are near theta equals +- 90 degrees and you can see that this corresponds to the large number of horizontal lines in the image that are some of the dominant linear features [13].

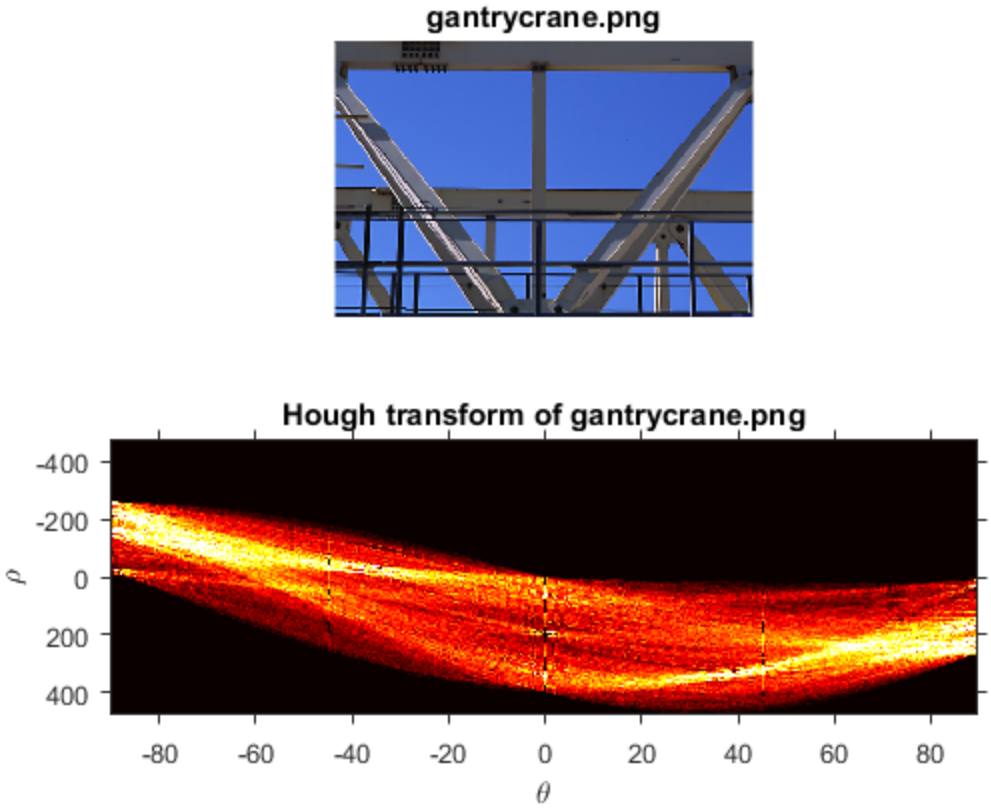



**Figure 11: Sample image with rho/theta map [13]**

In the context of this research, the Hough transform will be used to find the most dominant linear feature(s) in the central subaperture image of a light field. The light field is then cropped around the dominant feature(s) so that the size of the light field can be reduced, since the light field generated by the camera is too large to be quickly analyzed.

### 2.2 Hardware

Light field cameras are fairly new to the commercial market, first appearing in 2010 from Raytrix and shortly after Lytro. Currently these are the only 2 companies that manufacture light field cameras. Lytro caters to the consumer market, and Raytrix caters to the industrial and scientific market. Lytro currently offers 2 cameras, the First Generation and the Illum, pictured in Figure 12, and whose specifications are shown in Table 1. Since Lytro cameras cater to the consumer market, their main

application has been as a personal camera that can be refocused after the fact such as the examples shown in Figure 13. [14]

**Figure 12: The Lytro First Generation light field camera (left) and the Lytro Illum (right) [14]**

**Figure 13: Lytro example images [15]**

**Table 1: Lytro camera specifications [14]**

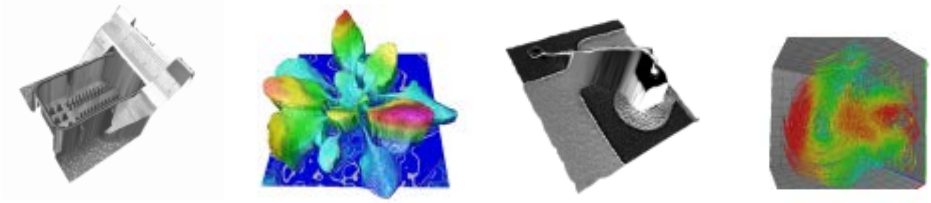
| Product Name: | First Generation | Illum |
|---|---|---|
| Megarays (Megapixels): | 11 | 40 |
| Sensor Type | CMOS | CMOS |
| Max Effective Resolution (MP) | ~1 | ~4 |
| Processed Image Size | ~125 MB<br>11 x 11 x 378 x 379 pixels | ~430 MB<br>15 x 15 x 434 x 626 pixels |
| Field of View (deg x deg) | ~45 x 45 | ~62 x 44 |

Raytrix offers 6 different cameras specialized in 4 different industrial and scientific applications: automated optical inspection (AOI), plant phenotyping, particle image velocimetry (PIV), and 3D

microscopy. Examples of the cameras and applications are shown in Figures 14 and 15 respectively. The technical specifications for these cameras are shown in Table 2 [16].


(a) (b) (c)

**Figure 14: (a) the R10 Series, (b) the R12 Series, (c) the R12 Micro Series [17]**



(a) (b) (c) (d)

**Figure 15: (a) Automated optical inspection, (b) plant phenotyping, (c) particle image velocimetry, (d) 3D microcopy [16]**

**Table 2: Raytrix camera specifications [17]**

| Product Name: | R42 | R8 | R10 | R12 | R29 | R5 |
|---|---|---|---|---|---|---|
| Megarays (Megapixels): | 42 | 8 | 10 | 12 | 29 | 4 |
| Sensor Type | CMOS | CMOS | CMOS | CMOS | CCD | CMOS |
| Pixel size ($\mu m$) | 1.12 | 2.24 | 2.24 | 5.5 | 5.5 | 5.5 |
| Max Effective Resolution (MP) | 10 | 2 | 2.5 | 3 | 7 | 1 |
| Microscope variation | No | No | Yes | Yes | No | No |
| Spectrum | Color | Color | Mono | Color, Mono | Color, Mono, NIR | Mono |

Lytro was chosen due to the commercial availably and price, which is substantially less than Raytrix. At this point in the research, all that was needed was to see if light field cameras would be feasible to use. As a result, Lytro cameras were sufficient even though they are not made for scientific use. Higher fidelity cameras, such as Raytrix, may be utilized in future research and applications.

## 2.3 Software packages

The methods used in this thesis were written in MATLAB R2015B due to the ease and speed of quickly testing new ideas. The following toolboxes were also used: Image Processing Toolbox, Optimization Toolbox, DSP System Toolbox, Signal Processing Toolbox, and Computer Vision System Toolbox. Also used Matlab Light Field Toolbox v0.4 for working with light field imagery created by Donald Dansereau [18]. Currently there is no standard file type for light fields as Raytrix and Lytro have their own formats for outputting light fields from their cameras, with different cameras having different output file types in some cases, and online archives have for an even greater variety of file types. This can make them difficult to work with from a processing and comparison standpoint, therefore before analysis, all light fields were converted into a .mat format for input into Matlab. This creates a basis from which all of the light fields can be analyzed and compared to each other. The Lytro files were converted from the standard Lytro format, .lfp, to a five dimensional matrix in the .mat format using Dansereau's Light Field Toolbox v0.4, while the synthetic light field was already in a .mat format when downloaded and only needed to have the order of the dimensions rearranged for consistency.

## 2.4 Related Research

### 2.4.1 Non-plenoptic range estimation

Range estimation is achievable in a variety of ways using different hardware and methods. Some use active systems that bombard a target with some sort of energy and gather information based on how that energy returns while some use passive systems that gather

information based on energy from other sources that is reflected off of or produced by a target. There are good and bad consequences from both that will be discussed.

Light detection and ranging (Lidar) is an active system that is frequently used to gather range information about a target. At its most basic form, LIDAR ranging transmits a pulse of light at a target and detects how long it takes to reflect of the target and return to the sensor, as given in Equation 5,

$$range = \frac{c * t}{2} \quad [5]$$

where c is the speed of light and t is the time between the transmission of the pulse of light and when the reflected light is detected by the receiver [19]. Using this method, LIDAR can produce very accurate range estimates, dependent mainly on the timing system; if the timing system is accurate, the range estimate is accurate and if the timing system is inaccurate, the range estimate is inaccurate. There are some drawbacks to LIDAR systems though, they require multiple components, such as some kind of light emitter and at least one detector, and because they are emitting energy, the range that they are effective over is proportional to the power emitted. This means that to be effective in the space environment, where distances are generally measured in km or hundreds of km, a LIDAR system would require either a large amount of power, which takes up a lot of space in a spacecraft, or a very sensitive detector, which creates its own set of problems. These reasons mean that employing a LIDAR system on a CubeSat sized spacecraft would not be an effective method for proximity operations as the system would likely be too small to be powerful enough to gather any useful information. Radar uses a similar concept as LIDAR but instead of light pulses, radio frequencies are used, and also has many of the same size and power requirements that LIDAR has and will not be discussed further [19].

Passive systems currently come in two main forms, multi-camera systems that employ two or more cameras, and monocular systems that employ single cameras. Multi-camera systems generally use two or more cameras of known relative position and a phenomenon known as parallax to determine depth information. Parallax is the apparent shift in the location of an object when viewed from different lines of sight. This shift is relative to the objects depth and the relative positions of the cameras. One of the problems that systems with only two cameras have that does not affect plenoptic cameras is that parallax can only be determined in one direction, horizontal or vertical. With more cameras, calibration becomes more problematic and there is a greater chance of more errors that are less likely with plenoptic cameras due to the more rigid structure of the main lens and lenslet than that of a camera array. Monocular methods generally involve a single camera taking multiple pictures of a target. In this case either the camera is stationary and the target is moving, or the camera is moving and the target is stationary, or both are moving, and the camera must capture multiple views of the target [20]. One monocular method is structure from motion in which a camera takes multiple pictures of a target from different views with its own location known. Then the change in the features of the target from each view are used to recreate the 3D structure of the scene [21]. One drawback of this method is that the location of the camera needs to be known in order to recreate the 3D structure, which is not the case for plenoptic cameras. Another monocular method is simultaneous localization and mapping (SLAM) in which multiple views of a target are used to create a map of the environment while computing the cameras' location at the same time. Since there are multiple images taken of the target at different times, it is usually assumed that the environment is relatively static, such as a stationary target or camera, which does not need to be assumed in a plenoptic camera because a light field is from a single instance in time [20].

### 2.4.2 Plenoptic range estimation

Northrop Grumman is currently researching the acquisition of passive 3D imagery of Low Earth Orbit (LEO) objects from a ground-based single-aperture light field camera system. Their method of range calculation involves using image refocusing and change detection. Their camera system consists of a telescope and light field camera for image acquisition and a computer for image processing with technical specifications shown below [22].

- Telescope: 14” Meade LX600 advanced coma free catadioptric telescope
    - f/8 (2845 mm focal length) at prime focus
    - Effective f/5 with an f/6.3 focal reducer
    - Effective f/24 with a 15mm f.l. eyepiece
    - Effective f/15 with 15mm f.l. eyepiece and f/6.3 focal reducer
- Light field camera
    - Raytrix R5 C2GigE camera
    - Frame rate: up to 30 frames per second
    - Depth resolution: up to 100 discrete layers
    - Microlens f/#: f/2.4
- Image acquisition and processing
    - Windows computer with nVidia GPU; light field images acquired to disk
    - User can configure for automated camera triggering
    - Raytrix software and Northrop Grumman custom software developed for processing long-range 3D images
        - Image refocusing
        - Depth map/range finding
        - Optional processing modules (change detection; 3D reconstruction)

They have conducted testing on the ground at a range of 1.39 km with two targets and 0.7 km with one target shown in Figures 16, 17, and 18 below.


**Figure 16: Target 1, SUV with front passenger door and window open, Range =1.38 km [22]**

**Figure 17: Target 2, scale model of notional Trinidad satellite, range 1.38 km [22]**

**Figure 18: Target 3, 4.5 m ground station antenna, range 0.7 km [22]**

The first target was the SUV with the results shown in Figure 19. They were able to accurately estimate the range of the target with and estimated resolution of 0.6-0.8 m. The open window is apparent on the depth map compared to the rest of the vehicle.

