ORTA DOĞU TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜ

TEZ ŞABLONU ONAY FORMU
THESIS TEMPLATE CONFIRMATION FORM

1. Şablonda verilen yerleşim ve boşluklar değiştirilmemelidir.
2. Jüri tarihi Başlık Sayfası, İmza Sayfası, Abstract ve Öz'de ilgili yerlere yazılmalıdır.
3. İmza sayfasında jüri üyelerinin unvanları doğru olarak yazılmalıdır.
4. Tezin son sayfasının sayfa numarası Abstract ve Öz'de ilgili yerlere yazılmalıdır.
5. Bütün chapterlar, referanslar, ekler ve CV sağ sayfada başlamalıdır. Bunun için **kesmeler** kullanılmıştır. **Kesmelerin kayması** fazladan boş sayfaların oluşmasına sebep olabilir. Bu gibi durumlarda paragraf (¶) işaretine tıklayarak kesmeleri görünür hale getirin ve yerlerini **kontrol edin**.
6. Figürler ve tablolar kenar boşluklarına taşmamalıdır.
7. Şablonda yorum olarak eklenen uyarılar dikkatle okunmalı ve uygulanmalıdır.
8. Tez yazdırılmadan önce PDF olarak kaydedilmelidir. Şablonda yorum olarak eklenen uyarılar PDF dokümanında yer almamalıdır.
9. **Bu form aracılığıyla oluşturulan PDF dosyası arkalı-önlü baskı alınarak tek bir spiralli cilt haline getirilmelidir.**
10. Spiralli hale getirilen tez taslağınızdaki ilgili alanları imzalandıktan sonra, Tez Juri Atama Formu ile birlikte bölüm sekreterliğine teslim edilmelidir.
11. Tez taslağınız bölüm sekreterliğiniz aracılığıyla format ve görünüm açısından kontrol edilmek üzere FBE'ye ulaştırılacaktır.
12. FBE tarafından kontrol işlemleri tamamlanan tez taslakları, öğrencilere teslim edilmek üzere bölüm sekreterliklerine iletilecektir.
13. Tez taslaklarının kontrol işlemleri tamamlandığında, bu durum öğrencilere METU uzantılı öğrenci e-posta adresleri aracılığıyla duyurulacaktır.
14. Tez taslakları bölüm sekreterlikleri tarafından öğrencilere iletileceği için öğrencilerimizin tez taslaklarını enstitümüzden elden alma konusunda ısrarcı olmamaları beklenmektedir.
15. Tez yazım süreci ile ilgili herhangi bir sıkıntı yaşarsanız, **Sıkça Sorulan Sorular (SSS)** sayfamızı ziyaret ederek yaşadığınız sıkıntıyla ilgili bir çözüm bulabilirsiniz.

1. Do not change the spacing and placement in the template.
2. Write defense date to the related places given on Title page, Approval page, Abstract and Öz.
3. Write the titles of the examining committee members correctly on Approval Page.
4. Write the page number of the last page in the related places given on Abstract and Öz pages.
5. All chapters, references, appendices and CV must be started on the right page. **Section Breaks** were used for this. **Change in the placement** of section breaks can result in extra blank pages. In such cases, make the section breaks visible by clicking paragraph (¶) mark and **check their position.**
6. All figures and tables must be given inside the page. Nothing must appear in the margins.
7. All the warnings given on the comments section through the thesis template must be read and applied.
8. Save your thesis as pdf and Disable all the comments before taking the printout.
9. **Print two-sided the PDF file that you have created through this form and make a single spiral bound.**
10. Once you have signed the relevant fields in your thesis draft that you spiraled, submit it to the department secretary together with your Thesis Jury Assignment Form.
11. Your thesis draft will be delivered to the GSNAS via your department secretary for controlling in terms of format and appearance.
12. The thesis drafts that are controlled by GSNAS, will be sent to the department secretary to be delivered to the students.
13. This will be announced to the students via their METU students e-mail addresses when the control of the thesis drafts has been completed.
14. As the thesis drafts will be delivered to the students by the department secretaries, we are expecting from our students no to insist about getting their theses drafts from the Institute.
15. If you have any problems with the thesis writing process, you may visit our **Frequently Asked Questions (FAQ)** page and find a solution to your problem.

