Single Chip LIDAR with Discrete Beam Steering by Digital Micromirror Device

SINGLE CHIP LIDAR WITH DISCRETE BEAM STEERING BY DIGITAL MICROMIRROR DEVICE

by

Braden Smith

A Thesis Submitted to the Faculty of the

COLLEGE OF OPTICAL SCIENCES

In Partial Fulfillment of the Requirements

For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

2017

STATEMENT BY AUTHOR

The thesis titled Single Chip LIDAR with Discrete Beam Steering by Digital Micromirror Device prepared by Braden Smith has been submitted in partial fulfillment of requirements for a master's degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that an accurate acknowledgement of the source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Braden Smith

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

____________________________________	*May 10, 2017*
Yuzuru Takashima Associate Professor of Optical Sciences	Date

ACKNOWLEDGEMENTS

I would like to acknowledge everyone who helped and supported me through my master’s studies at the University of Arizona. The support, encouragement, and guidance that everybody has given me throughout the two years of my master’s studies has had a huge impact on me and my work.

First, I would like to thank my advisor Dr. Takashima of the University of Arizona. He has been a great professor and mentor throughout my undergrad and graduate careers. Always more than willing to answer any question I had, Dr. Takashima has always been a great resource. During my master’s research project, Dr. Takashima’s experience and guidance were invaluable in creating a quality thesis, publication, and patent.

I would also like to acknowledge my colleagues that worked alongside me in Dr. Takashima’s lab that helped in the development of this project. A special thanks goes to Brandon Hellman, Adley Gin, and Alonzo Espinoza who, together, developed circuits, mechanics, and software necessary for the project to continue. All other members of the lab, whether directly involved with my project or not, provided support and were valuable resources throughout my project.

Lastly, I would like to thank all my friends and family that gave me support during my master’s studies. My friends and family gave me much encouragement during tough times working on my project. My project would not have been possible without them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A novel method of beam steering that utilizes a mass-produced Digital Micromirror Device (DMD) enables a large field of view and reliable single chip Light Detection and Ranging (LIDAR). Using a short pulsed laser, the micromirrors' rotation is frozen mid-transition which forms a programmable blazed grating which efficiently redistributes the light to a single diffraction order, among several. With a nanosecond 905nm laser and Si avalanche photo diode, measurement accuracy of < 1 cm for 3340 points/sec is demonstrated over a 1 m distance range and with a 48° full field of view.

CHAPTER 1: Introduction

1.1 Why Beam Steering

Laser beam steering technology is essential for Light Detection and Ranging (LIDAR) systems. For this reason, beam steering technologies have been actively researched. Along with mechanical and completely non-mechanical beam steering, Micro-Electro-Mechanical-Systems (MEMS) are one of the emerging beam steering fields that are especially suitable for LIDAR.

1.2 State of the Art Beam Steering

Mechanical scanning including gimbals, fast-steering mirrors, Risley prisms, rotating polygon mirrors and gratings have been used for wide wavelength ranges [1]. Although mechanical beam scanning modalities are widely adopted, fewer or no moving parts and smaller component inertia is more desirable to reduce SWaP, (size, weight, and power) while keeping scan speeds high [2, 3]. These qualities are especially required for autonomous vehicle and robotics applications.

In contrast, completely non-mechanical scanning such as programmable spatial light modulators, modulo 2π optical phased arrays, solid state phase arrays, and liquid crystal electro-optic scanners are emerging [1, 4-7]. These non-moving part devices enable large steering angles and are expected to be highly reliable, and are now actively researched.

In terms of small component inertia, Micro-Electro-Mechanical Systems (MEMS) are promising too thanks to their small size and weight, low production cost, high energy efficiency, and applicability to wide wavelength ranges. These MEMS devices include single resonant mirrors and shifting lenslet arrays [2, 8, 9]. However, for autonomous vehicle LIDAR applications, a large steering angle as well as a large beam size is needed to cover a large angle of scanning and minimize beam divergence due to diffraction. Unfortunately, resonant mirrors and shifting lenslet arrays are limited in angular range and maximum accommodated beam size. Current high-end resonant mirror MEMS scanning systems have moderate fields of view at 36° and scan rates of 21 kHz [2, 10]. However, a resonant mirror's maximum beam diameter is only increased at the expense of decreasing the maximum scan rate [9]. An optical amplification of the steering angle by an inverse telescope design has been reported; however, it requires a reduced beam diameter to conserve the Lagrange invariant which would limit the effective delivery of light over large distances due to beam spreading by diffraction [11, 12].

1.3 Proposed Novel Beam Steering Method

Thus, a beam steering system for use in LIDAR would ideally have a large beam size, a wide field of view, and a high scan rate while minimizing the number of moving parts. To simultaneously satisfy these requirements, we propose and demonstrate a new beam steering method by the commercially available Digital Micromirror Device (DMD) with a short pulsed laser. We also demonstrate its application to LIDAR with a large field of view, high scan rate and potentially a large beam size.

In chapter 2, the key idea of freezing the micromirror movement by a nanosecond laser pulse to form a programmable blazed grating is discussed, followed by a diffraction analysis of this programmable grating. The beam steering theory is experimentally demonstrated for three kinds of sub microsecond pulsed light sources: a collimated laser beam, a focused laser beam, and a quasi-collimated beam from a light emitting diode. In chapter 3, implementation of the beam steering method in a single chip DMD LIDAR system is discussed with experimental results of distance measurement accuracy and live image capturing. Finally, we will address possible optical design solutions to overcome the limited number of scanning points available to meet modern requirements for LIDAR systems.

CHAPTER 2: Discrete and Continuous Beam Steering by DMD

2.1 DMD Theory of Operation

In Fig. 1, the DMD is schematically depicted. This beam steering setup utilizes a 608x684 (horizontal by vertical) DMD chip (DLP3000, Texas Instruments). The micromirrors are positioned in a diamond configuration with a corner to corner period of 10.8µm as shown in Fig. 1 (a). On this DMD an array of micromirrors flip between an “on” and “off” state by rotating +/- 12° about an axis defined by the diagonal of the mirror. Thus, a DMD is designed for binary spatial light modulation and is not intended to be used for angular beam steering, unless additional optics to convert the spatial modulation to angular modulation are incorporated at the expense of light throughput [13].

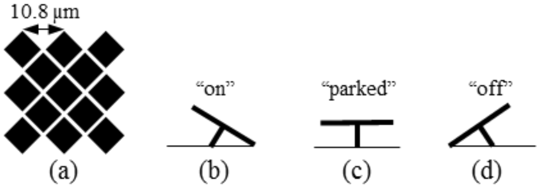


Fig. 1. Representation of the (a) DMD diamond pixel layout (Top View); (b) a mirror in the “on” position at +12°; (c) a mirror in the “parked” position at 0° when the DMD is powered down; (d) a mirror in the “off” position at -12° (Side View).

The DMD mirrors move continuously between the “on” or “off” states with a typical transition time on the order of micro seconds [14]. This unused transitional state of the DMD is utilized by a short pulsed laser whose pulse duration is much shorter than the transition time of the mirrors. With the short pulsed laser, the micromirror movement is “frozen” at an angle between the stationary “on” and “off” states. Thus, it is feasible to form a programmable blazed diffraction grating to discretely steer a laser beam with a collimated beam. It is also feasible to create a continuously scanned and diverging beam if the laser beam is focused on a single DMD mirror to remove the diffraction grating effects.

2.2 Numerical Analysis of Beam Steering of a Plane Wave

The principle of the beam steering for plane wave illumination is first numerically modeled. Each mirror was modeled as a series of point sources with an associated phase and optical path length (OPL) induced by the tilt of mirror while taking into account the angle of incidence of the incoming beam. Fig. 2 shows the OPL of a 5 x 5 mirror area with the micromirrors tilted at -12 degrees and a plane wave incident angle of +3 degrees. The field contributions from each point source are added together by the Huygen-Fresnel integral to calculate the electric field on the observation screen located 250 mm away from the DMD [15].

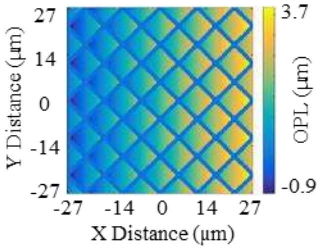


Fig. 2. Modeled OPL profile of a 5x5 mirror pattern. In this instance, the incidence angle is 3º and the micromirrors are fixed at -12º.

The results from a model containing a 70x3 (horizontal by vertical) mirror array is depicted in Fig. 3. Since only the horizontal locations of the diffraction orders are of interest in this model, a large number of mirrors was modeled in the horizontal direction to better simulate the actual DMD. To keep the amount of required processing low, a minimal number of mirrors were modeled in the vertical direction. The top five rows in Fig. 3 show “snap shots” of the diffraction intensity patterns normalized to the 0th order of diffraction. The very bottom row of Fig. 3 shows a “long exposure” shot. The diffraction orders from left to right model micromirror rotation angles of -10.3, -5.8, 0, +4.9, and +12 degrees, respectively. The incident plane wave is diffracted into one of the specific diffraction orders with relative diffraction efficiencies close to 100% since the frozen state of the tilted DMD mirrors is equivalent to a blazed grating where the slope of the mirror is set to the blaze angle. The diffraction pattern is further evaluated by modeling the reflectivity of the DMD’s cover glass, 4% with an optimized antireflective coating and 23% without, as described in detail in section 4.3.

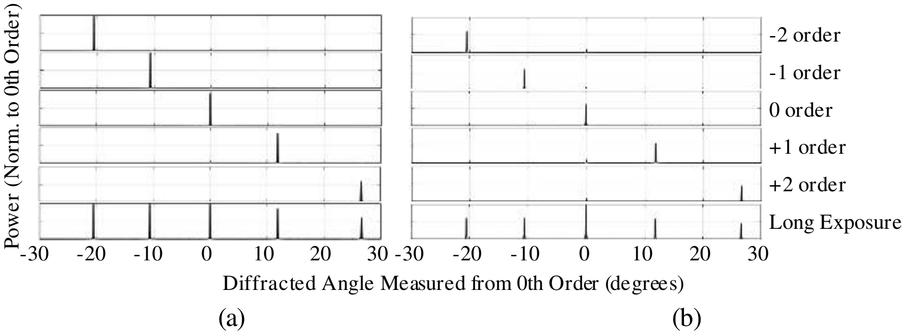


Fig. 3. Results of MATLAB simulation of scanning with a 30º angle of incidence plane wave with wavelength of 905nm. Snapshots of a scan as well as a long exposure of the entire scan are shown. (a) cover glass reflectivity modeled as 4%. (b) cover glass reflectivity modeled as 23%.

2.3 Demonstration of Beam Steering of a Plane Wave

Experimental setup for the beam steering of a plane wave is illustrated in Fig. 4. The source is an 8ns, 905nm laser diode (LS9-220-8-S10-00, Laser Components, Germany). The laser pulse is collimated by a 20x and NA 0.4 microscope objective lens (80.3071, Rolyn Optics) and directed toward the DMD surface at an incident angle of 30 degrees. The implementation of an optional focusing lens to focus light onto a single DMD pixel is discussed further in section 2.5.

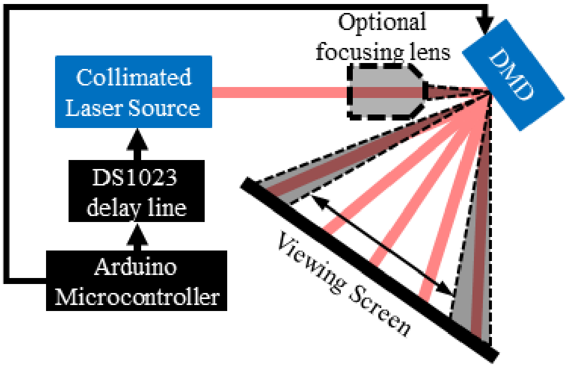


Fig. 4. Experimental component setup for beam steering

The DMD driver contains an external trigger port that was used to switch the mirror array between the “on” and “off” state by displaying an all-white or all-black bitmap image. Both the DMD driver and the pulsed laser source were controlled by a microcontroller (Arduino Uno, Arduino) by sending delayed trigger signals to synchronize the laser pulse with the movement of the micromirrors.

We experimentally determined that the micromirrors start transitioning about 218μs after the external trigger pulse is sent to the DMD driver and take about 2μs to complete transitioning. The complete timing diagram is depicted in Fig. 5. To increase timing precision, we added a serially programmable timing element (DS1023, Dallas Semiconductor) between the Arduino and laser source which adds an additional delay with 0.25ns timing precision. This allows the Arduino micro controller to produce virtually any time delay between triggering the DMD and laser with 0.25ns precision.

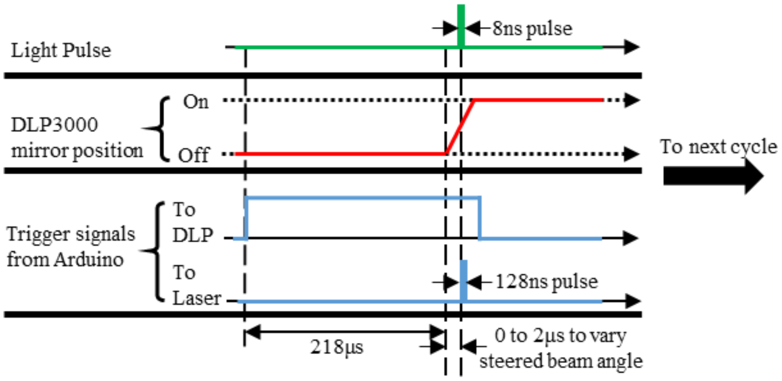


Fig. 5. Timing diagram of beam steering method using Arduino microcontroller.

Fig. 6 shows a captured image of the progression of a horizontal scan, showing the five discrete diffraction orders. This image was captured with a CMOS camera (DCC1545M-GL, Thorlabs) and an infrared lens (12VM1040ASIR, Tamron) by imaging the viewing screen placed 250 mm away from DMD as illustrated in Fig. 4.