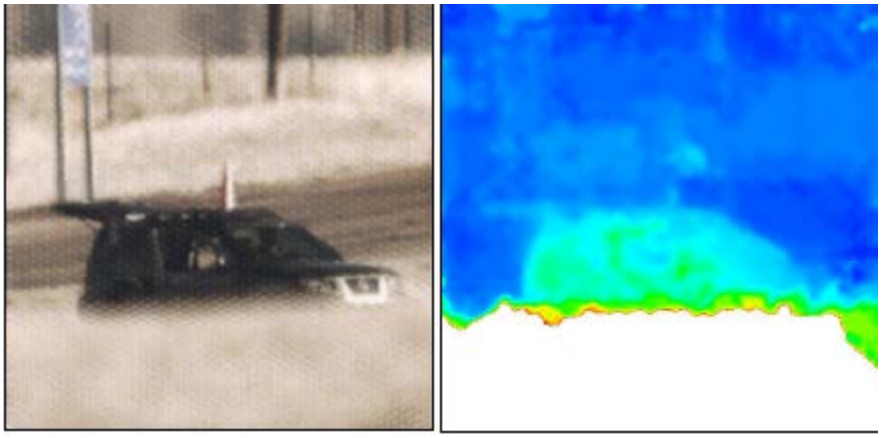


**Figure 19: Digitally refocused LF camera image (left), and depth map, telescope at f/8 (right) [22]**

During the testing of the next target, they discovered that there were better results when the f/# of the main aperture matched the f/# of the lenslet array. An example of this is shown in Figure 19. As the f/# mismatch increases, each lenslet is illuminated less, resulting in a reduction in the achievable angular and depth resolution.

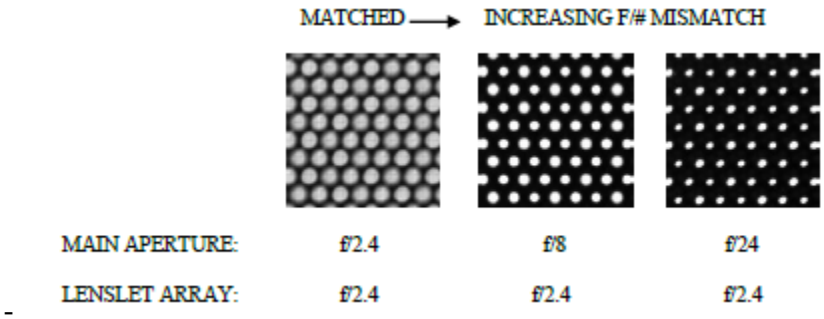



**Figure 20: Calibration images of lenslet pattern on sensor array show that an increasing f/# mismatch leads to a loss of angular and depth resolution [22]**

The results of the depth estimation for target 2 are shown in Figure 21 at two different f/#s. The results for f/5 were better than those for f/15 but an estimated range resolution was unable to be determined due to the low total depth of the target.

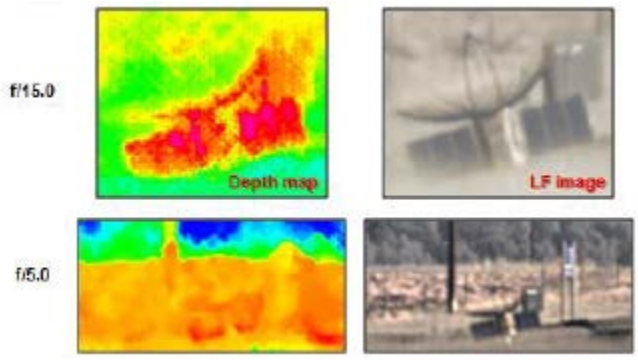



**Figure 21: Results of satellite model imaging, with telescope at different f/#s (top) f/15 and (bottom) f/5 [22]**

The results of the testing of the final target, the ground station antenna, are shown in Figure 22 below. With this test they were able to achieve an estimated range resolution of approximately 0.36 m.

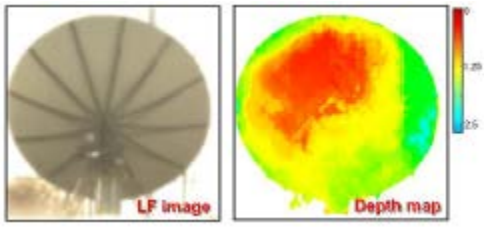



**Figure 22: Results of ground station antenna imaging with telescope at f/5 [22]**

The results of this research are promising from an accuracy standpoint but very little is said about how the data is processed or the speed of that processing. The processing algorithm that is stated is to generate an image stack of 2D images that are focused at different focused depths and then determine the distance to objects on a pixel by pixel basis using this refocused image stack with some additional processing options including change detection, face detection, and 3D reconstruction. There are no details on how the images are refocused, how distance is determined, or how long this process takes. This

research may be of use in the future for extending the accurate range of depth estimation by combining light field cameras with telescopes and should be examined [22].

### 2.5 Summary

This chapter provided a summary of the fundamentals of optics, the relevant theory of plenoptic cameras and the Hough transform that were used in the design of this light field depth estimation method, current hardware in the field of light field cameras, software used, and related research. This research will use the plenoptic theory along with the Hough transform to develop a real-time depth estimation method using the software discussed with data collected from Lytro light field cameras.

# III. Methodology

## Chapter Overview

This chapter provides a description of the method developed during the course of this research. First, Section 3.1 gives the specifics of the hardware and software systems used for development. Section 3.2 gives an overview of the depth estimation method used. Section 3.3 gives the experimental setup and finally, Section 3.4 summarizes this chapter.

## 3.1 Research Platform, Devices and Software

### 3.1.1 Hardware Description

The Lytro First Generation and Illum cameras, described in Section 2.2, were used for image acquisition. An Apple Mac Mini desktop was used for development and prototyping. Table 3 shows the computer specifications.

**Table 3: Computer specifications used in current research**

| | |
|---|---|
| Manufacturer | Apple |
| Model | Mac Mini |
| Processor | Intel Core i5 @2.8 GHz |
| Installed Memory (RAM) | 8.0 GB |
| Graphics Card | Intel Iris |
| Dedicated Graphics Memory | 1536 MB VRAM |
| Processor Cores | 2 |
| System Type | 64-bit |

### 3.1.2 Software

The method used in this thesis was written in MATLAB R2015B and uses the software packages described in Section 2.3 along with five unique routines developed to calculate the depth of the dominate linear feature(s) of a light field. There is one main script routine that calls four subroutines that load the light field, perform the Hough transform on it, take the gradient of it, and fill in areas of low confidence.

Table 4 gives an overview of the routines developed. There is a more in-depth explanation of each routine in Section 3.3 and the full code is shown in Appendix B.

**Table 4: Developed Matlab routines**

| Routine | Description | Toolboxes Required | Equations |
|---|---|---|---|
| real_time_depth | Main script that calls other functions, sets constants, crops light field, and displays final results | Matlab, Statistics and Machine Learning Toolbox, and Light Field Toolbox v0.4 | 12, 13, 14, 15 |
| load_LF_File | Loads light field into a single variable | Matlab and Light Field Toolbox v0.4 | N/A |
| Hough_Transform | Performs Hough transform, described in Section 2.1.3, on central image of light field to detect most prominent linear feature and outputs its location in the image | Matlab and Image Processing Toolbox | 4 |
| Take_Gradient | Computes the gradient, described in Section 3.2, of the cropped light field and outputs the depth map | Matlab and Light Field Toolbox v0.4 | 9, 10, |
| fill_nan | Fills holes in depth map where thresholding removed values | Matlab, Statistics and Machine Learning Toolbox | N/A |

### 3.2 Depth Estimation

Depth, or range, estimation is one of the key advantages that the light field gives, but accurately estimating depth for a whole light field is a computationally intensive process. In order for real time depth estimation, the time to calculate depth must be reduced either by reducing the size of the light field, or by reducing the accuracy of the depth estimation. Reducing the size of a light field can be done a couple of different ways; by cropping the *s*,*t* direction or in the u,v direction. Cropping in the *s*,*t* direction is

essentially the same as cropping a normal image, it's just done to every sub image in the light field. As long as the image isn't cropped too small, this reduces the size with minimal to no reduction in accuracy. Cropping in the *u,v* direction is essentially reducing the number of sub images, i.e. from a 81 sub images in a 9x9 square to 49 in a 7x7 square. While this can significantly reduce the size, it also results in a reduction in accuracy with each sub image that is removed. Reducing the accuracy of the gradient-based depth estimation can also be done a couple of different ways; by not using all three color channels, not using both vertical and horizontal directions, or by not thresholding. With three color channels in two different directions, there are a total of six gradient calculations that happen and are averaged to produce the final gradient values. Reducing the number of gradient calculation could increase the speed of the total gradient calculation by up to 83% if only one color channel in a single direction is used, but this could significantly affect the accuracy depending on what colors the image is primarily composed of and which direction the most significant gradients exist in. Since the depth information of the main object in the scene is the desired output, reducing the size of the light field by cropping in the *s,t* direction was implemented to decrease computation time without significantly affecting accuracy. This was done by using the Hough transform described in Section 2.1.3 to select the most prominent linear features in a scene and crop the light field around that line. This can be done for each linear feature with an increase in computation time to crop the image again and calculate the additional gradients.

The gradient-based estimation method begins with the Lambertian surface, a surface with ideal reflectance. A Lambertian surface can be considered an omnidirectional point light source in an infinitely small area, and exists in a light field as a plane of constant value. An example of this is shown in Figure 23 below in two dimensions.

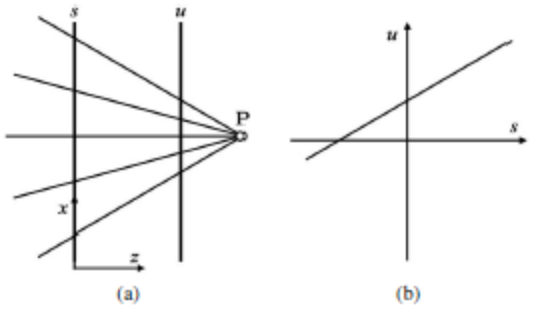



**Figure 23: 2D slice of point source of light with two reference planes (a) and a 2d slice of the corresponding light field [23]**

The orientation of these planes is dependent only on the depth of each point light source. Although two 4D vectors would normally be required to estimate the orientation of the plane in 4D, the orientation in light fields are the same in the *(s,u)* direction and the *(t,v)* direction. This allows for the use of a 2D gradient operator in both the *(s,u)* and *(t,v)* planes[3] and results in redundancy to help validate the results. Further redundancy is introduced because this is done in each color channel, increasing the total redundancy to a factor of 6. Observing an *(s,u)* slice of a simple light field, it is clear that a 2D gradient operator, applied at some point in the slice, will yield a gradient vector which points orthogonal to the plane passing through that point. When applied to a single color channel in the s and u directions, the 2D gradient operator is defined in Equation 6 as

$$\nabla^{su} L(n) = [\frac{\partial L(n)}{\partial s}, \frac{\partial L(n)}{\partial u}] \quad [6]$$

where *L(n)* is the discrete sampled light field, *n* is an index of the dimensions on the light field, *(s,t,u,v)*, $\frac{\partial L(n)}{\partial s}$ is the gradient in the *s* direction, and $\frac{\partial L(n)}{\partial u}$ is the gradient in the *u* direction. Given the gradient vector, the slope of the plane passing through each sample in the light field can be found, and from this slope the depth of the corresponding point in the scene can be found. From the point-plane

[3] Refer to Figures 6 through 8 for plane definitions.

correspondence, the slope of the plane corresponding to a point in the scene is given, in the *s* and *u* directions, by Equation 7

$$m_{plane} = 1 - \frac{d}{P_z} \quad [7]$$

where *d* is the separation of the reference planes, shown in Figure 24, which is determined during calibration, and $P_z$ is the range at that point.

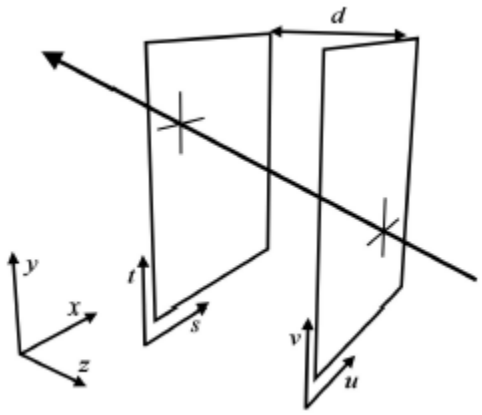



**Figure 24: Reference plane separation in two-plane parameterization of light rays [23]**

The direction of the gradient vector at the same point in the scene can also be expressed as a slope, as in Equation 8. This slope is the same slope as expressed in Equation 1, just defined using a different equation.

$$m_{\nabla}(n) = \frac{\nabla_u^{su} L(n)}{\nabla_s^{su} L(n)} = \frac{\frac{\partial L(n)}{\partial u}}{\frac{\partial L(n)}{\partial s}} \quad [8]$$

Assuming no regions of constant value, for which the magnitude of the gradient vector is zero, the gradient vector will always point orthogonal to the plane. This means that the relationship between the slope of the plane and the slope of the gradient vector can be expressed as $m_{plane} = -m_{\nabla}^{-1}$ . Rearranging to solve for the range at the point, $P_z$, yields Equation 9

$$P_z(n) = \frac{d}{1 + \frac{\frac{\partial L(n)}{\partial s}}{\frac{\partial L(n)}{\partial u}}} \quad [9]$$

which is easily generalized to the *t* and *v* dimensions. By applying Equation 9 throughout the light field, the depth of the scene that it models is estimated.