3D IMAGING VIA BINARY WAVEFRONT MODULATION FOR LIDAR AND MACHINE VISION APPLICATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÇAĞDAŞ ANIL YÜKSEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MICRO AND NANOTECHNOLOGY

SEPTEMBER 2020

Approval of the thesis:

3D IMAGING VIA BINARY WAVEFRONT MODULATION FOR LIDAR AND MACHINE VISION APPLICATIONS

submitted by Çağdaş Anıl YÜKSEL in partial fulfillment of the requirements for the degree of **Master of Science** i**n Micro and Nanotechnology, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar

Dean, Graduate School of Natural and Applied Sciences

Prof. Dr. Almıla Güvenç Yazıcıoğlu

Head of the Department, Micro and Nanotechnology, METU

Assist. Prof. Dr. Emre Yüce

Supervisor, Physics, METU

Examining Committee Members:

Assist. Prof. Dr. Emre Yüce

Physics, METU

Assoc. Prof. Dr. Serdar Kocaman

Electrical-Electronics Engineering, METU

Assist. Prof. Dr. Ihor Pavlov

Physics, METU

Assoc. Prof. Dr. Kıvanç Azgın

Mechanical Engineering, METU

Assoc. Prof. Dr. Halil Berberoğlu

Physics, AHBVU

Date: 22.09.2020

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Çağdaş Anıl Yüksel

Signature :

ABSTRACT

3D IMAGING VIA BINARY WAVEFRONT MODULATION FOR LIDAR AND MACHINE VISION APPLICATIONS

Yüksel, Çağdaş Anıl
Master of Science, Micro and Nanotechnology
Supervisor: Assist.Prof. Dr. Emre Yüce

September 2020, 95 pages

Autonomous vehicles have proven to be very efficient in daily routine jobs and their impact will continue to increase given the recent developments in artificial intelligence, boosted by increased computation capacity. These vehicles are generally equipped with 2D imaging sensors and asked to accomplish tasks in a 3D world, which hamper their functionalities. In this study, we experimentally investigate and develop 3D imaging technologies. We first demonstrate colorful 3D imaging via time of flight measurements in order to map the target scene to a 3D point cloud data and secondly perform triangulation method. For time of flight measurements, we employ a digital micro-mirror device with a resolution of 1920x1080 and a range finder in order to scan the target. We scan the target with a resolution of 8x2, 16x2, and 8x4 using our first method. Second, we employ structured light method to scan the target in 3D using again the digital micro-mirror device in conjunction with a white light source. Using triangulation, we were able to scan the target with a resolution 1920x1080. Our results indicate that in terms of resolution triangulation method gives better results. We can foresee that triangulation and time of flight measurement methods are combined a robust and high-resolution

3D vision can be developed which will increase vision capabilities of autonomous vehicles that shape our future.

Keywords: LIDAR, 3D Imaging, Digital Micro Mirror Device, Digital Light Processing, Machine Vision.

ÖZ

DALGA ÖNÜ MODÜLASYONU KULLANILARAK LİDAR VE MAKİNE VİZYONU İÇİN 3 BOYUTLU GÖRÜNTÜLEME

Yüksel, Çağdaş Anıl
Yüksek Lisans, Mikro ve Nanoteknoloji
Tez Yöneticisi: Assist.Prof. Dr. Emre Yüce

Eylül 2020, 95 sayfa

Otonom araçlar günlük rutin işler söz konusu olduğunda oldukça verimli olduklarını kanıtlamıştır ve artan bilgi işlem kapasitesi ile birlikte yapay zeka alanındaki gelişmeler de göz önünde bulundurulduğunda bu araçların etkileri artmaya devam edecektir. Bu araçlar genellikle 2 boyutlu sensörler ile donatılmıştır ve bu da 3 boyutlu dünyada işlevlerinin kısıtlanmasına yol açmaktadır. Bu çalışmada 3boyutlu görüntüleme teknolojilerini deneysel olarak araştırıyor ve geliştiriyoruz. İlk olarak hedef sahneyi 3 boyutlu nokta bulutu verileri ile haritalandırmak için uçuş süresi yöntemini gösteriyoruz ve çalışmanın ikinci kısmında ise üçgenleme yöntemini kullanarak hedef sahneyi gerçek renkli ve 3 boyutlu olarak haritalandırıyoruz. Uçuş süresi ölçümleri için 1920*1080 çözünürlüğe sahip dijital mikroayna aygıtını ve lazer mesafe ölçeri kullanarak sahneyi tarıyoruz. Hedef sahneyi 8x2, 16x2, 8x4 çözünürlüğünde tarayabiliyoruz. İkinci kısımda ise sahneyi, üçgenleme (nirengi) methodunu yine “dijital mikroayna aygıtı” ve beyaz ışık kullanarak tarıyoruz. Üçgenleme methodu ile sahneyi daha yüksek çözünürlükte tarama kabiliyetine ulaşabiliyoruz. Sonuçlarımız gösteriyor ki çözünürlük açısından üçgenleme methodu daha iyi sonuçlar veriyor. Çalışmalar sonucu öngörüyoruz ki üçgenleme ve uçuş