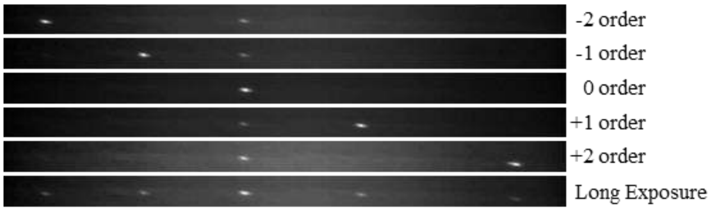


Fig. 6. Progression of a scan across the five discrete diffraction orders using a collimated 8ns 905nm laser source. The upper five images are “snapshots” of the system as it scans from the -2 to +2 diffraction orders, and the bottom image is a “long exposure” of the entire scan (see Visualization 1: https://arizona.box.com/s/wagfce3he1dd8105fbqpoanckdjhqcek).

2.4 Demonstration of Beam Steering with Light Emitting Diode Illumination

Since diffraction dominates the performance of DMD beam steering, quasi monochromatic and incoherent light sources are also usable. The laser source in Fig. 4 was replaced with a green LED (L-7113GT, Kingbright) and modulated and synchronized to the movement of the DMD mirrors. Fig. 7 shows a long exposure picture while the beam is scanned over the five diffraction orders. The LED used was not collimated to the degree of the laser used in plane wave illumination, causing larger spot sizes. As seen in Fig. 7, the scanning pattern contains a horizontal row of spots as well as spots above and below. These occur because of the micromirror pattern is a 2-dimensional pattern, creating a tilted grid-like diffraction pattern. While this grid-like pattern occurs for the 905nm laser scanning as well, it is much less noticeable because of the larger spacing between diffraction orders and smaller spot sizes.

It is important to note that DMD beam steering of LEDs is not feasible for applications that require a high optical power. LEDs generally have a lower maximum power output than pulsed lasers, meaning that the energy per pulse is generally much lower for LEDs. Additionally, LEDs cannot be switched on and off as fast as pulsed lasers, and a short pulse time is crucial for DMD beam steering. To capture the images shown in Fig. 7, a very long integration time was needed when compared to pulsed laser scanning. However, even with the reduced optical power of the LED, beam steering by LED is still shown to be feasible.

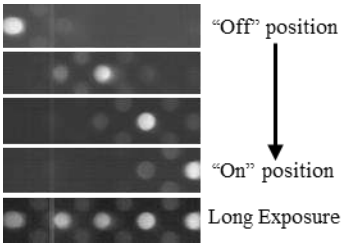


Fig. 7. Progression of discrete beam steering using a quasi-collimated LED. The upper four images are "snapshots" of the scan progression and the lower image is a "long exposure" of the scan (see Visualization 2: https://arizona.box.com/s/wagfce3he1dd8105fbqpoanckdjhqcek).

2.5 Demonstration of Beam Steering with Focused Laser Illumination

With a collimated laser or quasi collimated LED illumination, continuous scanning is not possible due to the diffraction effects of the relatively small DMD pixels. As an opposite case, we illuminated a single DMD pixel with a nano-second 532 nm laser (Vector 532-1000-20, Coherent). The laser was focused by a microscope objective (20x, Swift Optical Instruments) and controlled with the same synchronizing electronics. In this way, the diffraction effects no longer dominated the scan pattern. A CMOS camera (DCC1545M-GL, Thorlabs) and lens (12VM1040ASIR, Tamron) imaged the viewing screen, shown in Fig. 8.

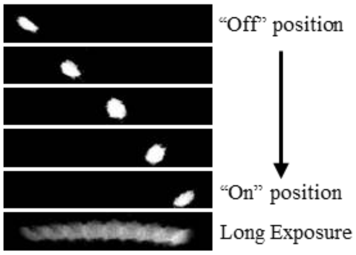


Fig. 8. Progression of a scan using a laser beam focused onto a single DMD pixel. The upper five tiles are "snapshots" of the steered beam at five locations across the scan. The lower tile shows a "long exposure" of the entire scan (see Visualization 3: https://arizona.box.com/s/wagfce3he1dd8105fbqpoanckdjhqcek).

CHAPTER 3: Integration into LIDAR System

3.1 Motivation for LIDAR system

This method of beam steering requires pulsed beams, thus it is well suited for LIDAR systems based on a Time of Flight (TOF) measurement. To demonstrate this application of DMD beam steering, we integrated DMD based beam steering into a 1D line scanning LIDAR system. This LIDAR system makes TOF measurements along each of five diffraction orders within the DMD's field of view of 48 degrees.

3.2 Physical LIDAR setup

To make TOF measurements, an avalanche photodiode (APD) (C12702, Hamamatsu) and a fold mirror were added to the optical setup as illustrated in the schematic in Fig. 9 (a). A 3D printed mount to reduce cross talk between transmitting and receiving optical passes was also added as illustrated in Fig. 9 (b). As illustrated in Fig. 9 (a), the laser pulse travels from the collimating objective through an adjustable aperture and is directed by a fold mirror onto the DMD at a 30º incident angle. The reflected light retraces this path through the DMD to the APD. The entire physical optical setup of the LIDAR system is shown in the photograph in Fig. 10.

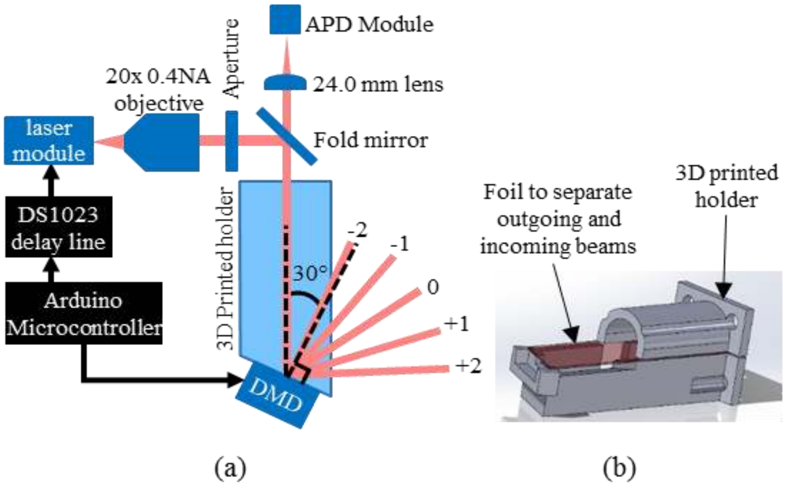


Fig. 9. Illustrations of (a) the optical setup used in 1D linescan LIDAR system and (b) the optical isolation scheme.

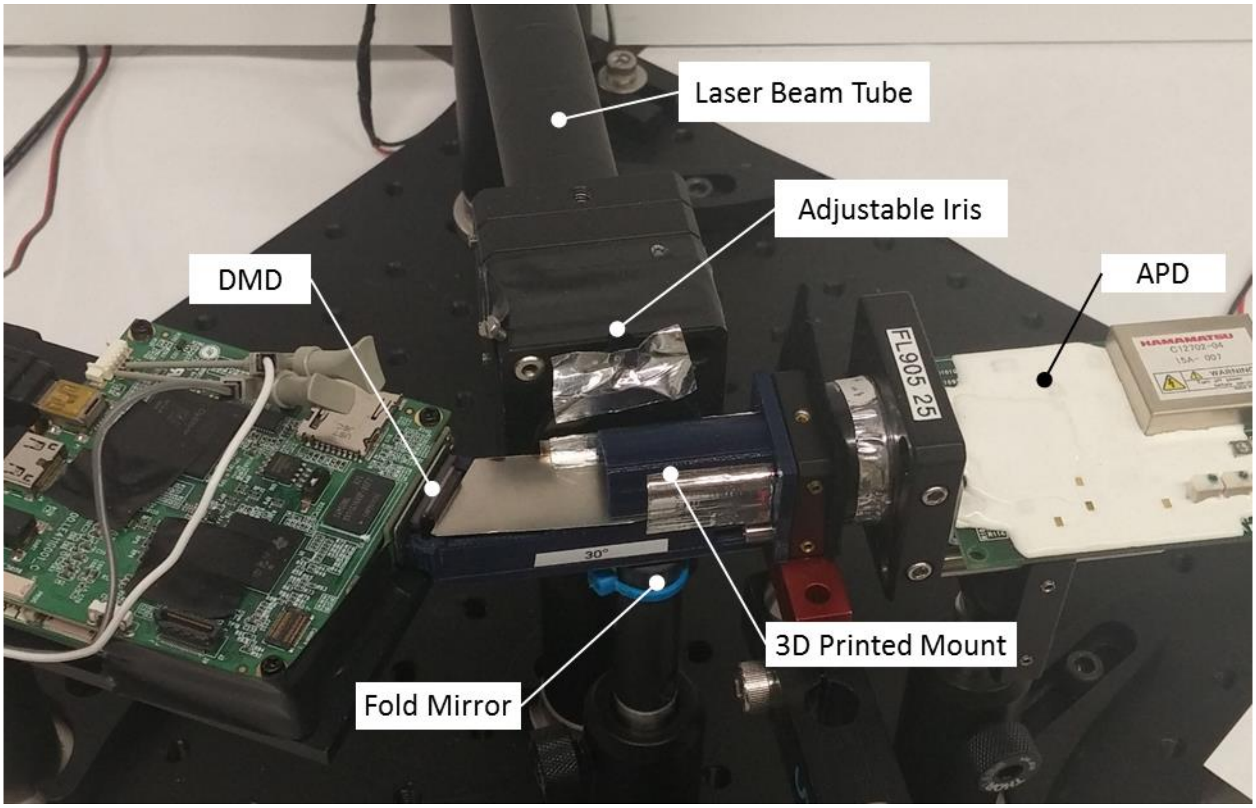


Fig. 10. Photograph of complete optical LIDAR setup.

The 3D printed mount held the APD, fold mirror, and DMD in such a way that allowed the outgoing pulses to be spatially isolated from the APD, but still allows incoming pulses to be detected. As can be seen in Fig. 11, the mount is made up of two plastic pieces that sandwich a piece of thin metal foil separating the DMD surface into an upper and a lower half. The outgoing pulse is only incident on the lower half of the DMD while the APD only detects the part of the return pulse that is incident on the upper half of the DMD. Both the upper and lower half of the DMD are set to steer in the same direction so that the incoming pulse is steered directly back into the system.

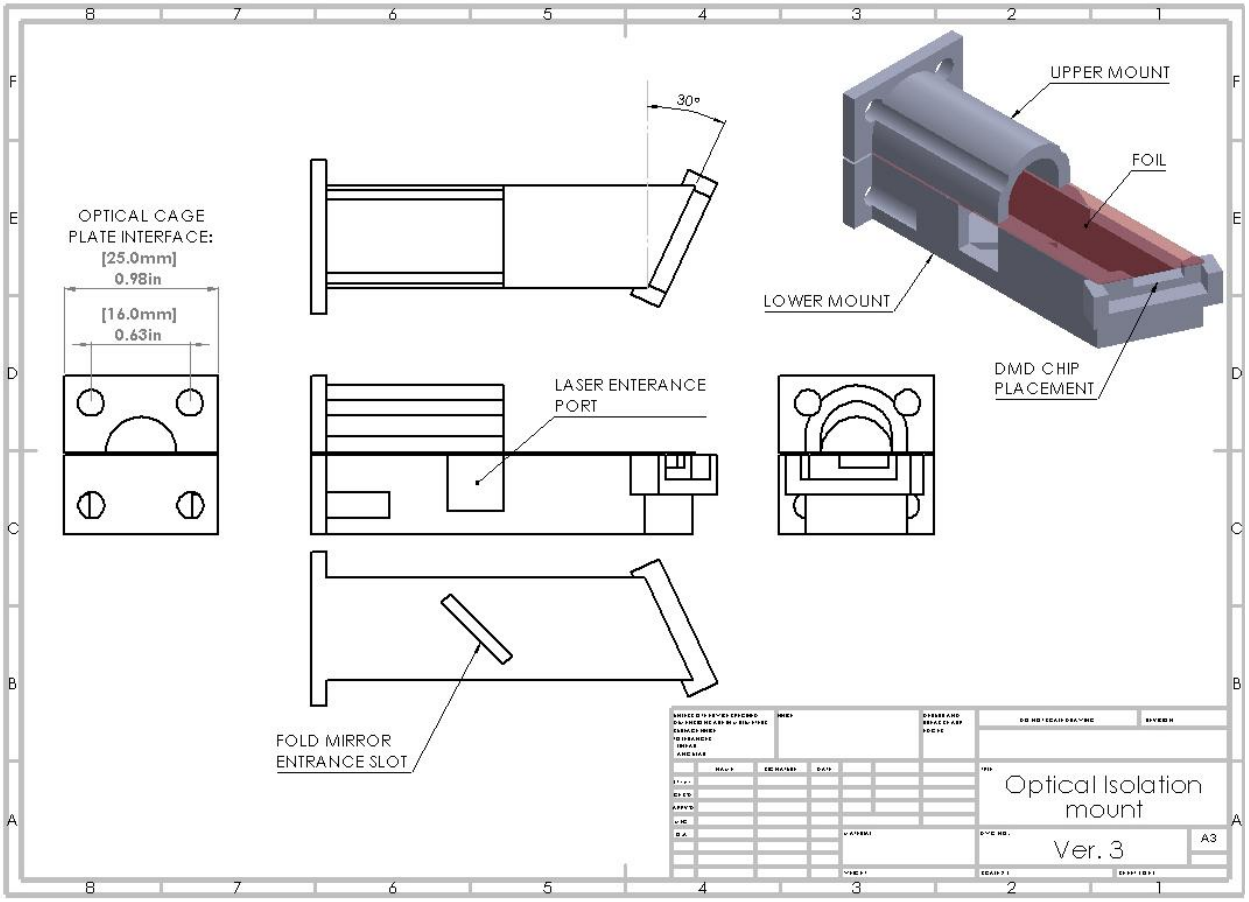


Fig. 11. Drawing of 3D printed mount (designed in SolidWorks)

Fig. 12 shows a block diagram of the electronic circuit for the TOF measurements. A time to digital converter (TDC7200, Texas Instruments) was used to measure the TOF of each pulse. The rising edge of the Arduino's trigger pulse to the laser module was the "start" signal and the rising edge of the APD's electrical response was the "stop" signal.