Because of the redundancy associated with having three independent color channels, with two independent depth estimates per channel, some method of optimally combining the estimates is in order. One way of doing this is to take the weighted sum of the six depth estimates, where the weight is taken as some measure of confidence. Given that the Euclidean norm of the gradient vector,$||\nabla L(n)||$, is essentially an indication of the contrast of the light field at each sample, it is a good indicator of confidence. Areas of low contrast, which yield little information about a scene's depth, will have small gradient norms, while areas of high contrast will have large gradient norms. The weighted sum can be expressed in Equation 10 as

$$\bar{P}_z(n) = \frac{\sum_{i=1}^{6} P_{z,i}\left|||\nabla_i L(n)||\right|}{\sum_{i=1}^{6} ||\nabla_i L(n)||} \quad [10]$$

where the i$^{th}$ gradient and depth estimate correspond to one of the six unique combinations of color channel and direction. Because the denominator of this expression is an indication of overall confidence, it can be used in a thresholding operation, which allows inadequate depth estimates to be ignored altogether, and become equal to the threshold when surpassed [23].

### 3.3 Experimental Setup

The first step in setting up the experiment was to calibrate the cameras. The Light Field Toolbox V0.4 calibration tool was used to determine the reference plane separation and other plenoptic intrinsic parameters, used in Equations 9 as *d*, in the horizontal, *du* with *s* and *u*, and vertical, *dv* with *t* and *v*, directions, shown in Table 5 [18, 24]. The calibration tool uses a

procedure that has some similarities to calibrating a standard camera, such as collecting images of a calibration grid like the one shown in Figure 25 from different poses. According to Dansereau, ten different poses is sufficient for calibration. This is a 24 x 24 grid with 7.4 mm x 7.4 mm squares and 15 poses were used in calibration to ensure the accuracy of calibration. The algorithm for this procedure is: process the images to form an initial pose and plenoptic intrinsic parameters, optimize it without using distortion parameters, and fully optimize it using distortion parameters.

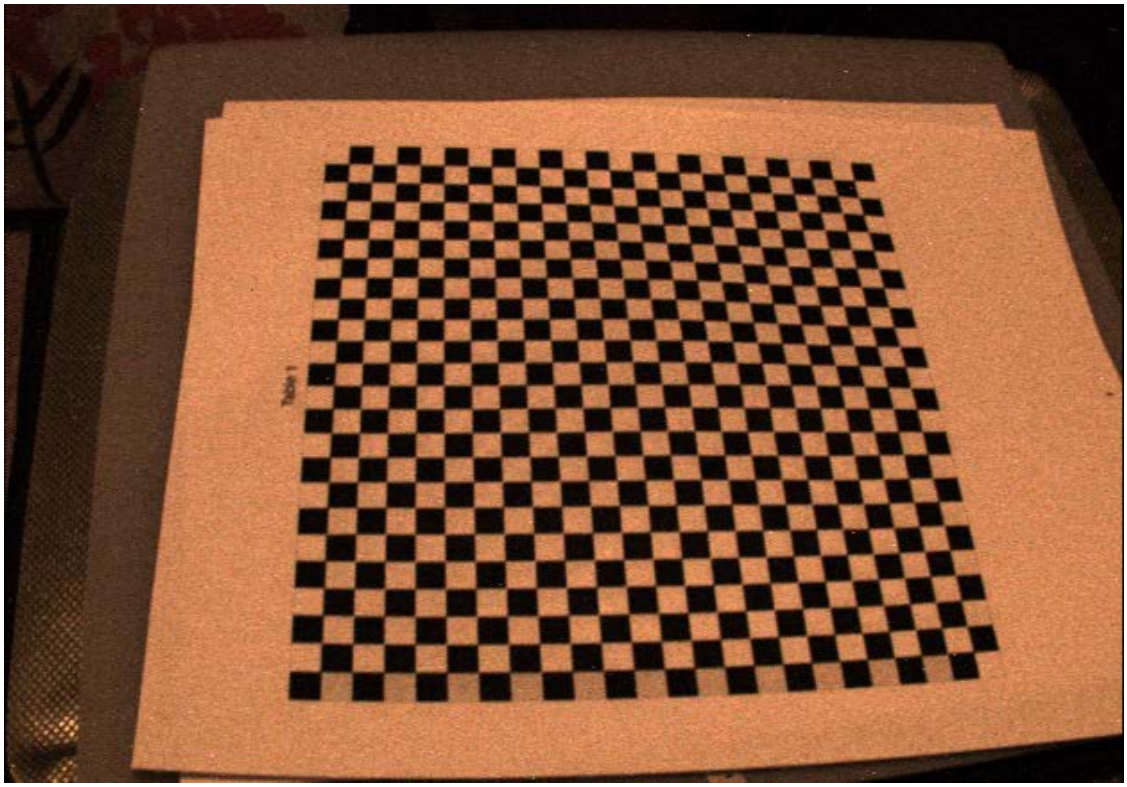


**Figure 25: Calibration image example**

**Table 5: Reference Plane Separation**

| | Illum | First Generation |
|---|---|---|
| *du* (cm) | 14.2827 | 10.0032 |
| *dv* (cm) | 12.2713 | 8.6651 |

After the camera calibration has been completed, the next step is to employ the depth estimation method developed in this thesis. The algorithm flow used in this method is shown in Figure 26 and an overview of the routines used was given in Table 4. First, a set of constants such as the light field name and thresholding parameters are set in the main routine, real_time_depth. Then, the load_LF_File routine is called to load the specified light field, such as the one shown below in the top image of Figure 27, and camera parameters into Matlab to be analyzed. The Hough_Transform routine is then called to perform the Hough transform on the light field locate the most prominent linear feature(s) in the central image of the light field and output the endpoints of the line that defines the linear feature. Before the Hough transform is performed the center subaperture image of the light field is converted to a black and white image using the edge function. This is because the transform can only be applied to two dimensional array and color images are three dimensional. The edge function was selected because some linear features may not be the same color and therefore may have been missed in other functions such as the image gradient function. The linear features are selected using a rho/theta map, such as the one in the bottom left image of Figure 27, and are shown as a red line for the most prominent linear feature and green lines for other prominent features in the top image of Figure 27. As stated above, this can be done for multiple linear features with an increase in computation time to crop the image again and calculate the additional gradients. The main routine will then use these points to create a rectangular area that the light field will be cropped to, such as in the bottom middle image of Figure 27. A small buffer zone is created outside this rectangular area so that the gradient can actually be calculated in the event that the linear

feature is nearly vertical or horizontal. The orientation and size of this area depends on the orientation and length of the linear feature located. The cropped light field and camera parameters are loaded into the routine Take_Gradient to compute the gradient map which is converted to a depth map using the specific camera parameters. Finally, the depth map is inputted into fill_nan to fill holes created by thresholding with the average values of the surrounding pixel and the depth map, such as in the bottom right image of Figure 27, is averaged to give an average scene depth. The bottom right image of Figure 27 only includes the depth for the bottom of the screwdriver shaft show in the middle image, not the entire depth map. Each cropped light field gives only one value of depth, but there can be multiple prominent linear features that are used to create cropped light fields so there are multiple cropped light fields. The depth value from a given cropped light field assumes that the background of the image has been sufficiently removed by thresholding, feature detection and cropping so that only the target data remains in the depth map giving an accurate approximation of the actual depth of the target.

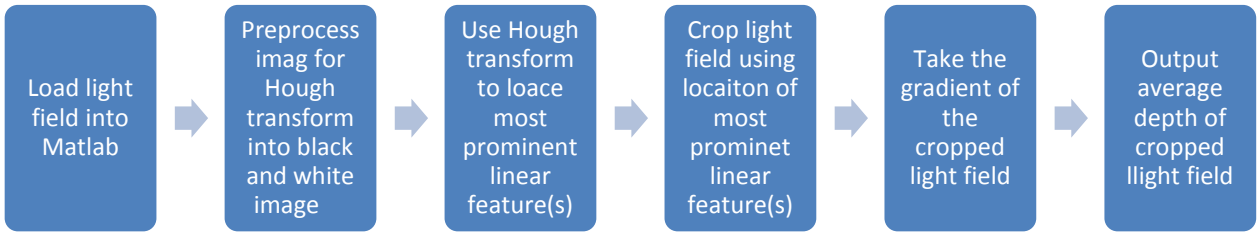



**Figure 26: Flow Diagram**

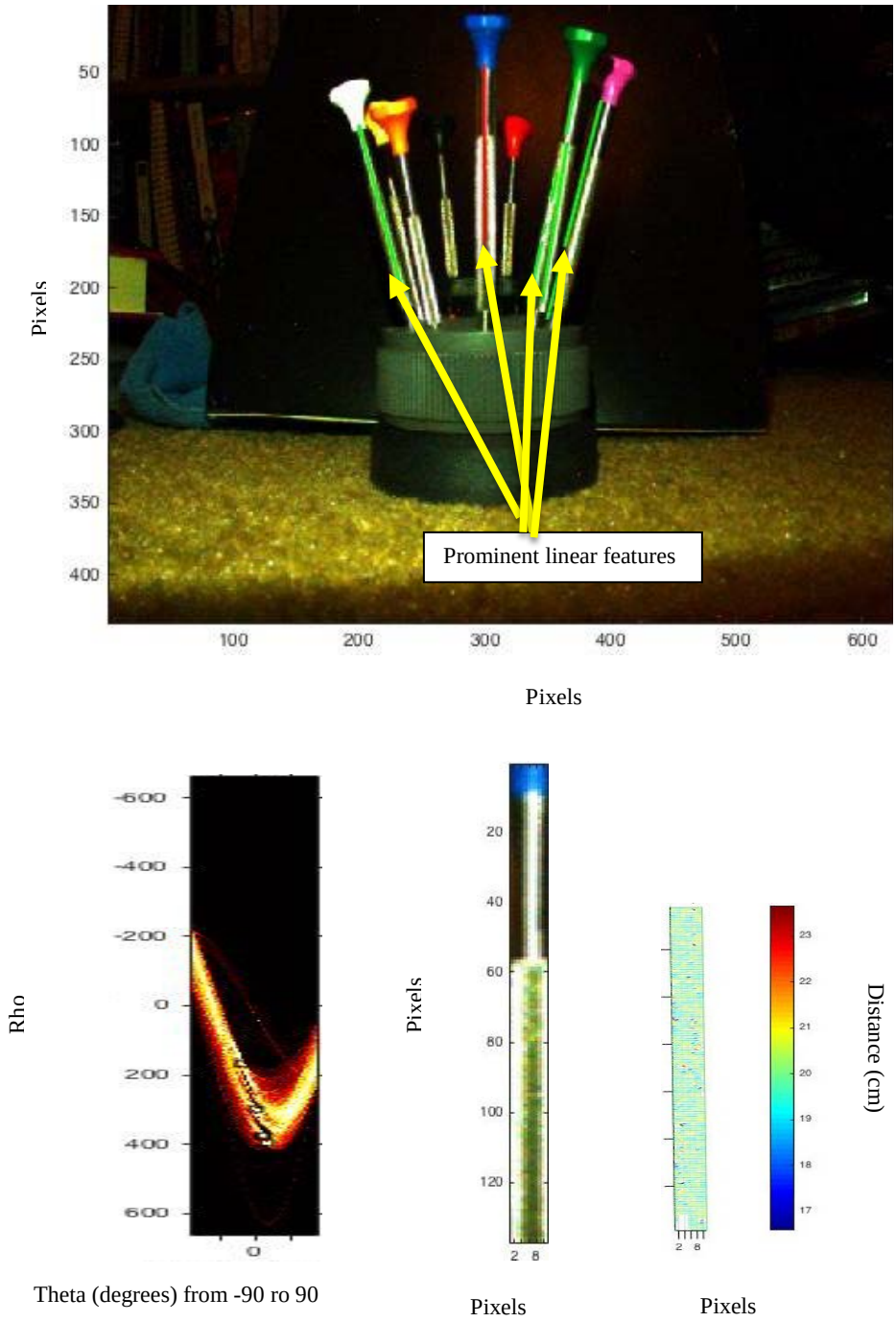



**Figure 27: Hough transform example, Initial image with linear features shown with green lines and most prominent in red (Top), Rho/Theta map (Bottom Left), image cropped around most prominent linear feature (Bottom Middle), depth map of bottom of screwdriver shaft and scale (Bottom Right)**

There are two qualities analyzed for this research, the accuracy of depth estimation and the speed of computation. The first analysis, the accuracy of depth estimation, analyzes three types of light fields, synthetic light fields from a data set shared by Wanner, Meister and Goldluecke and real light fields that were generated with two different Lytro light field cameras, whose specifications are shown in Table 6. Each of the color channels will be analyzed individually in each direction and as a whole.