süresi methodları birleştirildiği taktirde yüksek çözünürlüklü ve daha sağlam bir method geliştirilerek geleceğimizi şekillendirecek otonom araçların görüş yeteneklerini artırabiliriz.

Anahtar Kelimeler: LIDAR, adaptif optik (DMD), Dalga-önü Şekillendirilmesi, Uzaktan 3 Boyutlu Kızılötesi Algılama, Atmosferik İletim.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

To my family

ACKNOWLEDGMENTS

This work has resulted in the contribution of many wonderful people with many years of experience in their specific field.

First of all, special thanks to METU gives me endless stories and a place that I met with my wife, gives me a chance of globally thinking and supports that work with METU-GAP-105-2018-2797.

My advisor, Emre YÜCE that I learned many, always positive, everyday encouraged, thought me as one of his own sons. He teaches me not only working hard but also working smart and clear and will continue to stay in my future life.

Middle East Technical University Music's and Fine Arts people providing us mannequin and large amount entertainment.

Middle East Technical University Physics Department Mechanical Group's Machining Assistance, with very friendly approach and understanding my desire in a perfect way as DMD mount.

M.Safa Dal, researcher at Nanomagtenics special thanks for the time of flight measurements.

Şahin KÜREKÇİ, appreciate for providing how to math, told me always interesting story about gravitational physics and everyday life events.

Berk Berkan Turgut, research group at TUBITAK SAGE (optical system design), for atmospheric transmittance calculation.

Volkan Ertürk and Serhat Durum, give me an ability of efficient coding perspective.

Melike Yüksel, my life, my love, my friend, my beautiful women as always be with me.

Can Demir Yüksel, my extremely sweet son, best thing that I have in my whole life.

My family, I would like to be with them as they exist, thanks my brother for charging my battery.

To the many not named here providing various levels of support (in many way), I greatly appreciate you and kindly ask you forgive my omission

CHAPTER 1

INTRODUCTION

The three-dimensional digital revolution is taking place. It is very much likely that orders will be delivered by drone couriers or people will be able to take world tours in self-driving cars. The world may be revolutionized in such a way that robots will do all the work and the only concern that remains for humanity will become art and science. This journey started with the motivation of a dream deeply rooted in our souls; contribution to humanity. On this path, artificial intelligence is our "child". The creation of our child is still in progress and now it opens its eyes into a three-dimensional world.

While varying 3D imaging techniques are useful for different applications, time of flight (TOF) and triangulation methods have met the criteria including accuracy and the effective range. The main goal of this study is to reconstruct a 3D image of a scene with two different active techniques; TOF and triangulation.

1.1. 3D IMAGING

We are in the age of transition where machines and autonomous vehicles are taking an essential role in daily applications as well as scientific modeling. A large amount of data is required in order to train the learning algorithms that are part of the highly skilled machines. 3D imaging technologies are rapidly growing, given the increasing demand for machine vision technologies. Having more data is directly proportional to having more benefits for the machines for training their networks more precisely. There occurs a huge effort for supplying data's for the machine. Specific sensors such as LIDAR's are the candidate as data suppliers. 3D technologies are required for utilization such as remote sensing and object recognition.[1] Moving from two dimensions to the third gives an opportunity for analyzing volume, shape, position, as well as defects, which are difficult to

reconstruct in 2D. However, the data capacity also increases with increasing dimension: Mega voxel includes 1000 times more data than a Megapixel.