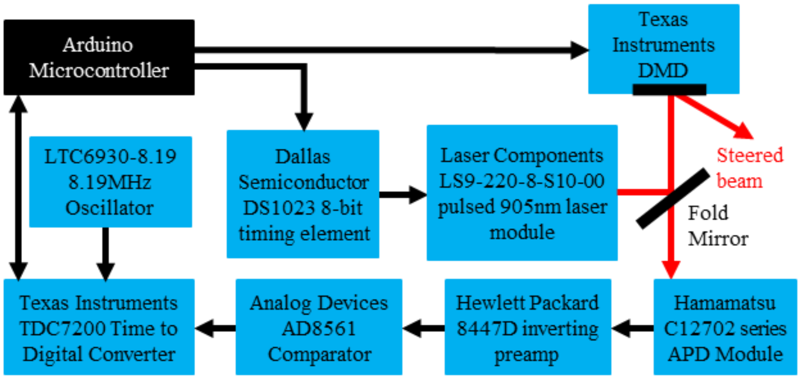

Arduino Microcontroller
Texas Instruments DMD
LTC6930-8.19 8.19MHz Oscillator
Dallas Semiconductor DS1023 8-bit timing element
Laser Components LS9-220-8-S10-00 pulsed 905nm laser module
Steered beam
Fold Mirror
Texas Instruments TDC7200 Time to Digital Converter
Analog Devices AD8561 Comparator
Hewlett Packard 8447D inverting preamp
Hamamatsu C12702 series APD Module

Fig. 12. Block diagram of electrical components used in TOF circuitry.

After a TOF measurement is made, the data is retrieved through a serial interface and transferred to a host computer. The data is sent through the Arduino's built in serial monitor for real time data collection. Alternatively, for off-line and faster data collection, the data is sent to a static random access memory (SRAM) chip (23LC1024, Microchip) via serial peripheral interface (SPI). The SPI interface was used because much higher data transfer rates could be achieved compared to the Arduino's serial monitor. The Arduino serial monitor allowed data points to be read at a 433Hz rate whereas saving data points to the SRAM via the SPI interface allowed data points to be saved at a speed of 3.34kHz.

CHAPTER 4: Experimental Results

4.1 Motivation

Several different tests were performed that were designed to test the abilities of the LIDAR system. The measurement range over the five diffraction orders as well as the accuracy were of great interest. To complete these tests, paper targets were constructed of varying sizes and reflectivities. Brown paper was used for the low reflectivity targets and white paper was used for the high reflectivity targets. These targets were made in three sizes: small, medium, and large. The small targets were 25mm x 25mm, medium targets were 50mm x 50mm, and large targets were 100mm x 100mm. These targets were then placed at varying distances within the field of view of the LIDAR system. The real distance at which the targets were placed was compared to the reported distances from the LIDAR system. Multiple configurations involving these targets are discussed below.

4.2 Diffraction Efficiency Test

The diffraction efficiencies and the angles of diffraction for all five horizontal diffraction orders are tabulated in Table 1. The angles of diffraction were measured with respect to the 0th order in the horizontal plane. Positive angles were measured in a counterclockwise direction. Table 1 contains both the measured and predicted diffraction angles.

The transmitted power was measured by aiming the diffracted beams onto the sensor of a CMOS camera (DCC1545M-GL, Thorlabs). The mean counts of each captured image were proportional to the incident power. Table 1 provides the diffraction efficiencies normalized to the 0th order diffraction power for each diffraction order as well as the absolute diffraction efficiency. The absolute diffraction efficiency was measured with respect to the light incident on the DMD surface. Factors such as DMD mirror reflectivity and micromirror fill factor reduce the absolute efficiency.

Table 1. Diffraction Angle Test Results

Diffraction Order	Diffraction Angles		Normalized Power		Absolute Efficiency
	Model	Measured	25cm Range	50cmRange	
-2	20.5°	20.5°	0.67	0.64	41%
-1	10.6°	11.0°	0.69	0.63	42%
0	0°	0°	1.00	1.00	63%
+1	-11.9°	-11.5°	0.55	0.50	33%
+2	-26.6°	-27.5°	0.40	0.38	25%

The data from this test supports the MATLAB model discussed in section 2.2. As depicted in Fig. 13, the measured and predicted normalized diffraction efficiencies are plotted together for each of the five diffraction orders. The fact that the 25cm measurements do not differ greatly from the 50cm measurements suggest that each of the scan angles is well collimated.

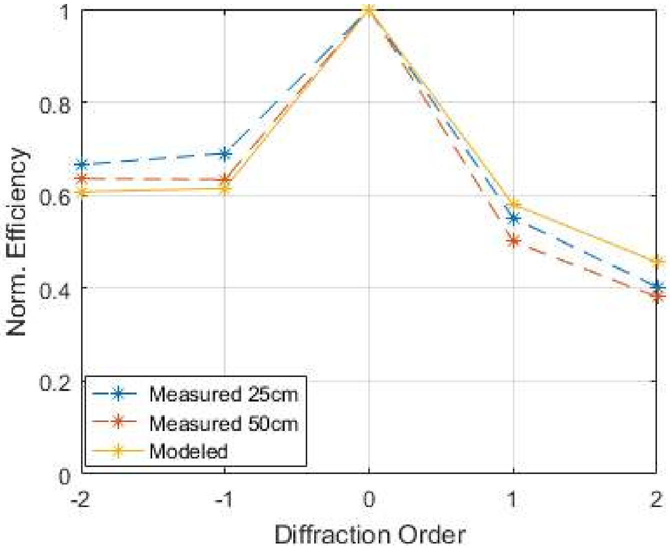


Fig. 13. Plot comparing measured and predicted normalized diffraction efficiencies.

4.3 Maximum Measurable Distance Test

The maximum measurable distance test was performed to quantify the system's distance measurement accuracy. To do this, we performed a linear conversion from the reported digital number to reported distance using a linear least squares best fit line. Small (25mm by 25mm) and high reflectivity (white) paper targets were placed at different distances to calibrate each arm. Varying numbers of measurements were averaged together to show the effects of averaging on the accuracy of the reported measurements.

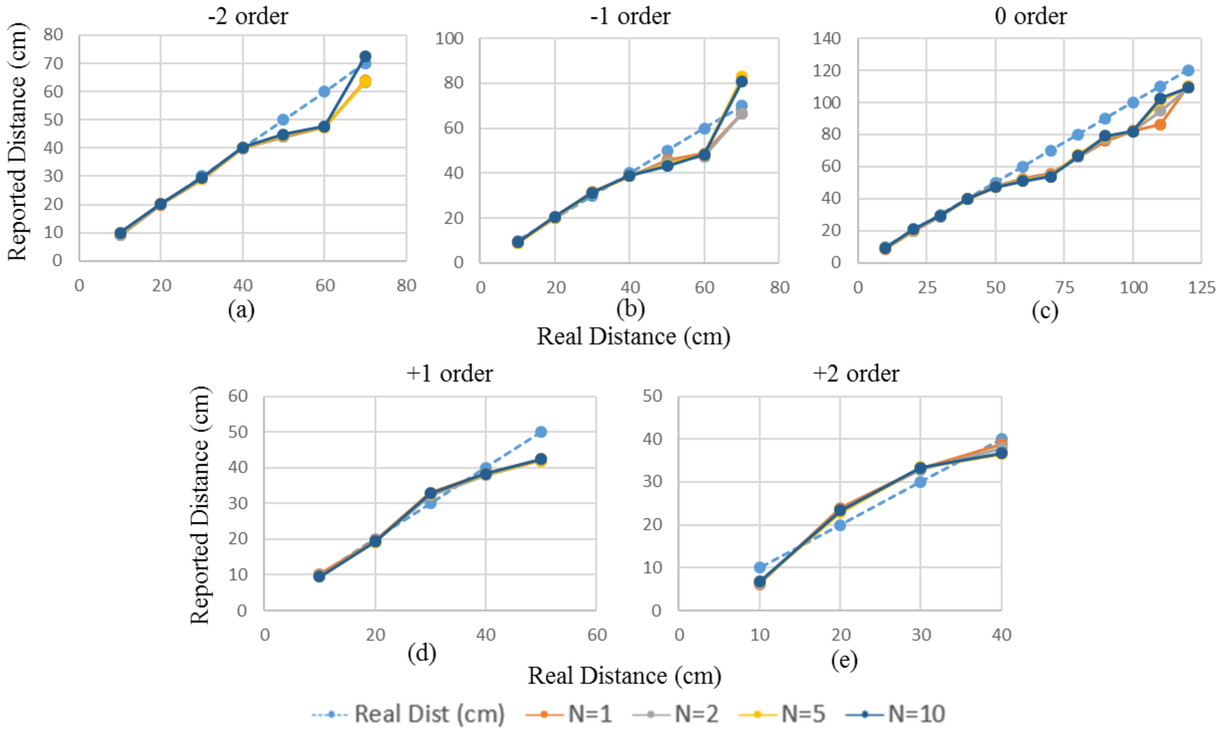


Fig. 14. Distance reporting capabilities of each of the five diffraction orders for N = 1 through N = 10 averages.

As shown in Fig. 14, the -2 through +1 diffraction orders were able to provide accurate distance measurements from 10cm to 50cm. For distances of 50cm onward, the reported distances tended to deviate from the real distances. The +2 diffraction order was only able to detect objects up to 40cmdue to its low diffraction efficiency, see Table 1. Due to the large tilt angle of the micromirrors and the 30º beam incidence angle to the DMD, the projected collection area of the DMD is smallest for the +2 order, as can be seen in Fig. 9.

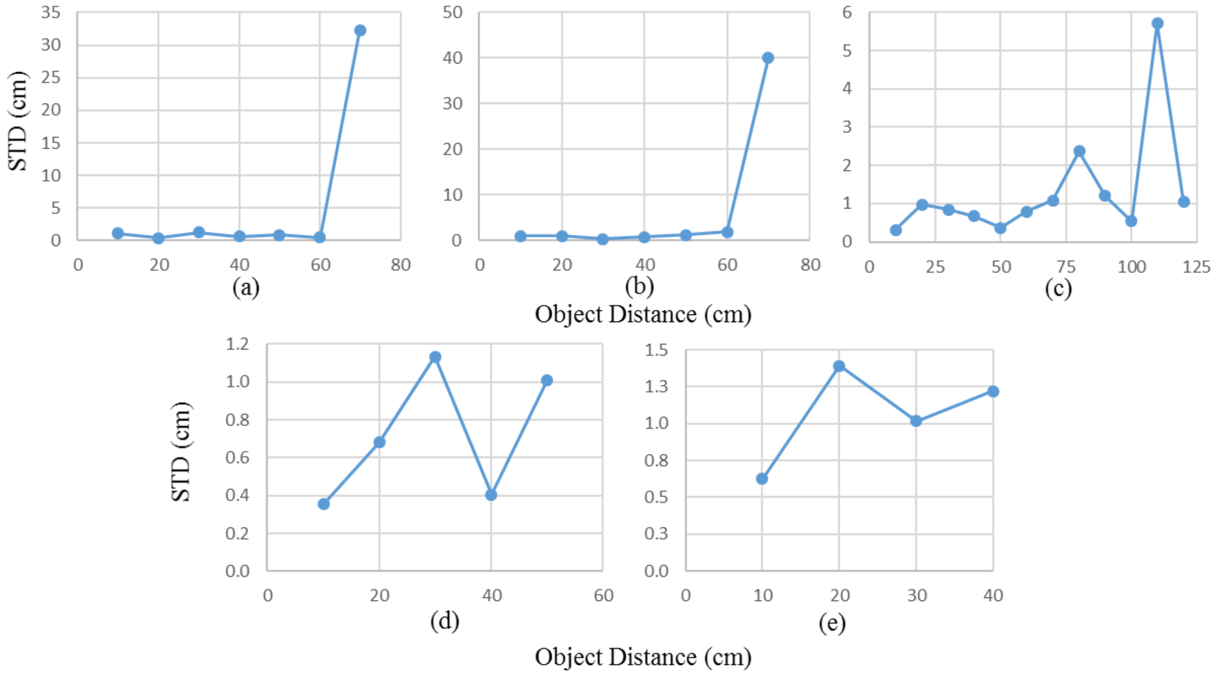


Fig. 15. Standard deviation as a function of distance for the (a) -2 order, (b) -1 order, (c) 0 order, (d) +1 order, and (e) +2 order.

To quantify the repeatability of the measurements, N data collects were averaged together to produce one measurement and N was varied to 1, 2, 5, and 10. The standard deviation (STD) of reported measurements as a function of distance is shown in Fig. 15 for each diffraction order. As shown, for all distances from 10 to 50cm, the reported STD is less than 1.5cm, and many of the reported STD values are less than 1 cm. This confirms that accuracy less than 1 cm is feasible for this system. The 10cm to 50cm distance measurements for N = 1 and N = 10 averaging of each of the five diffraction orders are compared together in Fig. 16. The shapes of the curves in Fig. 16(a) are very similar to the curves in Fig. 16(b), suggesting that N = 1 averaging provides sufficient accuracy.

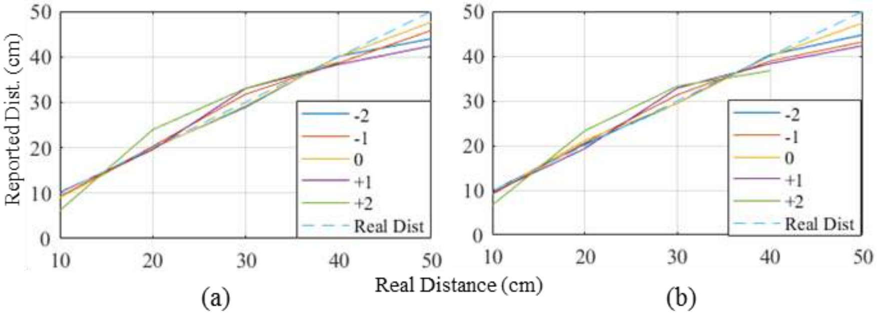


Fig. 16. Plots of the measured distances are shown for averages of (a) N = 1 and (b) N = 10.

The data summarizing the distance accuracy capabilities of the system from 10cm to 50 cm is tabulated in Table 2. This data includes the standard deviation of 50 cm measurements and the root-mean-square (RMS) errors for N=1, N=2, N=5, and N=10 averaging for object distances from 10 cm to 50 cm. This data suggests that a linear conversion from reported distance to real distance is not ideal, but perhaps a higher degree polynomial or lookup-table based conversion would yield higher accuracies.