**Table 6: Light field specifications**

| Light Field | Number of Subaperture Images | Subaperture Image Size (pixels) |
|---|---|---|
| Synthetic | 81 (9 x 9) | 768 x 768 |
| Illum | 225 (15 x 15) | 434 x 626 |
| First Generation | 121 (11 x 11) | 378 x 379 |

The synthetic light field, called Buddha, was taken from a data set shared by Wanner, Meister and Goldluecke, and includes a ground truth depth map [25]. The Lytro Illum and First Generation light fields, called 'screwdrivers' with an identifier for range and camera, were generated in a laboratory setting with the Lytro Illum and First Generation cameras respectively. The ground truth for the Lytro light fields was generated by using a tape measure, with a half centimeter uncertainty, to measure the distance from center of each target in the scene to the lens of the camera and then added to the principal plane. The center of the target refers to the middle of the shaft of each screwdriver as shown in Figure 28. The location of the principal plane, *x*, is determined by Equation 11

$$x = \frac{DF}{f + F - D} \quad [11]$$

where D is the distance from the main lens to the microlens array, F is the focal length of the main lens, and f is the focal length of the microlens [26].

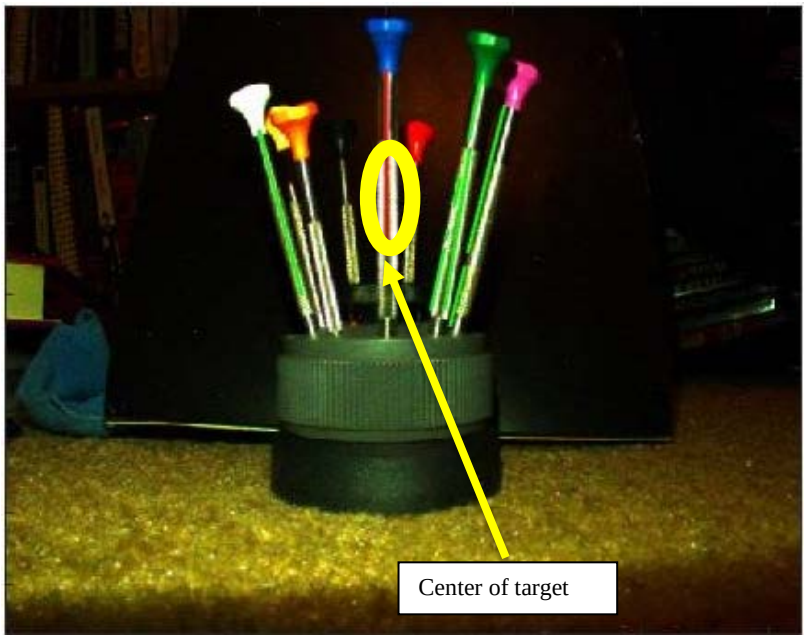



**Figure 28: Center of target diagram**

The percent of error instead of root mean square error is used to show if there is a trend towards over estimation or under estimation. The synthetic ground truth will be compared to the depth map generated from this depth estimation method and the error for each pixel will be calculated using Equation 12

$$Error_i = \frac{T_i - E_i}{T_i} * 100\% \quad [12]$$

where *T* is the ground truth depth, *E* is the generated depth, and *i* is the pixel. An example of this is shown in 29 where the top left is the truth, the top right is the estimate, and the bottom is the error between them. This is then used to generate a depth error map and to calculate the average error magnitude of the entire depth map using Equation 13

$$average\ error\ magnitude = \frac{1}{s * t} \sum_{i=1}^{i=s*t} ||Error_i|| \quad [13]$$

where *s* and *t* are the dimensions of the depth map in pixels. For the Lytro light fields, the depth of each target, or linear feature identified in the light field, was calculated by averaging the depth at the point where the ground truth depth was measured using Equation 14

$$D = \ average\ depth\ = \frac{1}{x * y}\sum_{i=1}^{i=x}\sum_{j=1}^{j=y} depth_{ij} \qquad [14]$$

where *depth* is the depth value for each pixel and *x* and *y* are the dimensions of the target area. The pixels for this target area were differentiated from the background pixels by only selecting a small sample around the center of the target. If the selected sample is too large, background pixels could skew the results. This average depth is then used to calculate the error with Equation 15

$$error = \frac{T - D}{T} * 100\ \% \qquad [15]$$

where *T* is the measured ground truth depth and *D* is the calculated average depth.

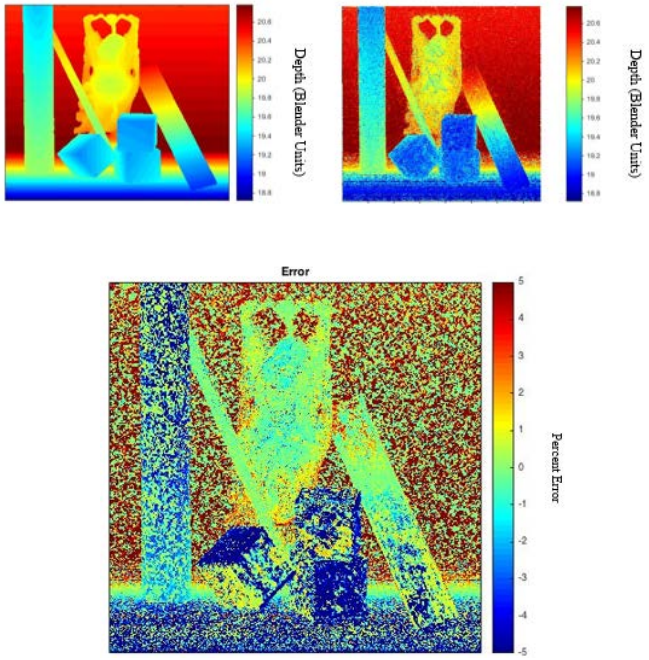



**Figure 29: Synthetic error example**

The second analysis was on the speed of computation. In any real-time system, calculation speed is a major consideration because if calculations take too long, then it is of no use in real-time systems. All

speed analysis was generated on the computer whose specifications are shown in Section 3.1.1. The first analysis was the calculation times of the gradient vs. the size of the light field. The gradient calculation gives the depth information so it is an essential part of the method. For each of the light field sizes, First Generation and Illum, four different light fields were analyzed; an actual light field taken with the camera, matrix of zeros the same size as each light field, a matrix of ones the same size as each light field, and a matrix of randomly generated numbers the same size as each light field. These light fields were selected to discover if the composition of the light field had any effect on the gradient computation time. The gradient was taken a total of 500 times for each of these light fields, with a different set of random number each time for the randomly generated numbers light field. Each light field was reduced in size by a factor of 1/50 until it was empty resulting in 50 different sizes for each light field. At each of these sizes the gradient computation time was calculated 10 times and averaged over those 10 times.

The requirements for these metrics were generated using similar real-time operations and the International Docking System Standard Interface Definition Document (IDSS IDD). Similar light field depth estimation techniques using different methods that are discussed in Dansereau's doctoral thesis approach real-time operation with run times of 0.5-2.6 seconds [27]. Using the lower end of these times, a threshold of 0.5 sec was established as the threshold for the computation run time. Since proximity operations in space are relatively "slow events," these real-time metrics may not need to be better define and may in fact be too strict for proposed applications. For the depth estimation accuracy, the IDSS IDD gives a maximum misalignment of 0.11 m for the docking system which is approximately 1.1 m in diameter [28]. This results in an allowable percent error of 10%. This requirement comes from a NASA standard addressing size, not range so real requirements will still need to be established in the future. A summary of the test metrics and tests are shown in Table 7.

**Table 7: Test metrics**

| Test | Light Field | Metric | Threshold | Algorithm |
|---|---|---|---|---|
| Depth Estimation Accuracy | Buddha | Percent Error | 10% | Entire Routine |
| Depth Estimation Accuracy | Screwdrivers – Close - Illum | Percent Error | 10% | Entire Routine |
| Depth Estimation Accuracy | Screwdrivers – Mid - Illum | Percent Error | 10% | Entire Routine |
| Depth Estimation Accuracy | Screwdrivers – Far - Illum | Percent Error | 10% | Entire Routine |
| Depth Estimation Accuracy | Screwdrivers – Close – F01 | Percent Error | 10% | Entire Routine |
| Depth Estimation Accuracy | Screwdrivers – Mid – F01 | Percent Error | 10% | Entire Routine |
| Depth Estimation Accuracy | Screwdrivers – Far – F01 | Percent Error | 10% | Entire Routine |
| Computation time | Multiple light fields of different sizes | Run time | 0.5 seconds | Entire Routine |
| Computation time | Multiple light fields of different sizes | Run time | 0.5 seconds | Take_Gradient |
| Computation time | Multiple light fields of different sizes | Run time | 0.5 seconds | Load_LF_File |

The hardware setup included the Lytro cameras and a screwdriver set, an example of which is shown in Figure 30. Natural light, the afternoon sun shining through the window, along with an overhead light were the lighting conditions for this experiment.

**Figure 30: Camera arrangement for sample image (screwdriver set)**

### 3.4 Summary

This chapter provided a description of the method developed during the course of this research. First, Section 3.1 gave the specifics of the hardware and software systems used for development. Then, Section 3.2 gave an overview of the depth estimation method used in the thesis. Finally, Section 3.3 gave the calibration method, experimental setup, metrics analyzed, and hardware setup in this thesis.

# IV. Analysis and Results

## Chapter Overview

This chapter provides the analysis and results of the experimentation using the method developed during the course of this research. First, Section 4.1 presents results of the analysis of the accuracy of the depth estimation method developed. Then, Section 4.2 presents the results of the analysis of the speed of computation of the depth estimates are presented. Section 4.3 summarizes the analysis and results presented in this chapter.

## 4.1 Accuracy of Depth Estimation

The first analysis was of the accuracy of the depth estimation of this method. The first light field analyzed was the synthetic light field scene, called Buddha[4], shown in Figure 31 with the associated ground truth and generated depth maps shown in Figure 32 [25]. The units used in the synthetic images are an arbitrary unit called blender units. Using Equation 12 and 13, the average error magnitude for the depth estimation of this light field was calculated to be 2.84% which meets the minimum accuracy requirement of <10% set in Section 3.3. Figure 33 shows this error on a pixel by pixel basis. There are areas in the generated depth map where it appears to be more accurate than the ground truth map on the dice. This is a result of the sharp contrast between the black and the white in the black circles on the dice where there isn't actually any depth difference here. The smooth areas around it are of low confidence while this area is of high confidence because of the sharp color difference so there appears to be a change in depth here when it is really only a more confident depth estimation here. These areas of sharp contrast seem to have greater depth accuracy than most other areas in the image, however, there seems to be a specific depth that has less error as well. In the Lytro camera images, difference in depth accuracy for the areas of sharp contrast was even more apparent, but there did not seem to be any indication that was a specific depth that was more accurate. This may be a consequence of the synthetic image. Figure 34

[4] This file is spelled Buhhda by Wanner and Golduecke

shows a confidence map for the synthetic image. Most of the image seems to be of low confidence unlike the generated light fields confidence maps, shown in Figures 37 and 39, where there are areas of very high confidence, and there does not seem to be much relation between confidence and error for the synthetic light field while it does appear to be related in the generated light fields. Figure 35 shows that there is however a relationship between contrast and error and also between range and error for the synthetic light field. The left plot of Figure 35 shows that as contrast increase there is a trend towards less error but there appears to be trend towards a -10% error that is unexplained and may be related to range. The right plot of Figure 35 shows that range is also related to error, with closer ranges having a positive percent error and farther having a negative percent error with the least error at around the mean depth of the image.