3D sensor market is growing day by day. "*The level of interest for 3D imaging only as a LIDAR in self-driving car industry is indicated by Yole Development report that predicts the current US$ 0.8 billion automotive market for LIDAR is likely to expand to ~US$5 billion by 2023, and to ~US$ 28 billion by 2032"*[2]**.** There are many techniques to handle specific 3D imaging challenges. In **Hata! Başvuru kaynağı bulunamadı.** different techniques so far are given

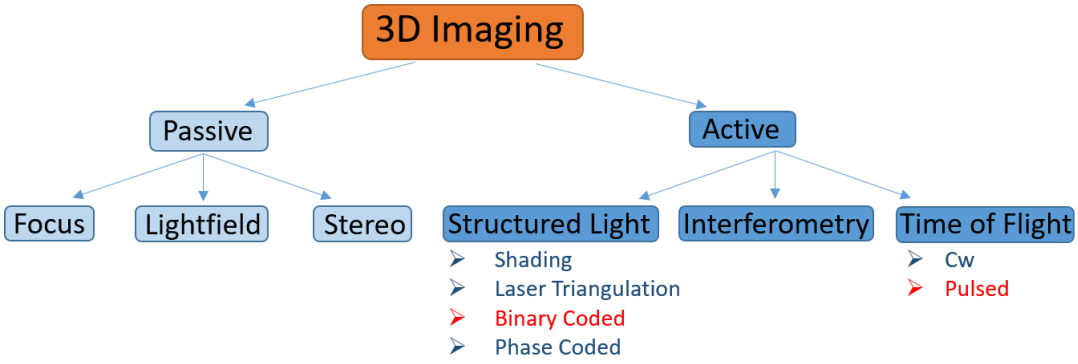


Figure 1.1 Main 3D imaging techniques in the literature. 3D Imaging mainly divided into two as active and passive 3d imaging. Passive imaging is classified according to the reconstruction technique. Focus technique capture image with changing focal length and determine the distance of an object using sharpness value. Lightfield technique is based on a light field camera which has a main lens and additional small lenses for every pixel on camera which measures wavefront coming from the specific pixel. The stereo technique uses one dynamic aperture camera or two cameras and it is based on the triangulation principle. Active 3D imaging is classified with the title Structural Light, Interferometry and Time of Flight. Structural Light based on a triangulation principle like stereo imaging, however, it actively uses a different structured light technique to reconstruct 3D of the scene. The interferometry principle uses different wavelengths and reconstructs the scene based on interferometric techniques. The time of flight technique uses pulsed or continuous wave light source and based on the flight time of light from source to distance.

In Figure 1.1, it is demonstrated that 3D Imaging mainly divided into two groups which are passive and active imaging techniques. In passive techniques, no touch (in any way such as sound or light source) on screen is allowed. Naturally

emitted sources or reflectance of unrelated sources are used as information on the scene and gathered by instruments such as camera, photodiode or hydrophone etc. However, in active technique one can contact the screen. For example, sound or light pulses can be used for direct time of flight detection to reconstruct the screen.

Light Detection & Ranging (LIDAR) incorporates shorter wavelengths compared to radio waves in a RADAR (Radio Detection And Ranging). For this reason, LIDAR offers a better spatial resolution. The time of flight (TOF) technique is used a pulsed source to be transmitted to and received from the target, which measures the time difference between them to reconstruct 3D data. TOF measurements are indeed the backbone of LIDAR systems and are based on measuring the time delay between the transmitted and received pulses.[3].

Structured light that is obtained by modulating both amplitude and phase is commonly used for triangulation methods. Spatial light modulator (SLM for our case as a binary amplitude modulator, digital micro-mirror device (DMD)) is used for modulating light beam and scanning the target scene. In general, 2D imaging cameras coupled with SLMs that create structured light for triangulation require less computational work as compared to the stereo vision method that employs two cameras or more.[49] **Hata! Başvuru kaynağı bulunamadı.** illustrates the performance (the resolution vs. the effective working distance) of the most common methods that are shown in Figure 1.1. Among these methods, triangulation performs outstandingly in terms of resolution and effective range. Time of flight (TOF), on the other hand, stands out with its effectiveness at long distances. In this work, we focus on these two methods.