Table 2. Measurement Repeatability

Diffraction Order	STD	RMS (N = 1)	RMS (N = 2)	RMS (N = 5)	RMS (N = 10)
-2	0.86	1.2	1.2	1.1	1.1
-1	1.25	1.0	1.1	1.3	1.4
0	0.37	0.5	0.5	0.6	0.6
+1	1.01	1.7	1.7	1.8	1.7
+2	1.22	1.6	1.5	1.6	1.6

Both the STD and RMS values are in units of centimeters. The reported STD shows the degree of noise present in each diffraction order when detecting an object at 50cm. The RMS errors quantifies the measurement error for different amounts of averaging.

4.4 Extended 0th Order Range

The 0th order is a special case in that the Fresnel reflection from the cover window provides more optical power than the other diffraction orders, see Table 1. Thus, the distance measurement errors are generally lower for the 0th order, as shown in Table 2. The DLP3000 DMD contains a Corning Eagle XG glass protective window with an antireflective coating optimized for visible light [16]. When using infrared light, this coating creates a strong specular Fresnel reflection that overlaps with the 0th order of diffraction. We experimentally measured this Fresnel reflection to be about 20% with a 30° incident angle. This extra power gave the LIDAR system a much longer range and a lower RMS error in the 0th order. The range of the 0th order was extended to 120cm as shown in Fig. 17.

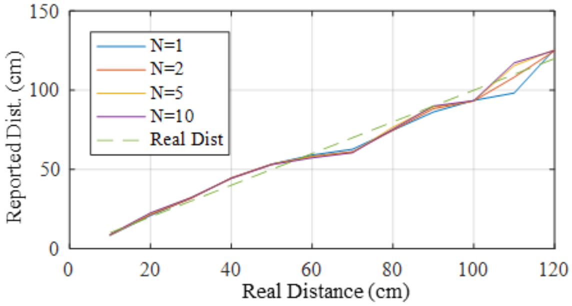


Fig. 17. Performance of the LIDAR system when making measurements in the 0th order. The range of this order is larger than the other orders because of the higher optical power in this order.

4.5 Object Motion Capture Test

The object motion capture test was performed to illustrate the capabilities of the LIDAR system in measuring the distance of moving objects in real time. The LIDAR system captured live footage of a pendulum swinging within its field of view. The pendulum consisted of a white paper cube with a side length of 25mm. This cube was suspended from a thin wire about six feet long. The live video was captured with a digital camera (PowerShot D30, Canon). The LIDAR output and the live video were then combined using MATLAB. A representation of the videos consisting of a series of snapshots is shown in Fig. 18 to illustrate the performance of the LIDAR system in real time. The system was successfully able to capture the pendulum swinging in elliptical paths as well as linear paths. Five linear paths along the direction of each of the five diffraction orders are represented in Fig. 18.

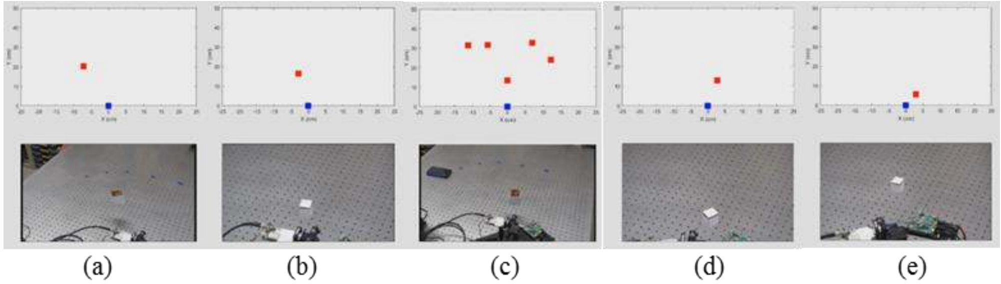
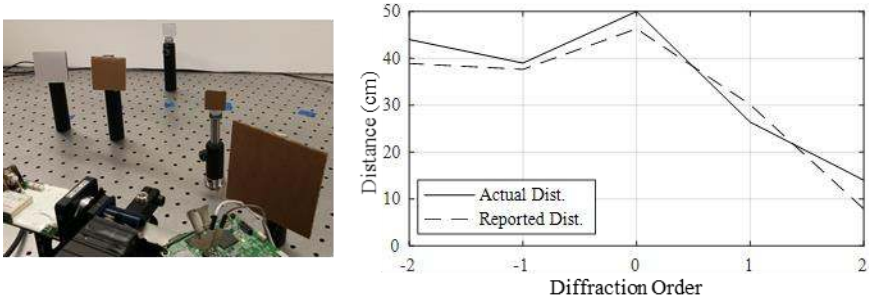


Fig. 18. Representation of a captured movie of the LIDAR system capturing swinging pendulums placed in each of the five scanning diffraction orders. (a) through (e) correspond to -2 through +2 diffraction orders respectively (see Visualizations 4-8: https://arizona.box.com/s/wagfce3he1dd8105fbqpoanckdjhqcek).

Note that in Fig. 18c artifacts appear in the -2, -1, +1, and +2 diffraction orders when the object swings through the 0th order. It was experimentally determined that these artifacts appear when the pendulum is closer than about 40 cm to the LIDAR system, as shown in Fig. 20. This implies the presence of crosstalk between the 0th order and all other orders which is explained further in section 5.

4.6 Random Object Distance Measurement Test

The random object distance measurement test quantified the ability of the LIDAR system to measure objects with varying reflectivities, sizes, and distances. High and low reflectivity targets (white paper and brown cardboard respectively) were used as targets. Small (25mm x 25mm), medium (50mm x 50mm), and large (100mm x 100mm) sized targets were randomly placed in the system's field of view. One such measurement is shown in Fig. 19. From left to right the targets used were medium size high reflectivity, medium size low reflectivity, small size high reflectivity, small size low reflectivity, and large size low reflectivity.



(a) (b)

Fig. 19. Illustration of a sample random object distance measurement consisting of (a) the object arrangement and (b) the reported results.

4.7 Quantifying Cross-talk between Diffraction Orders

A specialized Random Object Distance Test was performed to specifically investigate the degree of cross talk between the 0th order and the four other orders. In this setup, five stationary test targets were placed in the -2, -1, +1, and +2 diffraction orders at 44cm, 39cm, 26cm, and 14cm respectively. The object in the 0th diffraction order was then moved from 60cm to 10cm. The resulting reported distances as a function of the position of the 0th order object are shown in Fig. 20. As shown, the reported distances of the objects in the -2, -1, +1, and +2 orders are unaffected when the 0th order object is at 40cm or greater. However, when the 0th order object is closer than 40cm, the other four orders are affected. As the 0th order object is moved closer to the DMD, the reported distances for the other objects appear to get closer to the DMD. This reason this phenomenon occurs is discussed more in Chapter 5.

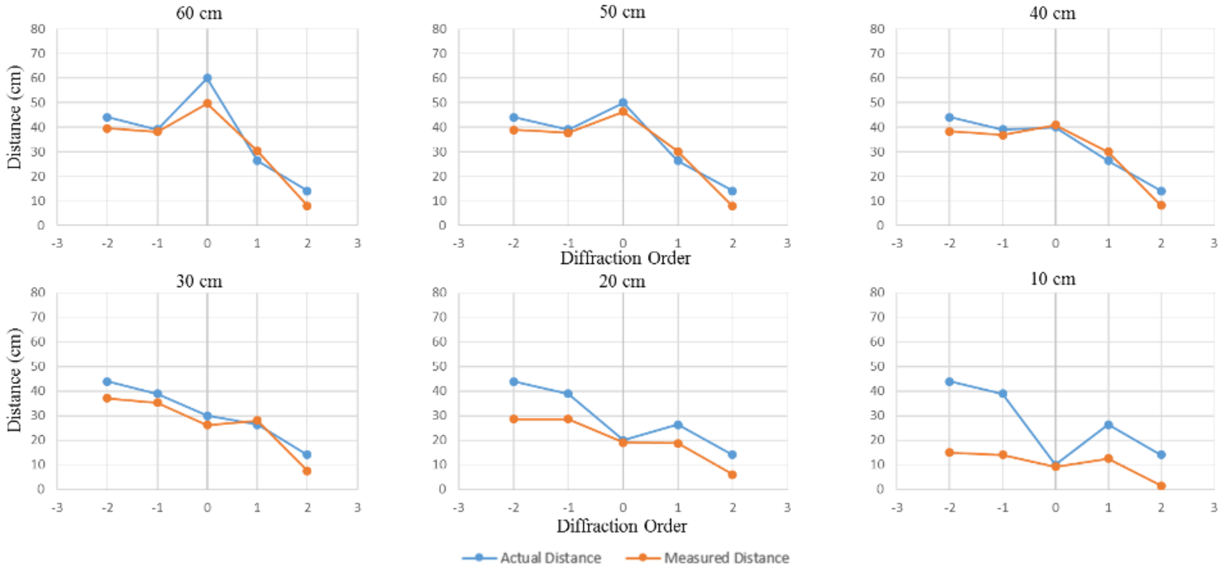


Fig. 20. Specialized Random Object Distance Test showing the degree of cross talk between diffraction orders in the system.

Chapter 5: Discussions

5.1 Origins of cross-talk in system

This experimental demonstration confirms that measurement accuracy is currently less than 1.5cm over a half meter range for all of the 5 diffraction orders for N = 1. In Fig. 17, one can notice that the 0th order can reach far objects at 1.2m but cross talk exists from adjacent diffraction orders when the object is located closer to the DMD chip. The origin of this artifact can be explained as follows. When the micromirrors steer the beam to, for example the +1 order diffraction angle, there is also a specular reflection from the cover glass towards the 0th order diffraction angle. As the object approaches the DMD, both the +1 and 0th order beams illuminate the object. As a result, the system recognizes the object in the 0th order as an object in the 1st order. Each frame of the video represented in Fig. 18 is the sum of all objects detected in one left to right five point scan. Thus, multiple object points are plotted for one single object located close to the DMD in the 0th order. The mechanism causing the artifact is confirmed by the fact that the DMD cover glass has a high specular reflectivity. The reflectivity of the Corning Eagle XG DMD cover glass was measured to be about 20% for 905nm light at a 30° angle of incidence, this reflectivity agrees with literature [16]. We expect that the artifacts can be eliminated by an appropriate antireflection coating since the artifact only occurs for near objects in the 0th diffraction order [16].

5.2 System Proposal to Increase Number of Scanning Spots

Currently, the number of scanning angles is limited to 5 which is equal to the number of diffraction orders available when using 905nm light. The number of scanning angles can be further increased by employing a larger DMD micromirror pitch, and/or a shorter wavelength as well as cascading multiple DMDs. As a matter of fact, the Texas Instruments model DLP9500 DMD has a 15.3µm corner to corner pixel pitch which produces seven horizontal diffraction orders using 905 nm light at a 30° incident angle. In addition, the number of scanning angles can be further increased by using multiple arrayed light sources as shown in the following analysis.

Here, we present one possible solution for increasing the total number of scanning angles by using a stacked laser diode array. We first assume the case of illuminating the DMD surface with normal incidence. The angle between the 0th and +1 diffraction order is defined as θ_{+1} in Eq. (1) where, *p*, is the corner to corner DMD pixel pitch, as described in Fig. 1, and λ is wavelength. Note that a factor of 2 is included in the expression which is necessary if the DMD has diamond-shaped pixels, such as the DLP3000 and DLP9500.

$$\theta_{+1} = \arcsin\left(\frac{2\lambda}{p}\right) \tag{1}$$

Thus, we wish to divide this angular space with N_{LD} laser beams to increase the total scanning resolution of the system by a factor of N_{LD}. These multiple laser beams are assumed to originate from a stack of N_{LD} laser diodes placed at the back focal plane of a collimating lens and these beams are directed at the DMD, as depicted in Fig. 21.

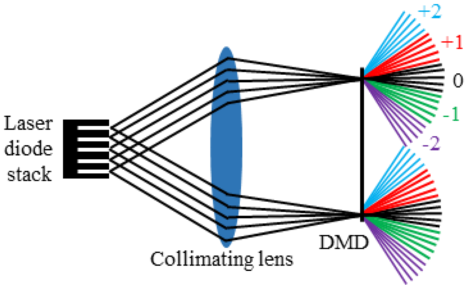


Fig. 21. Layout of proposed system designed to increase number of scan spots. Five laser diodes are split into five diffraction orders to create 25 beams total.