**Figure 31: Synthetic Buddha light field analyzed in accuracy analysis [25]**

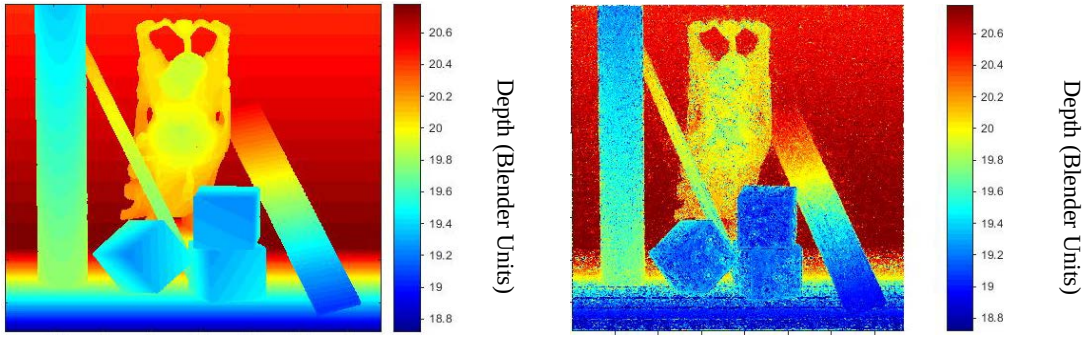



**Figure 32: Provided ground truth (left), and generated depth map for Buddha light field [25]**

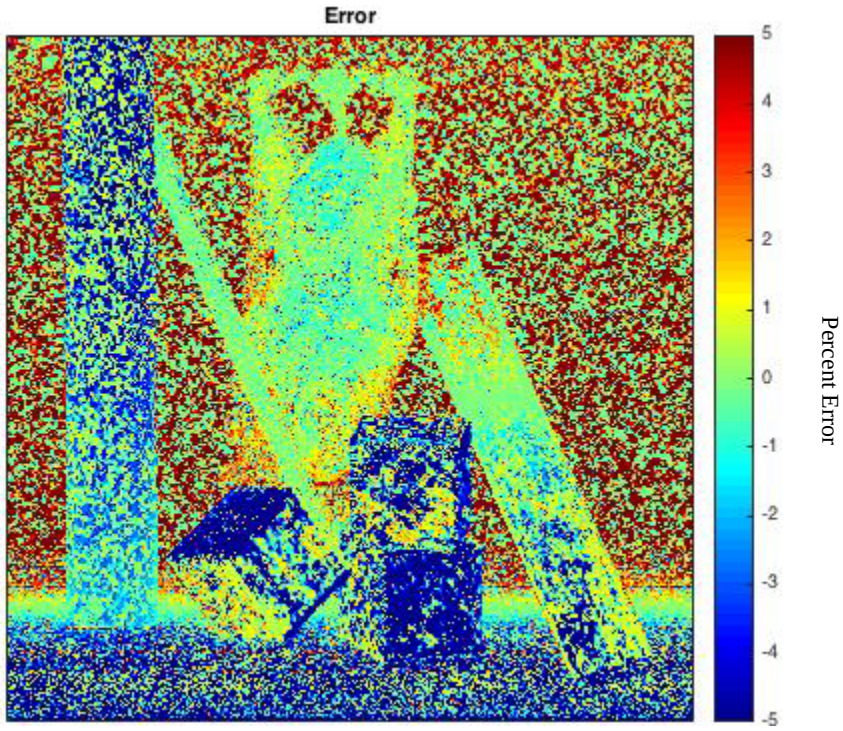



**Figure 33: Percent error for generated depth map from ground truth**

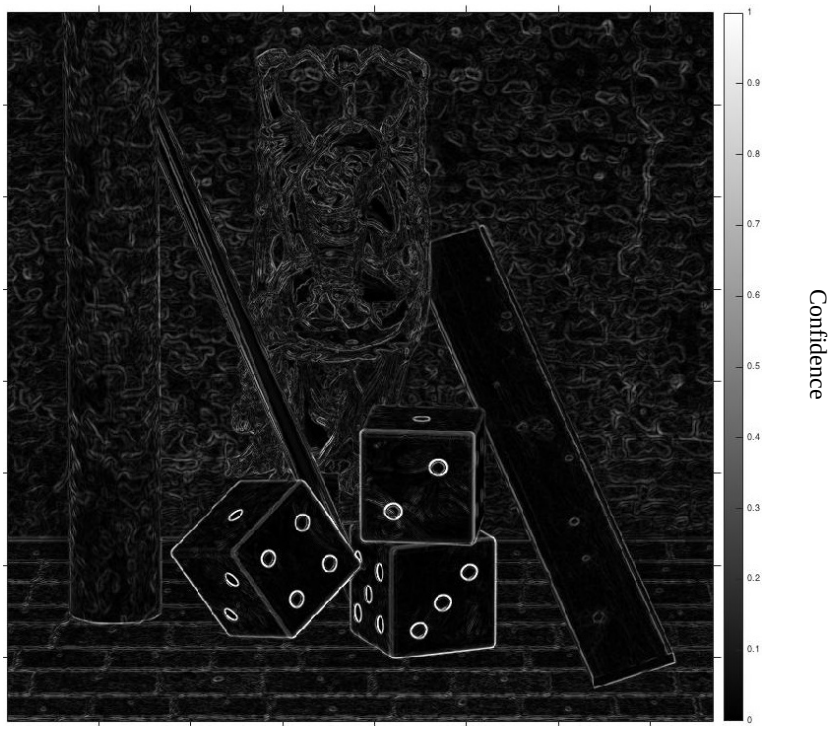



**Figure 34: Confidence map for Buddha light field**

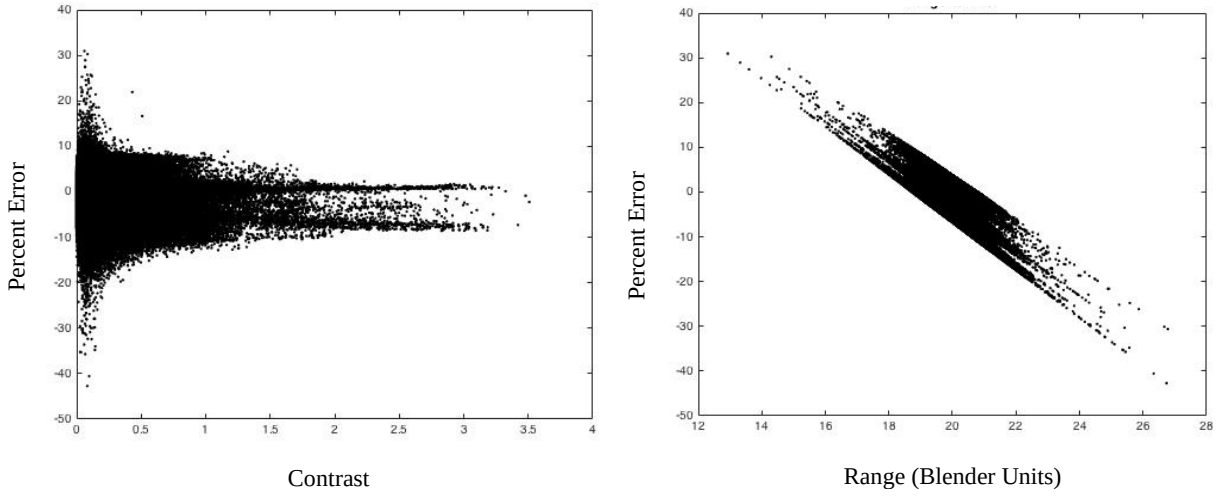



**Figure 35: Contrast vs Error plot (left) and Range vs Error plot (right) for the Buddha light field**

The next accuracy analysis will be for the Illum camera. Three cases are presented, one at a range of 20-30 cm, one at a range of 140-150 cm, and one at a range of 250-260 cm. The truth data for this was measured from the middle of the shaft of each screwdriver to the lens of the camera with a tape measure. Then the distance to the principal plane was added to get the true distance. The results were measured by averaging the depth of the middle of the shaft of each screwdriver and are shown, along with the truth data and percent error, which was calculated using Equations 14 and 15, in Table 8. The depth of each screwdriver was determined by cropping to the middle of the shaft of each screwdriver and averaging to find the value of depth at the middle of the shaft of each screwdriver. As the range increased, the amount of targets decreased because the resolution was not sufficient to support more. As shown, all the targets' ranges are estimated to within the minimum requirement of <10% error from Section 3.3 except for one target, the green screwdriver in the close range, that is significantly different than the others. The other target's ranges are all slightly underestimated. Figure 36 shows an example of the close range image and corresponding depth map and Figure 37 shows the corresponding confidence map. From the confidence map and depth map it is apparent that the areas of low confidence have been removed from the depth map using confidence as a threshold.

**Table 8: Results using Illum Camera**

| Close-Range | | | |
|---|---|---|---|
| Target Screwdriver | True Range +-0.5 (cm) | Estimated Range (cm) | Percent Error (%) |
| White | 23.9 | 23.7 | -2.8 to 1.3 |
| Orange | 22.1 | 20.5 | -9.3 to -5.1 |
| Black | 29.1 | 28.8 | -2.7 to 0.70 |
| Blue | 21.8 | 21.5 | -3.6 to 0.94 |
| Red | 29.6 | 27.2 | -9.6 to -6.5 |
| Green | 22.5 | 25.9 | 13 to 18 |
| Average Error Magnitude | N/A | N/A | 5.4 to 6.8 |
| Mid-Range | | | |
| Green | 142.6 | 130.3 | -9.0 to -8.3 |
| Yellow | 149.9 | 132.8 | -12 to -11 |
| Pink | 143.3 | 130.5 | -9.2 to -8.6 |
| Gray | 144.9 | 133.6 | -8.1 to -7.5 |
| Blue | 143.9 | 131.8 | -8.7 to -8.1 |
| Average Error Magnitude | N/A | N/A | 8.7 to 9.4 |
| Long-Range | | | |
| Gray | 256.4 | 235.9 | 7.8 to 8.2 |
| Blue | 255.5 | 238.8 | 6.4 to 6.7 |
| Average Error Magnitude | N/A | N/A | 8.7 to 9.4 |
| Total Average Error Magnitude | N/A | N/A | 6.9 to 7.9 |

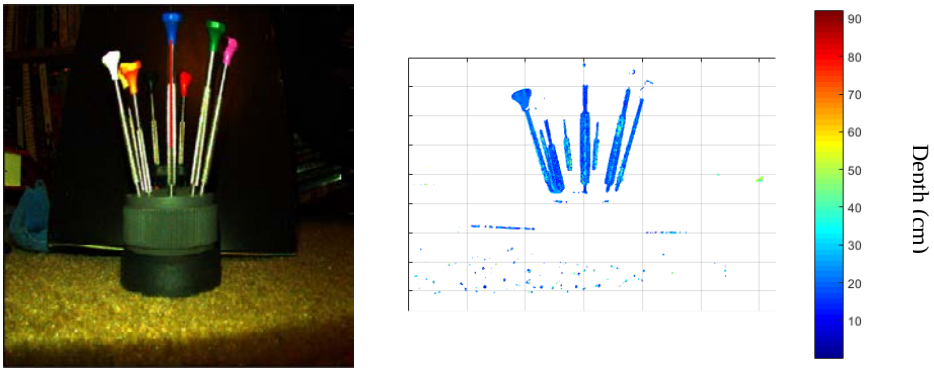



**Figure 36: Illum image to be analyzed (left), and resulting depth map (right)**


**Figure 37: Confidence map for Illum light field**

The next accuracy analysis will be for the First Generation camera at the same ranges and procedure as the Illum. The results of the accuracy analysis of each of the light fields are shown in Table 9. As shown, although a couple of the targets meet the error requirement individually, overall the average error is greater than the minimum requirement of <10% error from Section 3.3. Figure 38 shows an example of the close range image and corresponding depth map and Figure 39 shows the corresponding confidence map. From the confidence map and depth map it is apparent that the areas of low confidence have been removed from the depth map using confidence as a threshold. Some of the areas of medium confidence, such as the tops of the screwdrivers have not been removed but it is apparent that the depths displayed for them are of lower confidence that the screwdriver shafts and therefore less accurate.