The maximum spatial extent of the laser diode stack is $\pm\frac{N_{LD}d}{2}$. The output laser beams thus have a maximum angular divergence of $\pm\frac{\theta_{+1}}{2}$. These two quantities are related by Eq. (2), where d is the laser diode pitch.

$$\frac{\theta_{+1}}{2} = \arctan\left(\frac{N_{LD}d}{2}\frac{1}{f_{col}}\right) \tag{2}$$

The DMD is assumed to have an area of A_{rec}, thus the maximum linear dimension of the DMD is proportional to $\sqrt{A_{rec}}$. It is assumed that the collimating lens has a numerical aperture of NA_{col} and that the DMD area is completely illuminated. The focal length of the collimation lens, f_{col}, can thus be described by Eq. (3).

$$f_{col} = \frac{\sqrt{A_{rec}}}{2}\sqrt{\frac{1}{NA^2}-1} \tag{3}$$

Combining Eq. (2) and Eq. (3) allows us to create an equation describing the maximum possible number of laser diodes as a function of NA_{col}, A_{rec}, and θ_{+1}, as shown in Eq. (4).

$$N_{LD} \cong \tan\left(\frac{\theta_{+1}}{2}\right)\frac{\sqrt{A_{rec}}}{d}\sqrt{\frac{1}{NA_{col}^2}-1} \tag{4}$$

A_{rec} is related to the maximum measurable range, R, and is given by Eq. (5) [17]. E_T and E_S are the transmitted and received powers respectively, A_{ilm} is the area of the illumination spot, σ is the cross sectional area of the object being detected, and α is the reflectivity of the object. The transmission of the atmosphere and the system are given as η_{atm} and η_{sys}, respectively.

$$R = \sqrt{\frac{E_T}{E_S}\frac{\sigma}{A_{ilm}}\frac{A_{rec}}{\pi}\alpha\left(\eta_{atm}^2\eta_{sys}\right)} \tag{5}$$

We replaced the energy received by the APD, given as E_S in literature [17], with the photosensitivity of the detector, S, by using $E_S \propto \frac{1}{S}$, thus arriving at Eq. (6).

$$R \cong \sqrt{E_T S \frac{\sigma}{A_{ilm}} \frac{A_{rec}}{\pi} \alpha \left(\eta_{atm}^2 \eta_{sys} \right)} \tag{6}$$

The maximum point scan rate of a DMD-based LIDAR system is equal to the pattern refresh rate of the specific DMD used. The line scan rate will be defined as the pattern refresh rate divided by the total number of points within the scan field of view, which is equal to the number of laser diodes, N_{LD}, times the number of diffraction orders supported by the DMD at the current wavelength used, N_{Order}, as shown in Eq. (7).

$$Line\,Scan\,Rate = \frac{Pattern\,Refresh\,Rate}{N_{LD} N_{Order}} \tag{7}$$

We will now use Eq. (4) and Eq. (6) to calculate the approximate range and resolution for LIDAR systems using the DLP3000 and DLP9500 (Texas Instruments) under optimal conditions: $\sigma = A_{ilm}$, η_{atm} and $\eta_{sys} = 1$, and $\alpha = 1$. Eq. (7) will be used to calculate the maximum line scan rate given that the maximum pattern refresh rate of the DLP3000 and DLP9500 is 4kHz and 23kHz respectively. Common laser diode stack pitches range from 0.35mm to 2mm, but stack pitches as low as 0.15mm have also been reported [18]. NA values for collimating lenses of 0.5 have commonly been reported for Petzval lens types [19]. The active area of the DLP3000 and DLP9500 are 0.24cm^2 and 2.40cm^2 respectively. Currently, the sensitivity of the APD used in the demonstration is 23 kV/W. APD sensitivities of up to 1 MV/W have been reported for commercial APD's [20]. Also, the peak optical power of the current laser diode used is 73 W, but state of the art commercial infrared semiconductor laser diodes have peak optical powers up to 100W [21]. By scaling the variables in Eq. (6), a rough estimate of the performance of a single-chip DMD LIDAR system under optimal conditions and using state of the art components was calculated. Table 3 summarizes the estimated performance of LIDAR systems using such parameters for 905nm and 1550nm light.

Table 3. Performance Summary for Two DMD Types

Configuration		Range (m)	N_{LD}	N_{Order}	Total Number of Scan Angles	FOV (°)	Resolution (°)	Scan rate (lines/sec)
DMD Model	Wavelength							
DLP3000	905nm	55	5	5	25	49	1.9	160
DLP3000	1550nm	55	8	3	24	50	2.1	167
DLP9500	905nm	175	11	7	77	48	0.62	299
DLP9500	1550nm	175	18	5	90	60	0.65	256

To show the feasibility of such systems described in Table 3, a ray trace model was constructed in ZEMAX using the DLP3000 and a stack of five 905nm laser diodes to create 25 scan angles as illustrated in Fig. 22Fig. 22. An array of laser diodes (N_{LD} =5) with laser diode pitch $d = 0.15$ mm is collimated by a $NA_{col} = 0.5$ collimating lens ($f_{col} = 4.24$ mm) and redirected to the DMD by a fold mirror. The returning beam from the object is redirected to the $NA_{rec} = 0.14$ receiving lens ($f_{rec} = 17.0$ mm) by the DMD and imaged onto the APD (detection diameter = 3mm). The relationship between the collimating lens focal length and receiving lens focal length can be geometrically calculated assuming the entire area of the APD is filled as described in Eq. (8). The diameter of the APD photosensitive area is d_{APD} and the focal length of the receiving lens is f_{rec}.

$$\frac{N_{LD}d}{f_{col}} = \frac{d_{APD}}{f_{rec}} \tag{8}$$

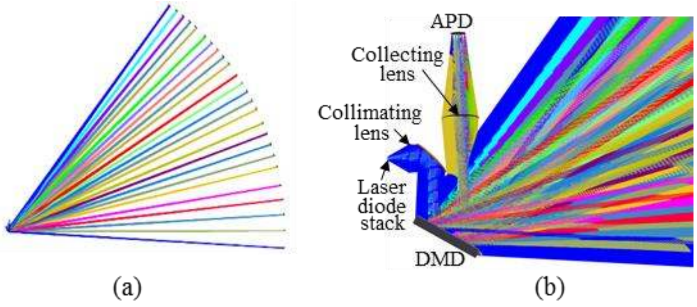


Fig. 22. Model of DMD beam steering using five laser sources to increase number of scan spots to 25. (a) shows the 25 scan directions, (b) shows the 5 laser diodes, collimating lens, DMD, and collection optics.

Commonly, laser diodes emit light in a non-circular manner. For example, we examined a 905nm stacked laser diode bar module from Hamamatsu Photonics (L13713-25P940, Hamamatsu Photonics, Japan). Each laser diode in the stack has a $1/e^2$ beam spread angle of 58° (NA=0.5) in the vertical direction and 15° (NA=0.13) in the horizontal direction [23]. Ideally, these beams would be collimated into circular Gaussian beams. An elliptical beam is undesirable for LIDAR because an elliptical beam will diverge faster along the narrow axis than the wide axis. This beam spreading could then reduce the angular resolution of the system. Thus, the collimating optics described in Fig. 22 would ideally shape the beam differently in the x and y axes. For example, the collimating optics could include an anamorphic prism pair, cylindrical lenses, or anamorphic lenses. Presented in Fig. 23 is one proposed design for a set of collimating optics that shape a 58° x 15° diverging beam into 5 mm diameter circular beam. The source is assumed to be five identical laser diodes spaced by 0.6 mm in a stack. After passing through three anamorphic lenses, each beam becomes circular and well collimated. The five collimated beams extend over a total angular extent of 10°, with 2° separation. As summarized in Table 3, this would increase the resolution of a DMD LIDAR system using 905nm light incident on a DLP3000 from 5 points to 25 points. The lens element prescriptions as well as the system performance is detailed in Appendix A.

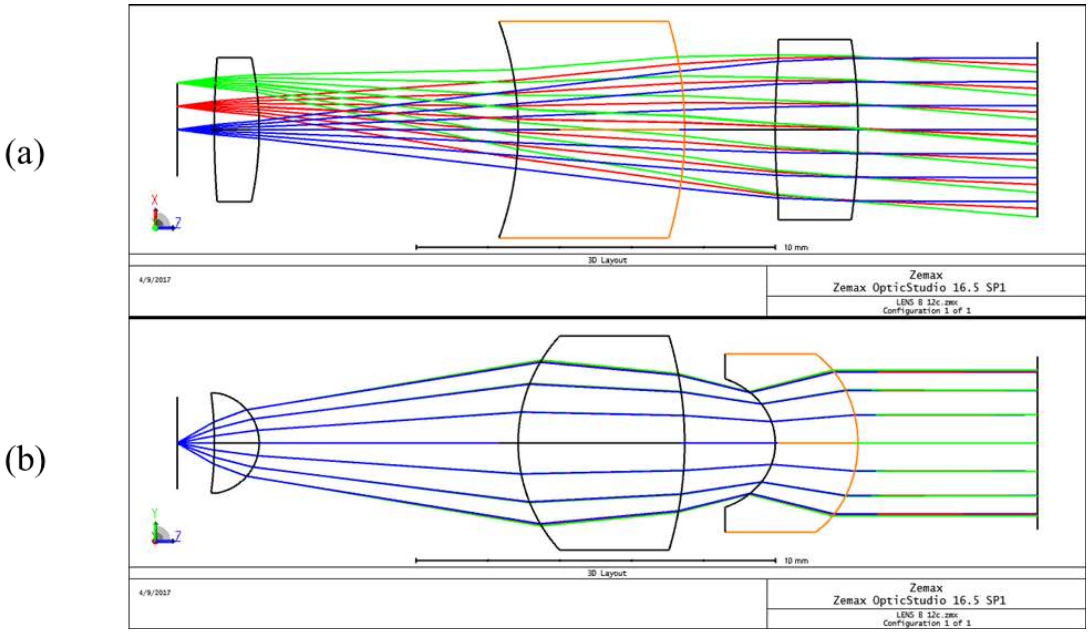


Fig. 23. Layout of an anamorphic collimation system created in ZEMAX. (a) is the cross section of the slow axis, (b) is the cross section of the fast axis.

In terms of scanning speed, currently a scan rate of 3.34k points/sec is demonstrated in our LIDAR system using the DLP3000. This corresponds to a line scan rate of 668 lines/sec with 5 scanning points and a single laser diode. Our analysis shows that a line scan rate of 256 lines/sec with 90 scanning points is feasible when a faster DMD is used, such as the DLP9500 which features a 23kHz point scan rate, which is equal to pattern refresh rate of the DMD.

5.3 Discussion of Dynamic Thresholding

When making distance measurements, our system currently uses a constant voltage threshold value in the TOF circuitry and contains no pulse shaping circuitry. This provides a disadvantage in that the reported TOF is affected by the shape of the return pulse. Factors such as surface reflectivity and measurement distance have an effect on the amplitude of reflected pulses. To further increase measurement accuracy, implementation of additional circuitry and signal processing could increase the accuracy of this LIDAR system. An analog DC detector offset to place the voltage threshold level in the center of the noise distribution has been reported to increase accuracy in LIDAR systems [22]. In addition, analog filtering that shapes the return pulse has also been reported to increase accuracy [22].

Chapter 6: Conclusions

For the first time, to the best of the authors' knowledge, we experimentally demonstrated a single chip LIDAR with an efficient DMD-based discrete beam steering capable of live imaging at 3.34k points/sec and a 48° full field of view. Based on the results of the demonstration, we performed a mathematical modeling of such a single chip LIDAR system given reasonable physical and optical design constraints. It predicts that range finding over 175 m with scanning rates of (256 lines/s) x 90 (points/line) = 23k points/sec and a field of view over 60° with 0.65° resolution is attainable. Such a system would still have the benefits of single chip DMD-based LIDAR, such as reliable Micro-Electro Mechanical System based scanning, flexible selection of wavelengths, and a large beam diameter.

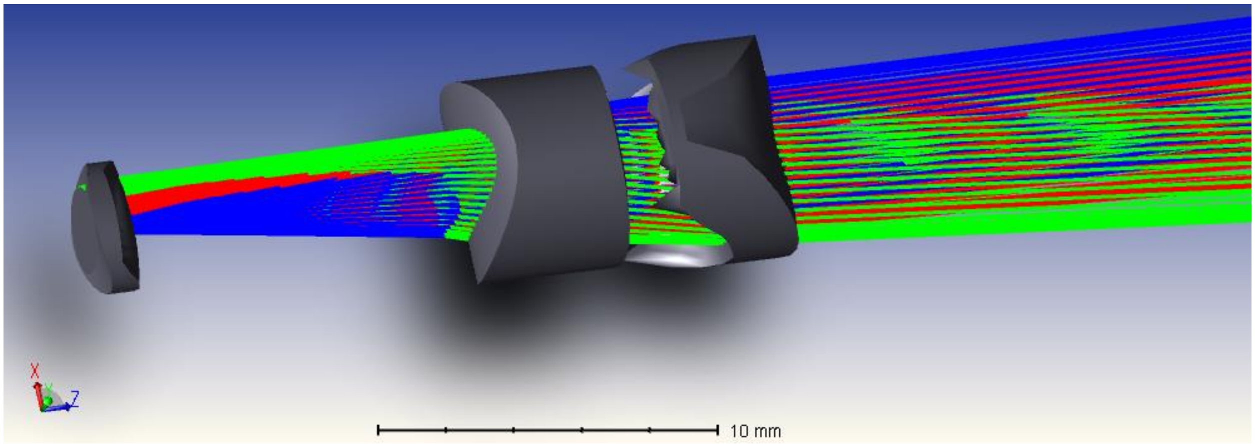
REFERENCES

1. W. C. Stone, M. Juberts, N. G. Dagalakis, J. A. Stone Jr., J. J. Gorman, “Performance Analysis of Next-Generation LADAR for Manufacturing, Construction, and Mobility,” NIST Interagency/Internal Report (NISTIR) – 7117, (2004).
2. R. Cho, A. Han, S. Ju, H. Jeong, J.H. Park, I. Kim, J.U. Bu, and C.H. Ji, "Electromagnetic biaxial microscanner with mechanical amplification at resonance," Opt. Express **23**(13), 16792-16802 (2015).
3. Niclass, K. Ito, M. Soga, H. Matsubara, I. Aoyagi, S. Kato, and M. Kagami, "Design and characterization of a 256x64-pixel single-photon imager in CMOS for a MEMS-based laser scanning time-of-flight sensor," Opt. Express **20**(11), 11863-11881 (2012).
4. T. DeRose, R. D. Kekatpure, D. C. Trotter, A. Starbuck, J. R. Wendt, A. Yaacobi, M. R. Watts, U. Chettiar, N. Engheta, and P. S. Davids, "Electronically controlled optical beam-steering by an active phased array of metallic nanoantennas," Opt. Express **21**(4), 5198-5208 (2013).
5. R. Dou, M. K. Giles, "Programmable phase grating and beam steerer by operating a LCTV," Proc. SPIE **2566**, Advanced Imaging Technologies and Commercial Applications, (23 August 1995).
6. S. Davis, S. Rommel, S. Johnson, G. Farca, N. Rebolledo, S. Selwyn, M. Anderson, "Electro-optic steering of a laser beam," SPIE Newsroom DOI: 10.1117/2.1201105.003715, (2011).
7. E. Ackerman, "Quanergy Announces $250 Solid-State LIDAR for Cars, Robots, and More," IEEE Spectrum, 7 January 2016.
8. Tuantranont, V.M. Bright, J. Zhang, W. Zhang, J.A. Neff, Y.C. Lee, "Optical beam steering using MEMS-controllable microlens array," Sensor Actuat. A-Phys. **91**(3), 363-373 (2001).
9. Mirrocle Technologies Inc., "Mirrorcle Technologies MEMS Mirrors – Technical Overview," Gimbal-les Two-Axis Scanning Micromirror Devices technical overview, 2016.
10. T. Sandner, M. Wildenhain, C. Gerwig, H. Schenk, S. Schwarzer, H. Wölfelschneider, "Large aperture MEMS scanner module for 3D distance measurement," Proc. SPIE 7594 75940D (2010).
11. R. Moss, P. Yuan, X. Bai, E. Quesada, R. Sudharsanan, B.L. Stann, J.F. Dammann, M.M. Giza, W.B. Lawler, "Low-cost compact MEMS scanning LADAR system for robotic applications," Proc. SPIE **8379** 837903 (2012).
12. X. Lee and C. Wang, "Optical design for uniform scanning in MEMS-based 3D imaging lidar," Appl. Opt. **54**(9), 2219-2223 (2015).
13. Texas Instruments, "DLPTM System Optics," Application Report, July 2010.
14. Kaeriyama, Toshiyuki, Damped Control of a Micromechanical Device, U.S. Patent Application No. 10/749,432, filed December 31, 2003.
15. J.D. Gaskil, *Linear Systems, Fourier Transforms, and Optics* (John Wiley & Sons, Inc., 1987), Chap. 10.
16. Texas Instruments, "Wavelength Transmittance Considerations for DLP® DMD Window," Application Report, May 2012 revised March 2014.
17. P. McManamon, *Field Guide to Lidar* (SPIE Press, 2015).
18. R. Feeler and E. Stephens, "High Density Pulsed Laser Diode Arrays for SSL Pumping," Northrop Grumman Cutting Edge Optronics Application Note #15, 2010.
19. W. Smith, *Modern Lens Design* (McGraw-Hill Professional, 2004).

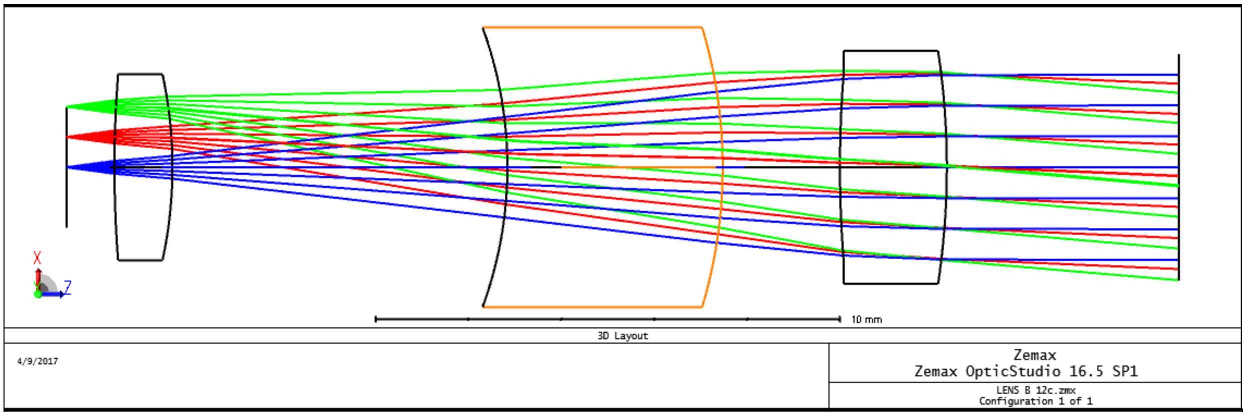
20. Analog Modules, Inc., "High Sensitivity APD Optical Receiver," Model 7510-1 Datasheet, May 2015.
21. Excelitas Technologies, "PGA Series of Single- and Multi-epi 905 nm Pulsed Semiconductor Lasers," Datasheet Photon Detection, 2013.
22. PulsedLight, Inc., "LIDAR-lite Operating Manual," (accessed 13 Feb 2017), www.pulsedlight3d.com.
23. Hamamatsu, "Pulsed Laser Diode Bar Module L13713-25P940," Datasheet, (accessed 17 April 2017), https://www.hamamatsu.com/resources/pdf/lsr/L13713-25P940_E.pdf.

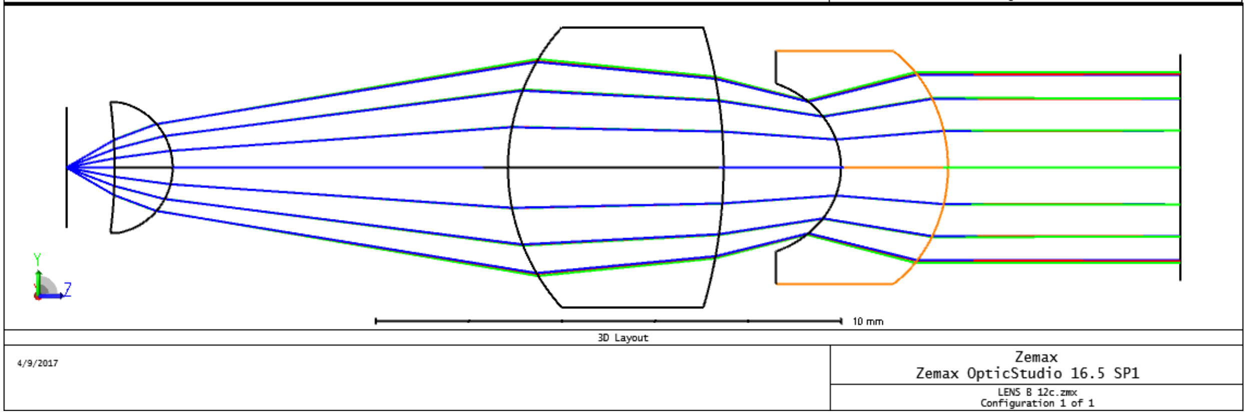
APPENDIX A: Anamorphic Collimator Design

3D System View

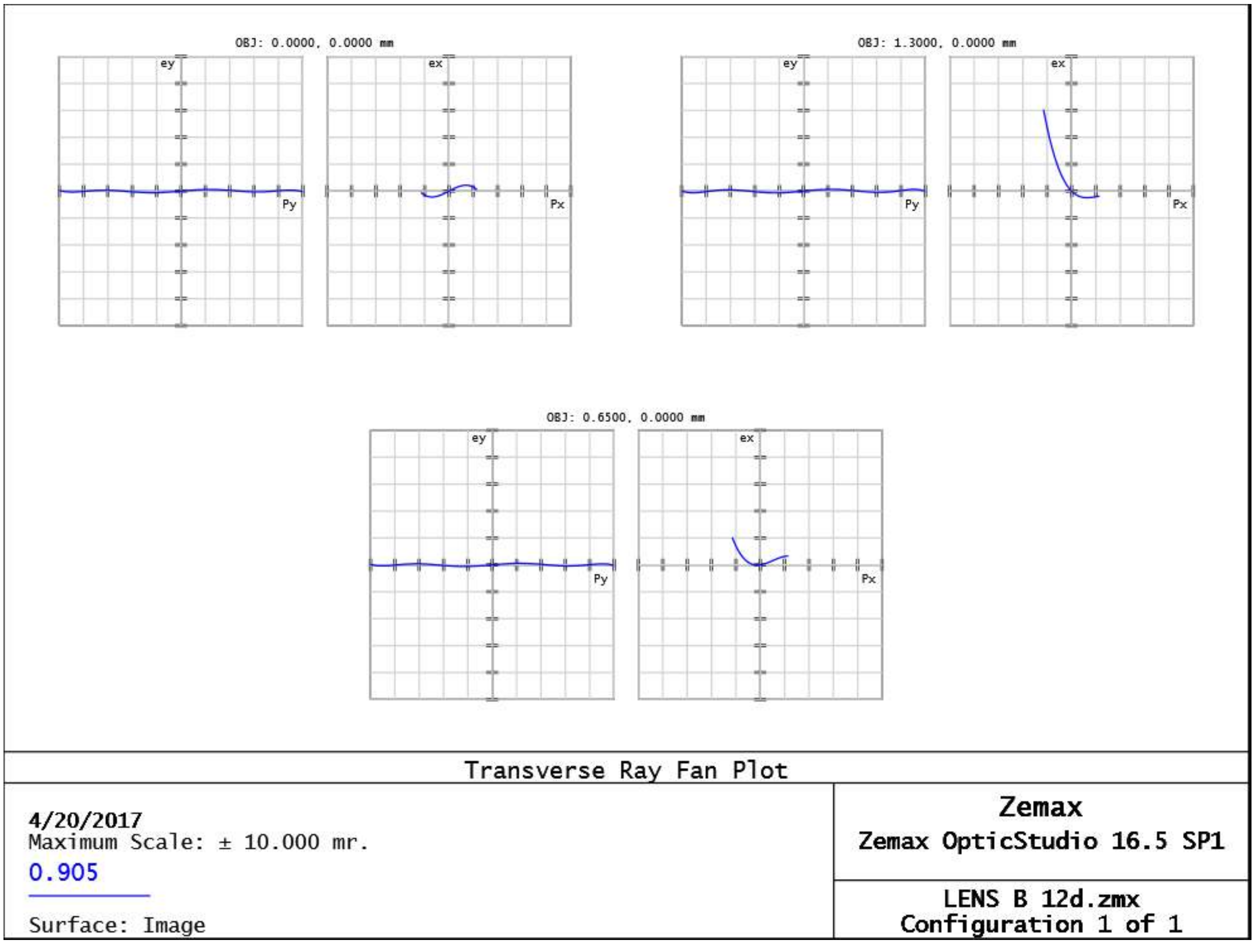


Fast and Slow Axes 2D views





Ray Fan Plot



Transverse Ray Fan Plot

4/20/2017
Maximum Scale: ± 10.000 mr.
0.905

Surface: Image

Zemax
Zemax OpticStudio 16.5 SP1

LENS B 12d.zmx
Configuration 1 of 1

Lens Prescriptions in mm:

Surface	X-radius	Y-radius	Thickness	Material	Semi-Diameter
Object	-	-	0.24		
1	-3.92	11.46	1.25	N-BK7	1.34
2	-0.81	-9.98	10.18		1.40
3	5.14	-395.26	4.63	N-BK7	3.00
4	-7.81	-15.00	2.74		3.00
5	-1.69	22.34	2.30	N-BK7	3.30
6	-3.62	-52.77	40.00		3.30