**Table 9: Results using the First Generation Camera**

| Closer-Range | | | |
|---|---|---|---|
| Target Screwdriver | True Distance +-0.5 (cm) | Calculated Distance (cm) | Percent Error Range |
| White | 20.7 | 27.1 | 27 to 34 |
| Orange | 19.0 | 20.7 | 6.2 to 12 |
| Black | 26.0 | 29.7 | 12 to 16 |
| Blue | 18.9 | 15.6 | -19 to -15 |
| Red | 26.5 | 28.2 | 4.4 to 8.5 |
| Green | 19.4 | 16.8 | -15 to -11 |
| Average Error Magnitude | N/A | N/A | 14 to 16 |
| Mid-Range | | | |
| Green | 145.1 | 118.5 | 18 to 19 |
| Yellow | 152.4 | 115.2 | 24 to 25 |
| Pink | 145.8 | 1123 | 22 to 23 |
| Gray | 147.4 | 119.8 | 18 to 19 |
| Blue | 146.4 | 121.9 | 16 to 17 |
| Average Error Magnitude | N/A | N/A | 20 to 21 |
| Long-Range | | | |
| Gray | 259.0 | 220.0 | 14 to 15 |
| Blue | 258.0 | 212.5 | 17 to 18 |
| Average Error Magnitude | N/A | N/A | 16 to 17 |
| Total Average Error Magnitude | N/A | N/A | 17 to 18 |

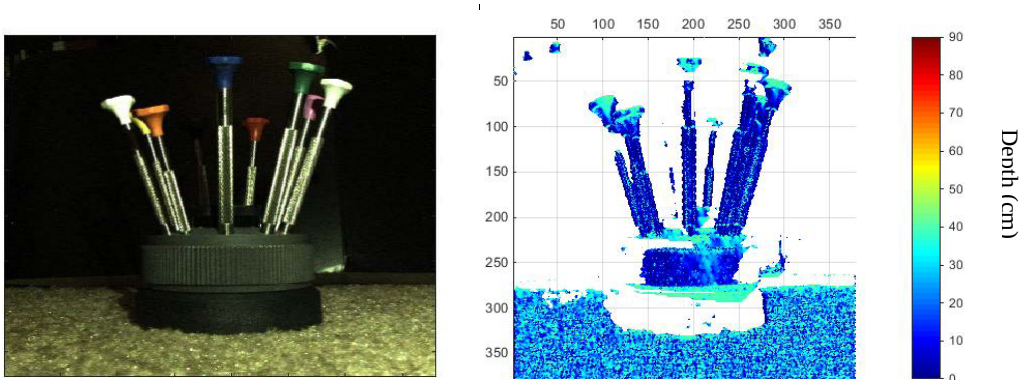



**Figure 38: First Generation image to be analyzed (left), and resulting depth map (right)**

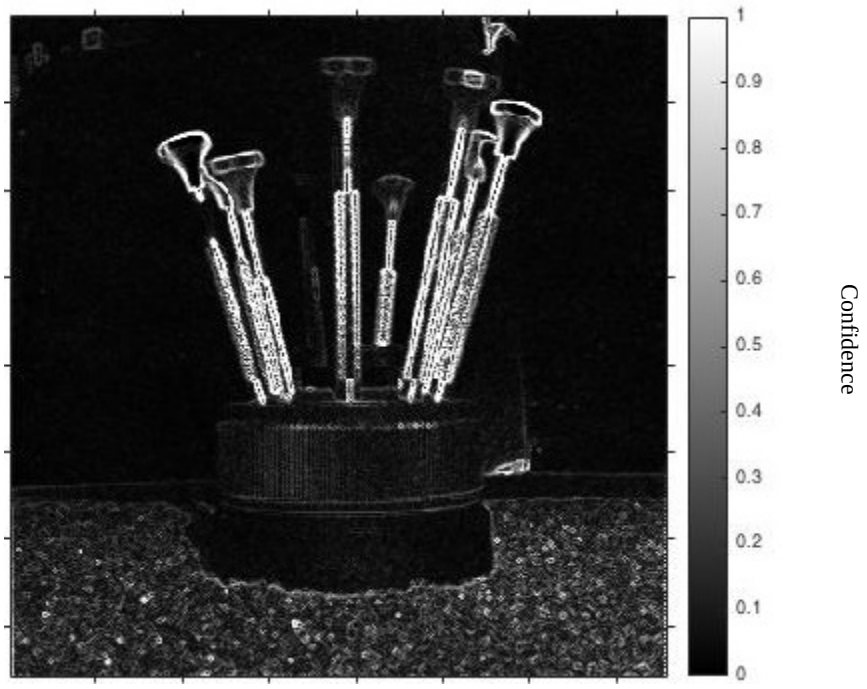



**Figure 39: Confidence map for First Generation light field**

Table 10 gives a summary of the results of this accuracy analysis. The Illum and First Generation average error is given in a range due to measurement uncertainty in the truth measurement from the three tests. It was apparent from the test that as range increased, accuracy generally decreased. The Illum was much more accurate than the First Generation. This can possibly be attributed to the larger lens and aperture of the Illum that allow for a greater perspective change across the light field. The difference may also be attributed to different sensors.

**Table 10: Depth accuracy analysis results**

| | Light Field | | |
|---|---|---|---|
| | Buddha | Illum | First Generation |
| Average Error Magnitude (percent error | 2.84 | 5.4 to 6.8 for close range<br>8.7 to 9.4 for mid-range<br>8.7 to 9.4 for far range | 14 to 16 for close range<br>20 to 21 for mid-range<br>16 to 17 for far range |
| Meets Minimum Requirement? | Yes | Yes | No |

During the course of this analysis, each color channel was analyzed separately to see if one had better performance. The result was greatly reduced accuracy with no significant difference in performance between the different colors overall. There was some variation on each light field, however, there was not one color that was best for each light field. The large decrease in performance of the accuracy estimation when only one color channel was used confirmed that choosing only one color channel would not be a suitable method for decreasing computation time.

### 4.2 Speed of Computation

The other analysis completed was measuring the speed of computation for the gradient calculation. The run time for the Take_Gradient routine, which calculates the gradient for a given light field using Equations 9 and 10, was found for the First Generation and Illum light fields. Figure 40 below shows the results of this analysis.

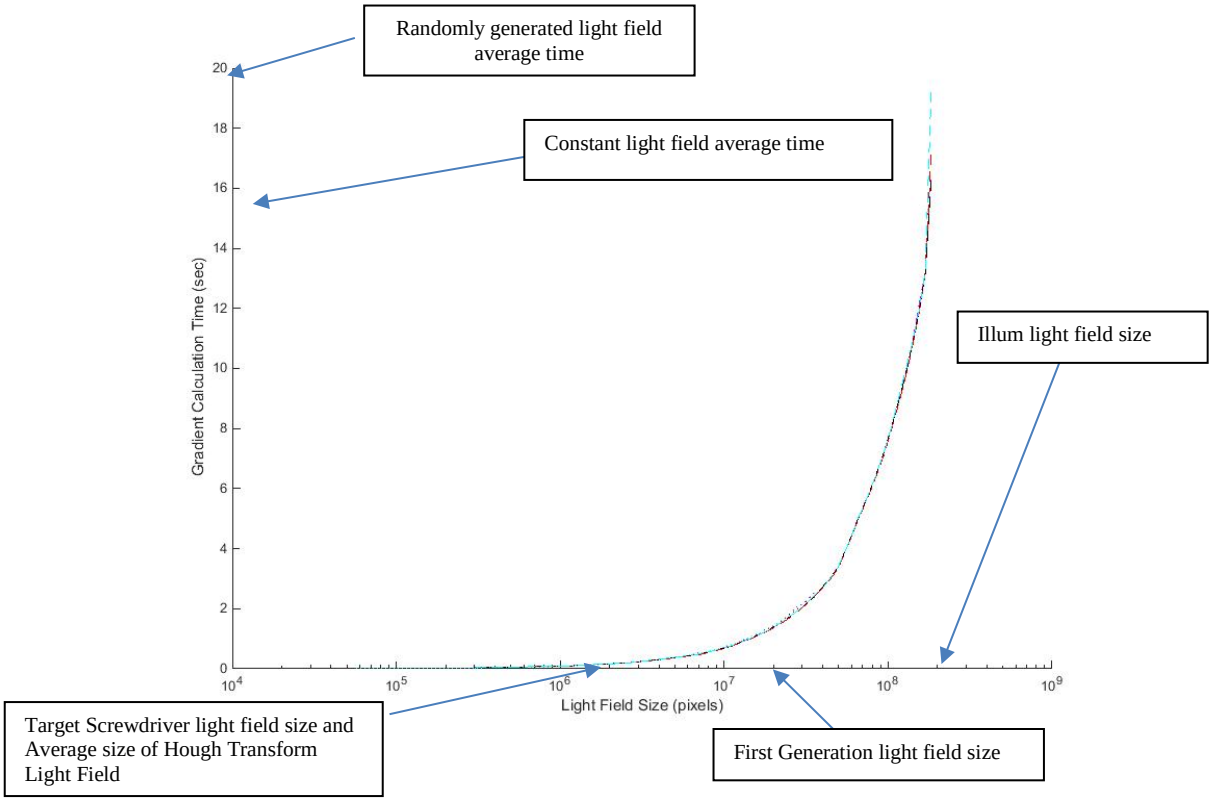



**Figure 40: Gradient calculation time vs light field size**

From Figure 39, it is apparent that as the size of the light field increases, the time required to compute the gradient increases at an exponential rate. There is a time difference of nearly 3 seconds between the randomly generated light field, taking approximately 19 seconds, and the constant light fields of ones and zeros, taking approximately 16 seconds with the actual light field falling in the middle, taking approximately 17 seconds on average. This is because the sparse nature of the gradient of the ones and zeros light fields leads to quicker calculations in the conversion from gradient to slope and depth while none of the random light field is sparse. There are areas of constant, or near constant value in the actual light field though that cause it to run faster than the random light field. The size of the First Generation light fields is considerably smaller than that of the Illum light fields, by approximately a factor of 5, and begins at the arrow in Figure 33. This leads to a calculation time for the whole of the light field being considerably less than the Illum light field as well, approximately 2.4 seconds. The Hough transform significantly reduces the size of each of these light fields and as a result leads to a much faster gradient

calculation time of approximately 0.2 seconds on average. This is a significant improvement from the multiple seconds needed for whole light fields however, the Hough transform takes approximately 1.5 seconds to calculate and reduce the light field's size so the true decrease in time is not as great. It is still a significant improvement though, especially for the Illum light fields.

Loading the light field was another area that took a significant amount of time, as shown in Table 11, it can take up to 15 seconds for larger synthetic light fields. This is unavoidable for this current research because the light field cannot be processed, cropped, without being loaded into the computer. However, this may not be an issue in a practical application as it will be integrated into the system but will likely have some effect so it will be discussed briefly here. The Illum light fields take between 2.5-6 seconds to load, as shown in Table 11 where it takes approximately 2.7 second, while the First Generation light field takes approximately 1 second.

**Table 11: Load time results**

| | Light Field | | |
|---|---|---|---|
| | First Generation | Illum | Synthetic |
| Load time (sec) | 1 | 2.7 | 15 |
| Size (pixels) | 2.5 *10^7 | 1.5*10^8 | 6*10^8 |

Between the loading the light field and taking the gradient, estimating depth can take anywhere between approximately 3 seconds and nearly a minute to calculate. The lower end of that spectrum is near real-time but with a larger light field, that is not possible even if only a small portion of the light field is analyzed for depth because of loading time. The average run time for the Illum light fields was approximately 6 seconds after reducing its size considerably by selecting significant regions with the Hough transform for the gradient calculation. As with the previous camera, the majority of this is time is

spent loading, which averages just under 3 seconds for the Illum light fields, and calculating the Hough transform, which averages approximately 1.5 seconds. There is a significant reduction in time for the gradient calculation though, from nearly 16 seconds to less than 0.25 seconds on average which reduces the total time considerably. In the example shown in Table 12, the gradient calculation was reduced to 0.149 seconds using the Hough transform but the Hough transform took 1.76 seconds to run. The average run time for the First Generation light fields was approximately 3 seconds after reducing its size with the Hough transform, which is a little more than half of its average run time of approximately 7 seconds for the entire light field. The majority of this time is spent loading, which averages about 1 seconds for the First Generation light fields, and calculating the Hough transform which averages approximately 1.5 seconds. This decreases the gradient calculation time from approximately 2.4 seconds on average to less than 0.2 seconds on average, which does not reduce the total time considerably when the Hough transform time is taken into account. Neither the Illum nor the First Generation meet the minimum computation time requirement of 0.5 seconds. Table 12 shows an example profile summary from an Illum light field. It shows the time that each function takes to run and how much of that time is that function running and how much it is calling another function. For example, real_time_depth is the main function and it takes a total of 5.705 seconds to run but most of that time, 5.476 seconds, is taken up by other functions that were called in real_time_depth. The function load_LF_File takes nearly the entire 2.711 seconds of its run time to load the light field however and only 0.004 seconds calling other functions. An expanded profile summary is shown in Appendix A.

**Table 12: Profile summary from Illum light field**

**Profile Summary**

*Generated 01-Mar-2016 08:08:29 using performance time.*

| Function Name | Calls | Total Time | Self Time* | Total Time Plot (dark band = self time) |
|---|---|---|---|---|
| real_time_depth | 1 | 5.705 s | 0.218 s | |
| load_LF_File | 1 | 2.711 s | 2.707 s | |
| Hough_Transform | 1 | 1.730 s | 0.236 s | |
| imshow | 2 | 0.614 s | 0.066 s | |
| colorbar | 3 | 0.546 s | 0.013 s | |
| initSize | 1 | 0.418 s | 0.024 s | |
| movegui | 1 | 0.378 s | 0.326 s | |
| edge | 1 | 0.335 s | 0.152 s | |
| ColorBar.ColorBar>ColorBar.ColorBar | 3 | 0.300 s | 0.060 s | |
| ColorBar.doSetup | 3 | 0.180 s | 0.018 s | |
| ColorBar.ColorBar>ColorBar.set.Axes | 3 | 0.164 s | 0.001 s | |
| imfilter | 2 | 0.164 s | 0.004 s | |
| ColorBar.ColorBar>ColorBar.set.Axes_I | 3 | 0.161 s | 0.002 s | |
| ColorBar.ColorBar>ColorBar.setAxesImpl | 3 | 0.159 s | 0.002 s | |
| imadjust | 1 | 0.156 s | 0.003 s | |
| Take_Gradient | 1 | 0.149 s | 0.061 s | |

## 4.3 Summary

This chapter provided the analysis and results of the experimentation done on the method developed during the course of this research. First, Section 4.1 presented the results of the analysis of the accuracy of the depth estimation method developed. Then, Section 4.2 presented the results of the analysis of the speed of computation of the depth estimates are presented.

# V. Conclusions and Recommendations

## Chapter Overview

This chapter provides a summary of the research and analysis completed for this thesis. It gives the conclusions and significance of the research and recommendations for future work.

## 5.1 Conclusions of Research

During the course of this research the following investigative questions, given in Section 1.3, were answered:

- What is the availability of commercial light field hardware?
    - Currently there are two companies commercially offering plenoptic cameras. In its current state with the proposed depth estimation method, Lytro offers reasonable depth estimation and is readily available. Using Raytrix may also need to be explored because they are designed for industrial and scientific use and require a greater depth confidence than the average consumer, so they may offer better results.
- What accuracy of range estimation is attainable from light field cameras?
    - The average range error from the Lytro Illum and First Generation was 6.9-7.8% and 17-18% respectively. At close range the error war approximately 1% better for both cameras. The Illum was within the minimum accuracy requirement of 10% from Section 3.3 while the First Generation was not but once a the requirements are better defined, it may.
- What is the speed of computation of range estimation/ is it suitable for real-time range estimation?
    - The speed of computation of range estimation ranges between approximately two seconds and nearly 30 seconds. At the current time, the Lytro First Generation and Illum are suitable for near real-time operations by using the Hough transform to

reduce the size of each light field. This brings the run time down to approximately 2 seconds, which approaches real-time but is in the upper portion of Dansereau's range. Therefore, the speed of computation is not currently suitable for the real-time operational requirements established here with this computer hardware, but once better requirements are defined, different hardware used, and the method is optimized, this may become suitable.

- Is custom processing needed and if so, in what specific areas?
    - Custom processing is needed for real-time application to reduce the size of a light field by selecting specific features before analyzing for depth. This is because the gradient calculation to determine depth for the entire light field takes too long so the size of the light field needs to be reduced and depth determined only for specific features within the light field.

Based upon the answers to these questions, this research concludes that Lytro light field cameras may be suitable for use on spacecraft for proximity operations with optimization of processing. The accuracy of depth estimation for the Illum light fields were within the minimum requirement and it is possible to output an estimate of depth for a small portion of the light field in real-time. The Hough transform is a good way to select features in a light field, especially if multiple features are desirable, because it only needs to be performed once in to detect all of the linear features in the image. Also, the Hough transform can significantly reduce the processing time, especially for larger light fields such as the Illum light fields. During the course of this research there were some other notable conclusions that were able to be drawn. First, using single color channel instead of all three in order to reduce the gradient calculation time is not a practical because it greatly reduces the accuracy of the depth estimation. Second, when cropping, a buffer zone needs to be established around the end points for the linear feature so that in the event of a vertical or horizontal linear feature, the cropped light field is large enough to generate a meaningful gradient. Thirdly, lighting was found to have a significant effect on the quality of the depth

estimation. This was not investigated, but it was noted that images taken with better lighting conditions of the same target generated better results than those taken in lower quality lighting conditions. Lastly, as range increases, there is a decrease in the accuracy of the depth estimation, and this should be investigated in greater detail in future work.

### 5.2 Significance of Research

This research contributes to the research into implementing a plenoptic camera for real-time depth estimation for proximity operations and docking. It showed that a full light field cannot be analyzed using this method in real-time without considerable computing power. It also showed a small portion of a light field can be analyzed using this method in near real-time and provide usable depth information.

### 5.3 Recommendations for Future Research

There is much that can be done in the future regarding this research. First, light field cameras by Raytrix should be investigated as they are tailored more for industrial and scientific purposed and may provide increased depth accuracy and speed. Raytrix cameras are already used in industry for very close depth map generation for detecting defects in different types of components. Since it is being used by industry, Raytrix may have smaller file sizes and/or already customized software that would lead to faster processing.

Testing for depth accuracy at a greater distance using both Lytro and Raytrix cameras is also recommended to determine the range at which commercially available light field cameras would be suitable for depth estimation and to see what the max distance that usable depth information can be gathered. This would also be an opportunity to see how adding additional lenses, such as a telescope, would impact depth accuracy. There is already research on using a telescope lens on the end of a Raytrix camera that can be expanded on [22].

Optimization of this method, such as reducing extraneous display functions that have no value in real world applications, should be accomplished to determine if run time can be reduced along with implementing method on computer hardware that is comparable to space certified hardware to give a more accurate view of how suitable this method is for real world applications. In addition to optimization, additional depth estimation methods and methods to select features beyond gradient based methods and the Hough transform respectively should be explored and compared to determine if another method is more suitable for real-time operations.

The use of relative motion could lead to improvements in depth estimation. Although relative motion does not need to be accounted for, accounting for it could lead to high fidelity depth estimation and opens doors for multiple ways to analyze and validate data.

The effect of lighting should be studied more to determine how much of an affect it has on depth estimation and to use exoatmoshperic lighting conditions to imitate real applications.

### 5.4 Summary

The depth estimation method presented in this research accomplished the goals of this research. There are currently 2 companies that offer commercially available light field cameras, with the Lytro Illum and First Generation cameras having an accuracy of approximately 7% and 18 % respectively. The depth estimation method developed in this thesis took approximately two seconds to run after custom processing by using the Hough transform to reduce the size of the light field. Without the Hough transform run time increased by at least 100% depending on the size of the light field. This confirmed the need for custom processing to reduce the size of a light field before it is analyzed. This work is a step in the right direction toward putting a light field camera in space for to use for proximity operations and docking.

## Appendix A

An example of the Profile Summary from the running of the real-time depth code. It shows the time that each function takes to run and how much of that time is that function running and how much it is calling another function. For example, real_time_depth is the main function and it takes a total of 5.705 seconds to run but most of that time, 5.476 seconds, is taken up by other functions that were called in real_time_depth. The function load_LF_File takes nearly the entire 2.711 seconds of its run time to load the light field however and only 0.004 seconds calling other functions.

| Function Name | Calls | Total Time | Self-Time* | Total Time Plot (dark band = self-time) |
|---|---|---|---|---|
| real_time_depth | 1 | 5.705 s | 0.218 s | |
| load_LF_File | 1 | 2.711 s | 2.707 s | |
| Hough_Transform | 1 | 1.730 s | 0.236 s | |
| imshow | 2 | 0.614 s | 0.066 s | |
| colorbar | 3 | 0.546 s | 0.013 s | |
| initSize | 1 | 0.418 s | 0.024 s | |
| movegui | 1 | 0.378 s | 0.326 s | |
| edge | 1 | 0.335 s | 0.152 s | |
| ColorBar.ColorBar>ColorBar.ColorBar | 3 | 0.300 s | 0.060 s | |
| ColorBar.doSetup | 3 | 0.180 s | 0.018 s | |
| ColorBar.ColorBar>ColorBar.set.Axes | 3 | 0.164 s | 0.001 s | |
| imfilter | 2 | 0.164 s | 0.004 s | |
| ColorBar.ColorBar>ColorBar.set.Axes_I | 3 | 0.161 s | 0.002 s | |
| ColorBar.ColorBar>ColorBar.setAxesImpl | 3 | 0.159 s | 0.002 s | |
| imadjust | 1 | 0.156 s | 0.003 s | |
| Take_Gradient | 1 | 0.149 s | 0.061 s | |
| imadjust>parseInputs | 1 | 0.148 s | 0.005 s | |
| stretchlim | 1 | 0.142 s | 0.013 s | |
| imfilter>filterPartOrWhole | 2 | 0.141 s | 0.125 s | |

| | | | | |
|---|---|---|---|---|
| LFDisp | 3 | 0.140 s | 0.066 s | |
| hough | 1 | 0.138 s | 0.128 s | |
| mat2gray | 1 | 0.135 s | 0.006 s | |
| legendcolorbarlayout | 3 | 0.132 s | 0.047 s | |
| imlincomb | 1 | 0.129 s | 0.124 s | |
| imhist | 1 | 0.128 s | 0.124 s | |
| ColorBar.doMethod | 3 | 0.117 s | 0.002 s | |
| ColorBar.doMethod>set_contextmenu | 3 | 0.116 s | 0.044 s | |

## Appendix B

The code developed in this research is shown in this appendix including the following routines: Real_Time_Depth, load_LF_File, Hough_Transform, Take_Gradient, and Fill_nan.

**Real_Time_Depth routine**

```
 %% Depth Estimation
clear all
close all
clc;format short
profile on
n=100;
M=magic(n);

%% Set thresholding parameters
MinGradient = 0.0;
MaxSlope=5;
MinSlope=-5;
MaxRange=40;
MinRange=0;
%% Set Camera Intrinsics
%F01
Q=3
G=1;
if Q==1
    H =[0.0004427694605,0,-0.0003784125047,0,0.06924175913;...
        0,0.0003725917669,0,-0.0003957133815,0.07275213519;...
        -0.0008919267621,0,0.00182734436,0,-0.3418438678;...
        0,-0.0007489869818,0,0.001858191519,-0.347633371;...
        0.1455    0.1497   -0.3428   -0.3454    1.0000].*G;
    dh=10.0032*G;
    dv=8.6651*G;
    h=[0.0065 4.9532 6.0749].*G;
%Illum
elseif Q==2
    H=[0.000321768942,0,-0.0004386720945,0,0.1349495501;...
           0,0.0003127771329,0,-0.0004404492897,0.09329550344;...
           -0.000314804311,0,0.001655464672,0,-0.5164697402;...
           0,-0.0003172672541,0,0.001659341736,-0.3583686896;...
           0,0,0,0,1].*G;
    dh=14.2827*G;
    dv=12.2713*G;
    h=[0.0096 0.1680 9.8520].*G;
elseif Q==3
    H=[9.375 22.4999 22.4999];
    h=0;
    dh=48;
    dv=48;
    MinGradient = 0.00;
    MaxSlope=10;
```

```
    MinSlope=-10;

%     MaxRange=20.7812;
%     MinRange=18.7188;
end
%% Load file and display image

% LF=load_LF_File('Images/ScrewDrivers/IMG_0110__Decoded.mat');
LF=load('Images/LF_HCI_Budda.mat');
LF=LF.LF_HCI;
gt=load('Images/GT_Depth.mat');
gt=gt.gt;
% LF=load_LF_File('Images/F01/IMG_0001__Decoded.mat');
% LF=load_LF_File('Images/Illum/IMG_0029__Decoded.mat');
% LF=load_LF_File('BUNNY.mat');


LFDisp(LF);
%% Hough Transform
% select Hough, 1
select=1
if select==1
    xy_long=Hough_Transform(LF);
end

%% Crop light field to longest line
%define crop boundry
LF_size=size(LF);
ymax=LF_size(4);
xmax=LF_size(3);
if exist('xy_long','var')
    ad=5;
    mins=min(xy_long);
    maxes=max(xy_long);
    y1=mins(1)-ad;
    y2=maxes(1)+ad;
    x1=mins(2)-ad;
    x2=maxes(2)+ad;
elseif exist('boundary','var')
    y1=boundary(3);
    y2=boundary(4);
    x1=boundary(1);
    x2=boundary(2);
else
    x1=1;y1=1;x2=xmax;y2=ymax;
end
%make sure is in bounds of image
if y1<1
    y1=1;
end
if y2>ymax
```

```
    y2=ymax;
end
if x1<1
    x1=1;
end
if x2>xmax
    x2=xmax;
end
% x1=140;x2=255;y1=50;y2=190;
LF_crop=squeeze(LF(:,:,x1:x2,y1:y2,:));
% [LFcrop,rect2]=imcrop(squeeze(LF(5,5,:,:,1:3)));
% figure
%%
% y1=ceil(rect2(1));
% y2=ceil(rect2(1)+rect2(3));
% x1=ceil(rect2(2));
% x2=ceil(rect2(2)+rect2(4));
% LF_crop=squeeze(LF(:,:,x1:x2,y1:y2,:));
% show cropped image
LFFigure;
LFDisp(LF_crop);
axis image
title('In');
%% Take the gradient
slope=Take_Gradient(LF_crop,MinGradient,MaxSlope,MinSlope,H,dh,dv,MaxR
ange,MinRange,h);



%% Fill and display final
figure;
F=size(slope);
[X,Y]=meshgrid(1:F(2),1:F(1));
waterfall(X,Y,slope)
colormap jet
colorbar
axis image
title('Slope not filled')

Slope=fill_nan(slope);

% figure;LFDisp(Slope);
figure;
waterfall(X,Y,Slope)
colormap jet
colorbar
axis image
title('Slope')
Depth=nanmean(Slope(:))
% display ground truth
if exist('gt','var')
```

```
    caxis([18.7188,20.7812]);
    figure;LFDisp(gt)
    colormap jet
    colorbar
    G=squeeze(gt(5,5,:,:));
    er=sqrt(((G-Slope).^2)/length(G));

    figure;LFDisp(er);
    colormap jet
    colorbar
    axis image
    title('Error')
    er=abs(er);
    error=nanmean(er(:))
end

profile viewer
```