APPENDIX B: Arduino Code Written

```
// Braden Smith 8/30/2016
// Pulse is first sent through DS1023
// Pulse is sent to DMD, DMD is preset to flip between black and white
images.
// Two switches are present to flip mirrors to correct location
// Timing sequence added to subroutine so multiple spots can be
measured
//
// Rev C:
// Code contains 0 to 32 NOP's to be used for better delay
// Code only works for 45 degree incident angle
//
// Rev D:
// Timing was changed to work with a 35 degree incidence angle.
// Averaging For loop disabled
//
// Ref E:
// Changed timing so that laser now pulses on odd and even flips.
// Averaging For loop removed to allow for even and odd scanning.
// Averaging calculations removed.
// Serial Printing removed for speed.
//
// Ref F:
// Added capability to save to SRAM chip
// Serial interface is present, but not used duing data collection.

// Notes:
  // May need to disable interrupts to acheive a cleaner scanning
  // Starting mirror position assumed black

#include <arduino2.h> // include the fast I/O 2 functions
#include <SPI.h>  // Serial Peripheral Interface
// The following pins are pre-defined
// * Pin 11 = D_In
// * Pin 12 = D_Out
// * Pin 13 = SCLK

long TDC_mask = 0x7FFFFF;     // mask to remove unwanted bits from
read
long TOF = 0;                 // Output from TDC
long byte1 = 0;               // 3 bytes for output from TDC
long byte2 = 0;
long byte3 = 0;

const GPIO_pin_t CS_DL = DP10;    // Chip select pin for Delay Line
(active high)
const GPIO_pin_t Enable = DP4;    // Chip enable pin for TDC (Active
high)
```

```
const GPIO_pin_t CS_TDC = DP5;    // Chip select for TDC (active low)
const GPIO_pin_t CS_RAM = DP3;    // Chip select for SRAM (active low)
const GPIO_pin_t LaserOut = DP9;  // Pin output for laser diode
const GPIO_pin_t DMDPin = DP8;    // Pin used to flip the DMD
int mirror_flip_pin = 7;          // Switch used to flip mirrors
int continue_pin = 6;             // Switch used to continue after
mirror flip

int Init_mirror = false;
int Init_continue = false;
int mirror_state = false;
unsigned long address=0ul;        // Address variable for SRAM
unsigned int combinedData=0;      // Variable to store data read from
SRAM
int Index = 0;                    // Index to save 6000 data points
int Delay1023 = 0;        // 0 to 255
int DelayNOP = 0;         // 0 to 32
int DelayUS = 0;          // Around 220us

// Subroutine Definitions
void MirrorInit();
void InitializePins();
void InitializeTDC();
void InitializeSRAM();
void ReadSRAMdata();

#define NOP __asm__ __volatile__ ("nop\n\t")

void setup() {
  Serial.begin(115200);   // Start arduino serial interface

  SPI.begin();  // Start serial SPI interface
  SPI.beginTransaction(SPISettings(100000, MSBFIRST, SPI_MODE0));
//     data transfer rate
//     MSB sent first
//     SPI Mode1: Data output on clock rising edge

  // Set DS1023 delay chip to 0 delay
  digitalWrite2f(CS_DL,HIGH);
  SPI.transfer(B00000000);
  digitalWrite2f(CS_DL,LOW);

  InitializePins();
  MirrorInit();
  InitializeTDC();
  InitializeSRAM();

  noInterrupts();
}// End setup--------------------------------------------------------
-----

// DMD must be set to white/black image to start off with
// Delay on DS1023 must be set the correct amount to give 8ns pulse to
LD
void loop() {
Serial.println("Collecting Data...");
Serial.println("                  ");
Serial.println("                  ");
Serial.println("                  ");
Serial.println("                  ");
Serial.println("                  ");
Serial.println("                  ");
while(Index<6000){
Index = Index + 1;
// Scan series must start with an "L" type scan point.
// DMD must be set to start with the -2 diff order
// Subsequent scan spots must alternate "L", "R", "L", etc.

  // Spot +2 L
  Delay1023 = 128;      //0 to 255
  DelayNOP = 30;        //0 to 32
  DelayUS = 218;
  Get_Distance();

  // Spot +1 R
  Delay1023 = 128;      //0 to 255
  DelayNOP = 6;         //0 to 32
  DelayUS = 218;
  Get_Distance();

  // Spot 0 L
  Delay1023 = 128;      //0 to 255
  DelayNOP = 13;        //0 to 32
  DelayUS = 218;
  Get_Distance();

  // Spot -1 R
  Delay1023 = 128;      //0 to 255
  DelayNOP = 19;        //0 to 32
  DelayUS = 218;
  Get_Distance();

  // Spot -2 L
  Delay1023 = 128;      //0 to 255
  DelayNOP = 14;        //0 to 32
  DelayUS = 217;
  Get_Distance();

  // Spot +2 R
  Delay1023 = 0;        //0 to 255
  DelayNOP = 12;        //0 to 32
  DelayUS = 217;
  Get_Distance();
  // Spot +1 L
  Delay1023 = 0;        //0 to 255
  DelayNOP = 22;        //0 to 32
  DelayUS = 218;
  Get_Distance();

  // Spot 0 R
  Delay1023 = 0;        //0 to 255
  DelayNOP = 13;        //0 to 32
  DelayUS = 218;
  Get_Distance();

  // Spot -1 L
  Delay1023 = 128;      //0 to 255
  DelayNOP = 6;         //0 to 32
  DelayUS = 218;
  Get_Distance();

    // Spot -2 R
  Delay1023 = 100;      //0 to 255
  DelayNOP = 26;        //0 to 32
  DelayUS = 218;
  Get_Distance();

//--------------------------------------------------------------------
----
//  // Spot +2 L
//  Delay1023 = 128;      //0 to 255
//  DelayNOP = 30;        //0 to 32
//  DelayUS = 218;
//  Get_Distance();
//
//  // Spot +1 L
//  Delay1023 = 0;      //0 to 255
//  DelayNOP = 22;        //0 to 32
//  DelayUS = 218;
//  Get_Distance();
//
//  // Spot 0 L
//  Delay1023 = 128;      //0 to 255
//  DelayNOP = 13;        //0 to 32
//  DelayUS = 218;
//  Get_Distance();
//
//  // Spot -1 L
//  Delay1023 = 128;      //0 to 255
//  DelayNOP = 6;         //0 to 32
//  DelayUS = 218;
//  Get_Distance();
//
//  // Spot -2 L
//  Delay1023 = 128;        //0 to 255
//  DelayNOP = 14;           //0 to 32
//  DelayUS = 217;
//  Get_Distance();
//
//    // Spot -2 R
//  Delay1023 = 100;      //0 to 255
//  DelayNOP = 26;        //0 to 32
//  DelayUS = 218;
//  Get_Distance();
//
//  // Spot -1 R
//  Delay1023 = 128;      //0 to 255
//  DelayNOP = 19;        //0 to 32
//  DelayUS = 218;
//  Get_Distance();

//  // Spot 0 R
//  Delay1023 = 0;      //0 to 255
//  DelayNOP = 13;        //0 to 32
//  DelayUS = 218;
//  Get_Distance();

//  // Spot +1 R
//  Delay1023 = 128;      //0 to 255
//  DelayNOP = 6;         //0 to 32
//  DelayUS = 218;
//  Get_Distance();

//  // Spot +2 R
//  Delay1023 = 0;        //0 to 255
//  DelayNOP = 12;           //0 to 32
//  DelayUS = 217;
//  Get_Distance();

}// End While loop----------------------------------------------------
----
Serial.println("Reading Data...");
Serial.println("              ");
Serial.println("              ");
ReadSRAMdata();
}// End Main Loop-----------------------------------------------------
----

void InitializePins(){
  pinMode2f(LaserOut, OUTPUT);
  digitalWrite2f(LaserOut, LOW);

  pinMode2f(CS_DL, OUTPUT);
  digitalWrite2f(CS_DL,LOW);

  pinMode2f(CS_TDC, OUTPUT);
  digitalWrite2f(CS_TDC, HIGH);
  pinMode(mirror_flip_pin,INPUT);
  pinMode(continue_pin,INPUT);

  pinMode2f(Enable, OUTPUT);
  digitalWrite2f(Enable,LOW);

  pinMode2f(CS_RAM, OUTPUT);
  digitalWrite2f(CS_RAM, HIGH);
}//End InitializePins-------------------------------------------------
---

void MirrorInit(){
  Serial.println("Waiting for mirror setting...");
  Serial.println("                             ");
  Serial.println("                             ");
  Init_continue = digitalRead(continue_pin);
  Init_mirror = digitalRead(mirror_flip_pin);

  while(digitalRead(continue_pin)==Init_continue) {
    mirror_state = digitalRead(mirror_flip_pin);
    if(mirror_state!=Init_mirror) {
      // Flip the mirrors once
      Init_mirror = mirror_state;
      digitalWrite2f(DMDPin,HIGH);
      delayMicroseconds(400);
      digitalWrite2f(DMDPin,LOW);
      delayMicroseconds(400);
    } // End If statement
    // Send pulse to laser
    digitalWrite2f(LaserOut,HIGH);
    NOP;
    digitalWrite2f(LaserOut,LOW);
    delay(1);

    //Disable interrupts
    noInterrupts();

  } // End while loop
  Serial.println("Mirror Setting Complete");
  Serial.println("                       ");
  Serial.println("                       ");
} // End MirrorInit---------------------------------------------------
-

void InitializeTDC(){
  delay(3);       // Delay for TDC chip to start-up
  digitalWrite2f(Enable, HIGH);
  delay(3);

  // Initialize delay chip to constant delay
  SPI.beginTransaction(SPISettings(20000000, MSBFIRST, SPI_MODE0));
//SPI_MODE0 for TDC

  // Setup TDC7200 chip
  digitalWrite2f(CS_TDC, LOW);    // Config 2
  SPI.transfer(0b01000001);   // Address + write
  SPI.transfer(0b00000000);   //..calibration2_periods=2
  //SPI.transfer(0b01000000);   // Calibration2_periods = 10
                              // Perform 1 measurement cycle (No multi
cycle averaging mode)
                              // Singe Stop mode
  digitalWrite2f(CS_TDC, HIGH);

  // set stop mask to 1 clock cycle
  digitalWrite2f(CS_TDC, LOW);    // Counter stop mask high
  SPI.transfer(0b01001000);   // Address + write
  SPI.transfer(0);            // Write
  digitalWrite2f(CS_TDC, HIGH);

  digitalWrite2f(CS_TDC, LOW);    // Counter stop mask low
  SPI.transfer(0b01001001);   // Address + write
  SPI.transfer(0);            // Write
  digitalWrite2f(CS_TDC, HIGH);

  SPI.endTransaction();
} // End TDC Init----------------------------------------------------

void Get_Distance(){
    // Set delay on optional 2nd delay line to fine tune mirror
position
    digitalWrite2f(CS_DL,HIGH);
    SPI.transfer(Delay1023);
    digitalWrite2f(CS_DL,LOW);

    // Calibration sequence for TDC7200
    SPI.beginTransaction(SPISettings(20000000, MSBFIRST, SPI_MODE0));
    digitalWrite2f(CS_TDC, LOW);                        // Calibration 1
    SPI.transfer(0x1B);                             // Address for
calibration 1 register
    byte3 = SPI.transfer(0);                        // upper
    byte2 = SPI.transfer(0);                        // middle
    byte1 = SPI.transfer(0);                        // lower
    digitalWrite2f(CS_TDC, HIGH);

    digitalWrite2f(CS_TDC, LOW);                        // Calibration 2
    SPI.transfer(0x1C);                             // Address for
calibration 2 register
    byte3 = SPI.transfer(0);                        // upper
    byte2 = SPI.transfer(0);                        // middle
    byte1 = SPI.transfer(0);                        // lower
    digitalWrite2f(CS_TDC, HIGH);
    digitalWrite2f(CS_TDC, LOW);    // Config 1 - initiate measurement
    //SPI.transfer(0b01000000);   // Address + write
    //SPI.transfer(0b10000001);   // Write (perform calibration every 
interrupt - 0b10000000)
    SPI.transfer16(0x4081);
    digitalWrite2f(CS_TDC, HIGH);

    // Trigg is output as a rising edge signal
    // Stopped on rising edge of stop signal
    // Started on rising edge of start signal
    // Measurement mode 1

    // Send high signal to DMD to flip
    digitalWrite2f(DMDPin,HIGH);

    // Wait a critical time (adding the optional delay line time)
    delayMicroseconds(DelayUS);    //16 NOP's per 1 us delay

    switch(DelayNOP){ //switch statement delays at increments of 
62.5ns
    case(0):

      break;
    case(1):
      NOP;
      break;
    case(2):
      NOP;NOP;
      break;
    case(3):
      NOP;NOP;NOP;
      break;
    case(4):
      NOP;NOP;NOP;NOP;
      break;
    case(5):
      NOP;NOP;NOP;NOP;NOP;
      break;
    case(6):
      NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(7):
      NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(8):
      NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(9):
      NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(10):
      NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(11):
      NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(12):
      NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(13):
      NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(14):
      NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(15):
      NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(16):
      NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(17):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(18):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;
      break;
    case(19):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;
      break;
    case(20):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;NOP;
      break;
    case(21):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;NOP;NOP;
      break;
    case(22):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;NOP;NOP;NOP;
      break;
    case(23):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;NOP;NOP;NOP;NOP;
```

```
      break;
    case(24):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(25):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(26):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(27):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(28):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(29):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(30):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(31):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    case(32):

NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NO
P;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
      break;
    } //END SWITCH-CASE STATEMENT

    // Send signal to LD through delay line to hit mirror mid
transition
    // This also sends a start signal to the TDC7200
    digitalWrite2f(LaserOut,HIGH);
    NOP;
    digitalWrite2f(LaserOut,LOW);

    // Stop signal should hit APD and stop TDC7200 if object present
    // Read output from TDC7200
    digitalWrite2f(CS_TDC, LOW);       // Read signal
    SPI.transfer(0x10);                // TIME1 register address
    byte3 = SPI.transfer(0x00);        // read 8 upper bits
    byte2 = SPI.transfer(0x00);        // read 8 middle bits
    byte1 = SPI.transfer(0x00);        // read 8 lower bits
    digitalWrite2f(CS_TDC, HIGH);
    TOF = (byte3 << 16) + (byte2 << 8) + byte1;
    TOF = TOF & TDC_mask;

    // Wait for mirrors to stop flipping
    delayMicroseconds(25);

    // Send pulse to DMD to reset mirrors
    digitalWrite2f(DMDPin,LOW);

    // Save output in SRAM
    digitalWrite2f(CS_RAM, LOW);
    SPI.transfer(0x02);     //Write Instruction
    SPI.transfer((address<<8)>>24); // First 8 MSB of 24-bit address
    SPI.transfer((address<<16)>>24); // Second 8 bits of 24-bit
address
    SPI.transfer((address<<24)>>24); //last 8 LSB of 24-bit address
    SPI.transfer16(TOF); //trasnfers the data
    digitalWrite2f(CS_RAM, HIGH);
    address=address + 2;
    /*increment the address. If ARRAYSIZE=200, then we must increment
by 2*200,
     * because the SRAM stores 8bits at each address. We store an
integer, which
     * is two 8-bits. For Example: We store 1 integer in the SRAM.
This is 16 bits.
     * Each address of the SRAM stores 8-bits. Address[0] and
Address[1] are then
     * filled. We must increment to Address[2] to store the next
integer;the next
     * integer (16-bits) will be held in Address[2] (8 bits) and
Address[3] (8 bits).
     */
} // End Get_Distance-----------------------------------------------

void InitializeSRAM(){
  digitalWrite2f(CS_RAM, LOW);
  SPI.transfer(0x01);   //Set SRAM write mode instruction
  SPI.transfer(0b01000000); //Set Write mode to sequential operation
  digitalWrite2f(CS_RAM, HIGH);
} // End InitializeSRAM---------------------------------------------
```

```
void ReadSRAMdata(){
  digitalWrite2f(CS_RAM, LOW);
  SPI.transfer(0x03); //Read instruction
  SPI.transfer(0x00);
  SPI.transfer(0x00);
  SPI.transfer(0x00);
  int Display = 0;
  for(int i=0; i<6000; i++){
    combinedData=SPI.transfer16(0x0000);    //Get 16-bits from the
SRAM. Store those 16bits in combinedData.
    Display = Display + 1;
    Serial.print(combinedData);
    Serial.print("   ");
    if(Display==5){
      Display = 0;
      Serial.print("\n");
    } // End if loop
  } // End for loop
  digitalWrite2f(CS_RAM, HIGH);
  Serial.println("DONE");
  while(1){}  // Stop program in endless loop
} // End ReadSRAMdata--------------------------------------------------
```