**load_LF_File routine**

```
function LF=load_LF_File(file_location)

load(fullfile(file_location),'LF');
LF=LF;
end
```

**Hough_Transform routine**

```
function xy_long=Hough_Transform(LF)
S=size(LF);
x=floor(S(1)/2);
y=floor(S(2)/2);
f=S(3);
q=S(4);
LFrot=imrotate(squeeze(LF(x,y,5:(f-5),5:(q-5),1)),0,'crop');
LFrot=imcrop(LFrot,[0,0,626,250]);
% figure(1),imshow(LFrot)
BW=im2bw(LFrot,0.2);
BW=edge(LFrot);
figure(2), imshow(BW)
% Compute Hough Transform
[H,theta,rho]=hough(BW);
% Display
figure(3),
imshow(imadjust(mat2gray(H)),[],'XData',theta,'YData',rho,'InitialMagn
ification','fit');
xlabel('/theta (degrees)'),ylabel('/rho');
colormap(hot)
```

```
% Find peaks of Hough tranform and display
P=houghpeaks(H,5,'threshold',ceil(0.3*max(H(:))));
x=theta(P(:,2));
y=rho(P(:,1));
hold on
plot(x,y,'s','color','black');
% Find lines in images

lines=houghlines(BW,theta,rho,P,'FillGap',5,'MinLength',5);
% Plot lines on original image
figure(1),hold on
max_len=0;
X=zeros(length(lines),2);
Y=zeros(length(lines),2);
for k=1:length(lines)
    X(k,:)=lines(k).point1;
    Y(k,:)=lines(k).point2;
    xy=[lines(k).point1;lines(k).point2];
%     plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');
    % Determine endpoint of longest lines segment
    len=norm(lines(k).point1-lines(k).point2);
    if (len>max_len)
        max_len=len;
        xy_long=xy;
    end
end
% highlight the longest line segment
plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','red')
hold off
end
```

**Take_Gradient routine**

```
function
Pz=Take_Gradient(LF,MinGradient,MaxSlope,MinSlope,H,dh,dv,MaxRange,Min
Range,h)
PzEst=zeros(size(LF,3),size(LF,4));
% take gradient for each color channel
for ctr=1:3

    LF_temp=squeeze(LF(:,:,:,:,ctr));
    LF_temp=LFConvertToFloat(LF_temp,'double');
    [Ls,Lt,Lu,Lv]=gradient(LF_temp); %careful of the ordering of
outputs from the

    % gradient function
    % Reduce size
    S=size(Ls);
    x=floor(S(1)/2);
```

```
    y=floor(S(2)/2);
    Ls=squeeze(Ls(x,y,:,:));Lt=squeeze(Lt(x,y,:,:));
    Lv=squeeze(Lv(x,y,:,:));Lu=squeeze(Lu(x,y,:,:));

    Mag1=abs(Lu);
    Mag2=abs(Lv);
    TotMag=Mag1+Mag2;
    Slope1=Ls./Lu;
    Slope2=Lt./Lv;
    Slope1(TotMag<MinGradient)=NaN; % Threshold to remove invalid
values
    Slope2(TotMag<MinGradient)=NaN;
    SlopeEstTemp1=(Mag1.*Slope1+Mag2.*Slope2)./TotMag;
    InvalidMask=isnan(SlopeEstTemp1);
    Slope1=max(MinSlope,min(MaxSlope,Slope1));% Threshold to saturate
to in range values
    Slope2=max(MinSlope,min(MaxSlope,Slope2));
    S=size(H);
    if min(S)==5
        1
        duds1 = ((H(3,1) + H(3,3)) .* -Slope1) ./  ((H(1,1) + H(1,3)
).* -Slope1); % the negative is due to an axis direction mismatch
        duds2 = ((H(4,2) + H(4,4)) .* -Slope2) ./  ((H(2,2) + H(2,4))
.* -Slope2); % the negative is due to an axis direction mismatch
        Pz1=(dh./(1+duds1)); %Convert Slope to distance
        Pz2=(dv./(1+duds2));
    else
        2
        Pz1=(dh*H(1)/H(2))-Slope1;
        Pz2=(dv*H(1)/H(2))+Slope2;
    end

    PzTemp=(Mag1.*Pz1+Mag2.*Pz2)./TotMag;
    PzTemp=max(MinRange,min(MaxRange,PzTemp));









    PzTemp(InvalidMask)=NaN;
%     S=size(PzTemp);
%     x=floor(S(1)/2);
%     y=floor(S(2)/2);
%     PzTemp=squeeze(PzTemp(x,y,:,:));
    PzEst=PzEst+PzTemp;
end


Pz=(PzEst./3.);
end
```

**Fill_nan routine**

```
function Filled_pic=fill_nan(spic)
s=size(spic);
% for Q=1:3
for I=2:s(1)-1
    for J=2:s(2)-1
        if isnan(spic(I,J))
            around=[spic(I-1,J-1),spic(I,J-1),spic(I+1,J-1),spic(I-
1,J),...
                        spic(I+1,J),spic(I-
1,J+1),spic(I,J+1),spic(I+1,J+1)];
            ctr=0;
            for x=1:max(size(around))
                if isnan(around(x))
                else
                    ctr=ctr+1;
                end
            end
            if ctr>=4
                average=nanmean(around);
                spic(I,J)=average;
%             elseif ctr>=1
%                 spic(I,J)=NaN;
            end
        end
    end
end
% end
Filled_pic=spic;
end
```

## Bibliography


[1] B. C. Hacker and J. M. Grimwood, *On the Shoulders Titans: A History of Project Gemini,* Washinton, DC: National Aeronautics and Space Administration, 1997.

[2] A. C. M. Allen, C. Langley, R. Mukherji, A. B. Taylor, M. Umasuthan and T. D. Barfoot, "Rendezvous lidar sensor system for termial rendezvous, capture and berthing to the international space station," *Proc. of SPIE,* vol. 6958, pp. 69580S-69580S-8, 2008.

[3] The Air Force Research Laboratory, *ANGELS Fact Sheet,* Kirtland AFB: 377 ABW/Public Affairs, 2014.

[4] T. Bryan, R. Howard, J. Johnson, J. Lee, L. Murphy and S. Spencer, "Next Generation advanced video guidance sensor," in *Aerospace Conference, 2008 IEEE*, 2008.

[5] E. H. Adelson and J. R. Bergen, *The plenoptic function and the elements of early vision,* Massachusetts Institute of Technology: Vision and Modeling Group, Media Laboratory, 1991.

[6] R. I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004.

[7] R. A. Raynor, *Rangefinding with a Plenoptic Camera,* Dept. Eng. Physics, AFIT, Dayton, OH, 2014.

[8] E. H. Adelson and J. Y. Wang, "Single lens stereo with a plenoptic camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 14, no. 2, pp. 99-106, 1992.

[9] R. Ng, *Digital light field photography,* Dept. Computer Science, Stanford Univ., Stanford, CA, 2006.

[10] R. C. Bolles, H. H. Baker and D. H. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *International Journal of Computer Vision,* vol. 1, no. 1, pp. 7-55, 1987.

[11] S. Wanner, F. Janis and B. Jahne, *Generating EPI representations of 4D light fields with a single lens focused plenoptic camera,* Springer Berlin Heidelberg, 2011.

[12] D. Antolovic, *Review of the Hough Tranform Method, With an Implementation of the Fast Hough Variant of Line Detection,* Department of Computer Science, Indiana University and IBM Corporation.

[13] The MathWorks, Inc., "Hough," The MathWorks, Inc., 2016. [Online]. Available: http://www.mathworks.com/help/images/ref/hough.html?s_tid=gn_loc_drop. [Accessed 1 10 2016].

[14] Lytro, "Lytro Support," Lytro. Inc., 2016. [Online]. Available: support.lytro.com. [Accessed 20 February 2016].

[15] F. Anwar, "Is Lytro the future of Cameras, details and samples inside," Hardware Insight, 10 September 2012. [Online]. Available: http://www.hardwareinsight.com/is-lytro-the-future-of-cameras-details-and-samples-inside/. [Accessed 10 10 2016].

[16] Raytrix, 13 August 2015. [Online]. Available: www.raytrix.de/presse/. [Accessed 20 February 2016].

[17] Raytrix, "Products," Raytrix, 2016. [Online]. Available: www.raytrix.de/produkte/. [Accessed 20 February 2016].

[18] D. Dansereau, "Light Field Toolbox v0.4," 2015. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4. [Accessed 11 August 2015].

[19] K. Marcoe, "LIDAR an Introduction and Overview," 2007. [Online]. Available: http://web.pdx.edu/~jduh/courses/Archive/geog481w07/Students/Marcoe_LiDAR.pdf. [Accessed 29 March 2016].

[20] Y. Huang, "SLAM, Visual Odometry, Structure from Motion, Multiple View Stereo," LinkedIn Corporation , 3 November 2014. [Online]. Available: http://www.slideshare.net/yuhuang/visual-slam-structure-from-motion-multiple-view-stereo. [Accessed 10 10 2016].

[21] M. Fair, "Introduction to Structure From Motion and its Applications in Remote Sensing," Los Alamos Naitonal Laboratory, Los Alamos, 2010.

[22] K. P. Bechis and A. M. Pitruzzello, "Research into a Single-Aperture Light Field Camera System to Optain Passive Ground-based 3D Imagery of LEO Objects," Northrup Gruman Information Systems.

[23] D. Dansereau and L. Burton, *Gradient-Based Depth Estimation From 4D Light Fields,* Dept. of Electrical and Computer Engineering, University of Calgary, Alberta, Canada.

[24] D. G. Dansereau, O. Pizarro and S. B. Williams, *Decoding, Calibration and Rectification for Lenselet-Based Plenoptic Cameras,* Australian Centre for Field Robotics; School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, NSW, Australia.

[25] S. Wanner, S. Meister and B. Goldluecke, "Datasets and Benchmarks for Densely Sampled 4D Light Fields," *Vision, Modelin, and Visualization,* 2013.

[26] T. Georgiev, A. Lumsdaine and S. Goma, "Plenoptic Principle Planes," in *Imaging and Applied Optics,* Toronto, 2011.

[27] D. G. Dansereau, *Plenoptic Signal Processing for Robust Vision in Vield Robotics,* Australian Center for Field Robotics, School of Aerospace, Mechanical and Mechatronic Eng.,University of Sydney, Sydney, Austrailia, 2014.

[28] *International Docking System Standard (IDSS) Interface Definition Document (IDD).*

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 074-0188*



| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From – To)* |
|---|---|---|
| 22-12-2016 | Master's Thesis | September 2014 – December 2016 |

**TITLE AND SUBTITLE**

Using a Plenoptic Camera for Real-Time Depth Estimation

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

Anderson, Ryan J., 1st Lieutenant, USAF

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)**
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENY)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

**8. PERFORMING ORGANIZATION REPORT NUMBER**

AFIT-ENY-MS-17-M-002

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

(no sponsor enter: Intentionally left blank)

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

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

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

**14. ABSTRACT**

The plenoptic camera collects samples of the 4D light field, which allows for the collection of imagery and depth information simultaneously. The plenoptic camera differs from stereoscopic systems because the light field is captured by a single lens and sensor rather than two or more. This translates to less size, weight, and power (SWAP) which is ideal for space missions where imagery and depth information is needed such as proximity operations and docking missions. The main objective of this research is to design and evaluate performance of a method to autonomously output the depth of key elements of a scene in real-time. In this research, the depth is estimated using a gradient method and the key elements of the scene are selected using the Hough transform. A major finding of this research is that in order for this to run near real-time, only a small portion of the light field can be analyzed due to size of the data set. This results in the potential to miss important information that the light field has to offer. The average error of the Lytro Illum was ~7% while the First Generation's was ~17% error with a decrease in accuracy as the range increases. The average run time for the Illum and First Generation was approximately five seconds and three seconds respectively using the Hough transform to reduce the size of the light fields. The Hough transform is the most significant portion of the run time, but it still reduced the run time by than it increased it. This work lays the groundwork for using a plenoptic camera to autonomously output the depth information about a scene in real-time by developing a depth estimation method for specific features in light fields and concluding that the Hough transform is a good method for this, especially if multiple features are desired.

**15. SUBJECT TERMS**
Plenoptic Camera, Range Finding, Light Field, Passive, Monocular, Depth,

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Dr. Richard Cobb, AFIT/ENY |
| U | U | U | UU | 88 | 19b. TELEPHONE NUMBER *(Include area code)* (937) 255-6565, ext xxxx (NOT DSN) (richard.cobb@afit.edu) |



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