APPENDIX C: MATLAB Simulation Code Written

1: Scan_DMD: Simulates scanning DMD across all 5 diffraction orders.

```
function Scan_DMD
% Creates long exposure image of model of DMD scanning

%% Define some variables

% Define numerical model variables
Model.DMD_ang = [-10.3, -5.8, 0, 4.9, 12.0];          % New angles
rounded
% Model.DMD_ang = -12:0.5:12;
% Model.DMD_ang = [-10.8 -7.3, -3.7, 0, 4.1, 8.7, 12];

Model.Img_ang = 0:0.01:60;                   % Sample angles on
diffraction pattern (degrees)
% Model.Img_ang = 0:0.5:60;
% Model.Img_ang = [-20.5200  -10.6000   -0.0100   11.8400
26.5700]+30; % 70x3, 250000, 20spls
% Width = 0.1; % degrees
% Peaks = [-20.520 [Peaks(1)-Width/2: 0.01 :Peaks(1)+Width/2,...
%                  Peaks(2)-Width/2: 0.01 :Peaks(2)+Width/2,...
%                  Peaks(3)-Width/2: 0.01 :Peaks(3)+Width/2,...
%                  Peaks(4)-Width/2: 0.01 :Peaks(4)+Width/2,...
%                  Peaks(5)-Width/2: 0.01 :Peaks(5)+Width/2];

Model.Inc_ang = 30;                         % Incident angles
(degrees)
Model.wave = 0.905;                         % Wavelength in um
Model.Img_dist = 250000;                    % Set distance for use in
numeric integral (25cm)
Model.Cover_glass_present = 1;              % Sets if cover glass is
present in model
Model.Cover_refl = 0.23;                    % Reflectance of DMD
covrglass
Model.Cos_fact=1;                           % Sets to use cosine
correction factor

% Define mirror pattern variables
Mir.X = 70;              % Num mirrors in x direction
Mir.Y = 3;              % Num mirrors in y direction
Mir.width = 10.8;       % Horizontal pixel pitch in um
% Mir.width = 15.3;
Mir.spls = 20;          % Samples per mirror width (diagonal of
mirror) must be even

% Variable Delcarations
```

```
Long_exp = 0;

% Define plotting variables
Plot.DMD_pattern = 0;       % For plotting full DMD patten
Plot.Profile = 0;           % For plotting center DMD profile
Plot.Single_ang = 0;        % For plotting each iteration's
diffraction pattern


%% Run through scan
wht1 = waitbar(0,'Processing');
N=numel(Model.DMD_ang);
Temp_img = zeros(numel(Model.DMD_ang),numel(Model.Img_ang));
for a = 1:N
    [Temp_img(a,:),DifEq,H] =
Braden_DMD_Model(Model.DMD_ang(a),Model,Mir,Plot);
    if Model.Cos_fact==1
        Temp_img(a,:) = Temp_img(a,:).*cos(Model.Img_ang*pi/180);
    end
    % Temp: Intensity of diffracted light for one single DMD angle.
    %       Not normalized
    % Save current frame

Save_Data(Temp_img(a,:),Model.Img_ang,Model.DMD_ang,Model.Inc_ang,a);
    % Add together in long exposure
    Long_exp = Long_exp + Temp_img(a,:);
    waitbar(a/N,wht1);
end
close(wht1);
%% Plot results
% Normalize output
Long_exp = Long_exp/abs(max(max(Long_exp)));
M = 1;
% Plot results and theoretical points on same plot
if Plot.DMD_pattern == 1
    H(2) = figure(4);
else
    H(1) = figure(4);
end
plot(Model.Img_ang-Model.Inc_ang,abs(Long_exp));
% hold on;
% plot([DifEq(1),DifEq(1)],[M,0],'k-',...
%     [DifEq(2),DifEq(2)],[M,0],'k-',...
%     [DifEq(3),DifEq(3)],[M,0],'k-',...
%     [DifEq(4),DifEq(4)],[M,0],'k-',...
%     [DifEq(5),DifEq(5)],[M,0],'k-');
% hold off;
% legend('Numerically Modeled','Grating Equation');

%% Save Figures
Cur_dir = cd('Output Files');
```

```
FileName =
[num2str(Mir.X),'x',num2str(Mir.Y),'_',num2str(Model.Inc_ang),'degAOI_
',num2str(mean(diff(Model.Img_ang))),'deg.fig'];
savefig(H,FileName);
% Save data
X = Model.Img_ang-Model.Inc_ang;
Y = abs(Long_exp);
save('Long Exposure Data','X','Y');

Y = abs(Temp_img);
Y = Temp_img/max(max(Temp_img));
DMD_angles = Model.DMD_ang;
save('Individual Exposure Data','X','Y','DMD_angles');
cd(Cur_dir);
end

function Save_Data(Y,X,DMD_angles,Inc_ang,a)
Cur_dir = cd('Output Files');
X = X-Inc_ang;
Y = abs(Y);
% Plot pattern
h = figure;
plot(X,Y);grid on;
% Save plot
Name = ['Single Angle ',num2str(DMD_angles(a))];
title(Name);xlabel('Angle (Degrees)');ylabel('Relative intensity');
savefig(h,[Name,'.fig']);
close(h);
% Save data
save([Name,'.mat'],'X','Y');

cd(Cur_dir);
end
```

2: Braden_DMD_Model: Diffraction model of DLP3000 with 905 light

```
function [Img_int,DifEq,H] = Braden_DMD_Model(DMD_ang,Model,Mir,Plot)
% Model DLP 3000 with 905 light for use in Lidar system
% Braden Smith
% 12/16/2016
%
% Notes:
%

%% Process Variables
Model.Img_ang = Model.Img_ang*pi/180;   % Convert to radians
DMD_ang = DMD_ang*pi/180;               % Convert to radians
Model.Inc_ang = Model.Inc_ang*pi/180;   % Convert to radians

%% Create Phase Model of DMD
[Mir,H] = Create_DMD_phase(Mir,DMD_ang,Model.Inc_ang,Plot,Model);

%% Create distance matrices
% X distance matrix
xv = linspace(-Mir.X*Mir.width/2,Mir.X*Mir.width/2,Mir.X*Mir.spls);
yv = linspace(Mir.Y*Mir.width/2,-Mir.Y*Mir.width/2,Mir.Y*Mir.spls);
[X_ind,Y_ind] = meshgrid(xv,yv);

%% Create phase model of coverglass
if Model.Cover_glass_present==1
    Cover_X_ind = X_ind - min(min(X_ind));
    Cover.OPD = Cover_X_ind*sin(Model.Inc_ang);
    Cover.height = zeros(size(Mir.height));
    Cover.mask = ones(size(Mir.mask))*Model.Cover_refl;
end
%% Perform numeric diffraction integral
Img_field_DMD =
Numerical_Integral(Model,X_ind,Y_ind,Mir.OPD,Mir.height,Mir.mask);
if Model.Cover_glass_present==1
    Img_field_cover =
Numerical_Integral(Model,X_ind,Y_ind,Cover.OPD,Cover.height,Cover.mask
);
    % Add two image intensities together
    Img_int = abs(Img_field_DMD).^2 + abs(Img_field_cover).^2;
else
    Img_int = abs(Img_field_DMD).^2;
end
%% Calculate theoretical diffraction spot locations (Grating equation)
Orders = [-4, -2, 0, 2, 4];
DifEq = asin(Orders*Model.wave/Mir.width-sin(-Model.Inc_ang));
DifEq = DifEq - Model.Inc_ang;     % Convert to angle off of zero
order
DifEq = DifEq*180/pi;              % Convert to degrees

% Measured Angles
% Measured_diff_angles = [-20.5, -11, 0, 11.5, 27.5];

%% Plot results
if Plot.Single_ang==1
    Model.Img_ang = Model.Img_ang-Model.Inc_ang;  % Convert to degrees
off of zero order
    Model.Img_ang = Model.Img_ang*180/pi;         % Convert to degrees
    M = max(max(Img_int));
    % Plot results and theoretical points on same plot
    figure(3);plot(Model.Img_ang,Img_int);
    hold on;
    plot([DifEq(1),DifEq(1)],[M,0],'k-',...
        [DifEq(2),DifEq(2)],[M,0],'k-',...
        [DifEq(3),DifEq(3)],[M,0],'k-',...
        [DifEq(4),DifEq(4)],[M,0],'k-',...
        [DifEq(5),DifEq(5)],[M,0],'k-');
    hold off;
    title('Horizontal diffraction pattern');xlabel('Angle (degrees)');
    legend('Numerically Modeled','Grating Equation');
end % End if statement

end % Main Function
```

3: Create_DMD_Phase: Function to create phase layout of DMD with micromirrors at different angles.

```
function [Mir_out,H] =
Create_DMD_phase(Mir,DMD_ang,Inc_ang,Plot,Model)
% Creates DMD phase profile
% Braden Smith
% 12/27/2016
%
% Ver 2: Assumes normal incidence
% Ver 3: Does not assume normal incidence, needs DMD and incidence
angles
% Notes:
% Updates Mir structure

H = [];
Mir_out = Mir;
%% Create one DMD mirror layout
xv = linspace(-1,1,Mir.spls);
yv = linspace(1,-1,Mir.spls);
[X,Y] = meshgrid(xv,yv);
% Create single mirror layout
fx = 0.925*cos(DMD_ang);
fy = 0.925;
Mir.lay = double(abs(X)/fx+abs(Y)/fy<=1);

%% Create mirror height and mask
% Define max OPD across mirror diagonal
Max_height = tan(DMD_ang)*Mir.width;
Mir.height = linspace(Max_height/2,-Max_height/2,Mir.spls);
Mir.height = repmat(Mir.height,Mir.spls,1);
% Combine Mirror outline and Height Profile
Single_mir = Mir.height.*Mir.lay;
% Create single mirror template
Cell = zeros(Mir.spls);
Cell(1,1) = 1;
Cell(Mir.spls/2+1,Mir.spls/2+1) = 1;
% Create mirror map
Mir.map = repmat(Cell,Mir.Y+1,Mir.X+1);
% Create index matrix
xv = linspace(1,2*Mir.X+3,size(Mir.map,2));
X = repmat(xv,[size(Mir.map,1),1]);
Mir.indx = round(X.*Mir.map);
% Zero pad Mirror Map and index matrices
Mir.map =
[zeros(size(Mir.map,1),Mir.spls),Mir.map,zeros(size(Mir.map,1),Mir.spl
s)];
Mir.indx =
[zeros(size(Mir.indx,1),Mir.spls),Mir.indx,zeros(size(Mir.indx,1),Mir.
spls)];
% Convolve mirrors with map
```

```
Mir.height = conv2(Single_mir,Mir.map);
% Create mask for mirrors
Mir.mask = conv2(Mir.lay,Mir.map);

%% Create mirror OPD profile
N = max(max(Mir.indx));
Mir.OPD = 0;
for a=1:N
    % Get distance from edge
    ln = (a-1)/2*Mir.width;
    % Create height profile
    d = sin(Inc_ang)*ln;
    s = Mir.width/2*sin(DMD_ang+Inc_ang);
    a1 = d-s;
    a2 = d+s;
    Heightv = linspace(a1,a2,Mir.spls);
    Heightv = repmat(Heightv,[Mir.spls,1]);
    % Add height profile to one mirror
    Mir.Single = Mir.lay.*Heightv;
    % Convolve with all mirrors in collumn
    TempMap = double(Mir.indx==a);
    TempMap = conv2(Mir.Single,TempMap);
    % Add collumn to DMD surface
    Mir.OPD = Mir.OPD + TempMap;
end

% Trim Off extra
[SizeY,SizeX] = size(Mir.height);
A = Mir.spls;
SizeY = 0.5*A+1:SizeY-1.5*A+1;
SizeX = 1.5*A+1:SizeX-A*2.5+1;
Mir.height = Mir.height(SizeY,SizeX);
Mir.mask = Mir.mask(SizeY,SizeX);
Mir.OPD = Mir.OPD(SizeY,SizeX);
% Account for cover glass reflection
Mir.mask = Mir.mask*(1-Model.Cover_refl);
%% Display some plots
if Plot.DMD_pattern == 1
    Xmin = -Mir.X*Mir.width/2;
    Xmax = Mir.X*Mir.width/2;
    Ymin = -Mir.Y*Mir.width/2;
    Ymax = Mir.Y*Mir.width/2;
    Limits = [Xmin Xmax Ymin Ymax];
    Fontsize = 10;
    H(1) = figure(1);

    subplot(1,2,1);
    imagelbl(Mir.height,Limits);
    axis square
    xlabel('X distance (\mum)','FontSize',Fontsize);ylabel('Y distance
(\mum)','FontSize',Fontsize);
    h = colorbar;
    xlabel(h,'Mirror Height (\mum)','FontSize',Fontsize);

    subplot(1,2,2);
    imagelbl(Mir.OPD,Limits);
    axis square
    xlabel('X distance (\mum)','FontSize',Fontsize);ylabel('Y distance
(\mum)','FontSize',Fontsize);
    h = colorbar;
    xlabel(h,'Optical Path Difference (\mum)','FontSize',Fontsize);
end

if Plot.Profile == 1
    figure(2);
    subplot(2,1,1);
    plot(Mir.height(size(Mir.height,1)/2,:));title('Profile of mirror
height');ylabel('Height (um)');
    subplot(2,1,2);
    plot(Mir.OPD(size(Mir.OPD,1)/2,:));title('Profile of mirror
OPD');ylabel('OPD (um)');
end

% Save Variables
Mir_out.OPD = Mir.OPD;
Mir_out.height = Mir.height;
Mir_out.mask = Mir.mask;

end % End main function
```

4: Numerical_Integral: Function to perform Huygen-Fresnel integral for given micromirror configuration

```
function [Img_field] = Numerical_Integral(Model, X_ind, Y_ind, OPD,
height, mask)

% Process variables
Img_field = zeros(size(Model.Img_ang));   % Variable definition

% Perform numeric diffraction integral
wtb = waitbar(0,'Processing...');
% Loop through all object points
for Obj_ind = 1:numel(OPD)
    % Loop through image points
    for Img_ind = 1:numel(Model.Img_ang)
        % Get OPD of current point
        Cur_OPD = OPD(Obj_ind);
        % Get current distance
        Cur_mir_pt = [X_ind(Obj_ind),Y_ind(Obj_ind),height(Obj_ind)];
        Cur_img_pt =
[Model.Img_dist*sin(Model.Img_ang(Img_ind)),0,Model.Img_dist*cos(Model
.Img_ang(Img_ind))];
        Cur_dist = sqrt(sum((Cur_img_pt-Cur_mir_pt).^2));
        % Convert distance to phase
        Img_OPD = Cur_OPD + Cur_dist;
        Img_phase = exp(1i*Img_OPD*2*pi/Model.wave);
        % Add to current
        Img_field(Img_ind) = Img_field(Img_ind) +
1/(Cur_dist^2)*Img_phase*mask(Obj_ind);
    end % End image index loop
     waitbar(Obj_ind/numel(OPD),wtb);
end % End object index loop
close(wtb);

end
```

5: imagelbl: Function made to plot results using “imagesc()” function in MATLAB

```
function imagelbl( Image, Limits )
%IMAGELBL Plots an image with user-defined axis labels.
%Limits = [Xmin Xmax Ymin Ymax]
% Plot image
h = imagesc(Image);

Xmin = Limits(1);
Xmax = Limits(2);
Ymin = Limits(3);
Ymax = Limits(4);

% Create 11 labels
Xaxis = round(linspace(Xmin,Xmax,5),0)';
Yaxis = round(linspace(Ymax,Ymin,5),0)';

% Get data from size of image
X = size(Image,2);
Y = size(Image,1);

numx = numel(Xaxis);
numy = numel(Yaxis);

% Change axis labels and ticks
h.Parent.XTick = round(linspace(1,X,numx));
h.Parent.XTickLabel = num2str(Xaxis);
h.Parent.YTick = round(linspace(1,Y,numy));
h.Parent.YTickLabel = num2str(Yaxis);

end
```