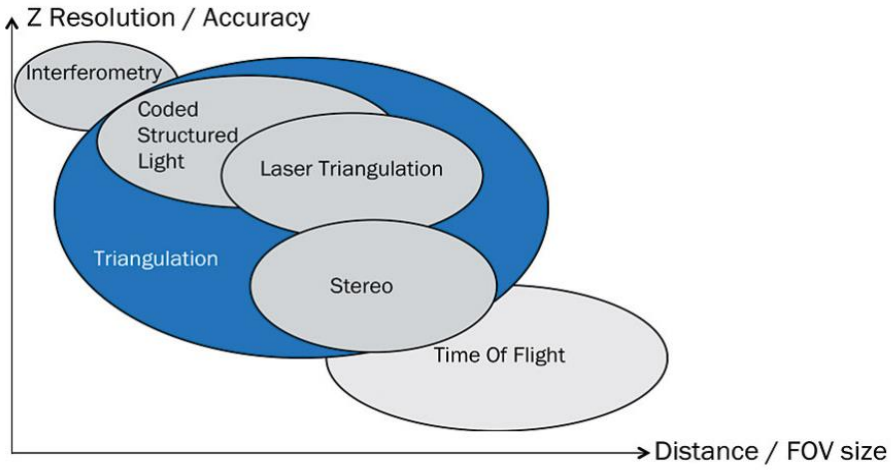


Figure 1.2 Comparison of different techniques with respect to the distance and depth resolution of different 3D imaging techniques. Triangulation and time of flight techniques cover a considerable areas in the graph. Time of flight reaches the highest distance among the whole techniques with relatively low depth resolution. Adapted from [29].

Coded structured light, laser triangulation, and stereo imaging are based on triangulation. In this thesis, we focus on coded structured light. Laser triangulation is similar to coded structured light in some cases such as it uses a camera and structured light also. However coding the screen with light results in less algorithmically complex work done with camera to the laser triangulation and stereo imaging.

1.2.LIDAR OPERATING AT INFRARED REGIME

Infrared light is an electromagnetic wave with longer wavelength as compared to visible light. It cannot be seen by human eyes. Its wavelength extends from the red edge of the visible regime at 700 nm to the 1 mm. Beyond infrared (longer wavelength regime) there is a microwave regime in electomagnetic spectrum. In the Figure 1.5 below some of usage area of different wavelengths is shown from smaller wavelength (Gamma Rays in left) to the larger (Radio Waves in right).

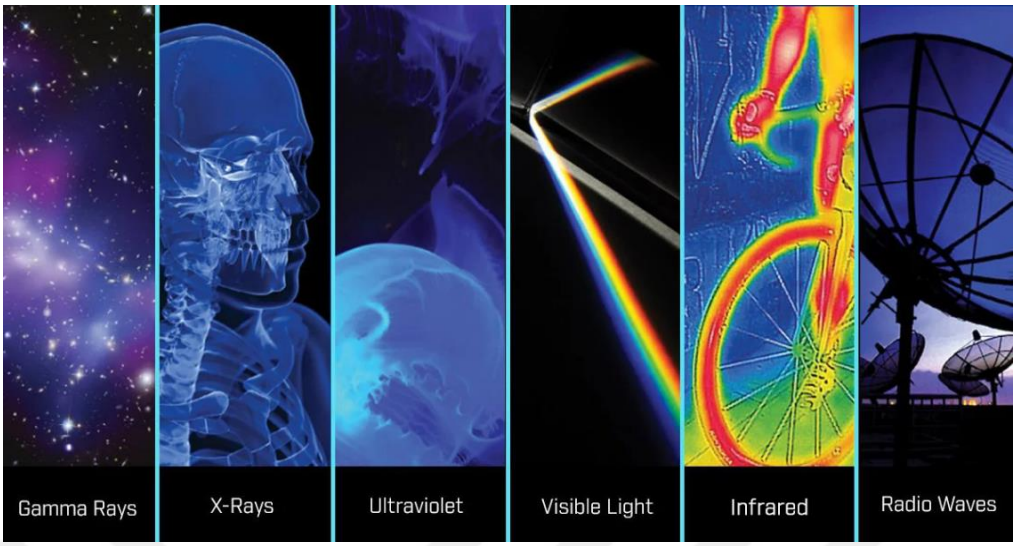


Figure 1.3 Electromagnetic Spectrum, from left to right: from shorter to the longer wavelengths. The shortest wavelength among them is Gamma rays. Gamma rays are arising from the radioactive decay of atomic nuclei. X-rays are longer than Gamma rays, X-ray is referred to as Röntgen radiation after the German scientist Wilhelm Röntgen who discovers it. Ultraviolet means beyond violet and violet is the color of the highest frequencies of visible light. Visible light is the electromagnetic radiation that is visible to the human eye. Infrared has a longer wavelength than visible light and it is capable of resolve small object for LIDAR applications. Radio Waves are electromagnetic radiation, which has a wavelength lies between meters to millimeters. Adapted from [32].

Infrared science term was not appearing until late 19th century. William Herschel first published his results in 1800. He used a refractive prism and sunlight for detecting infrared light, beyond the last visible spectrum (red) he discovered that there is an increase in temperature on the thermometer. He called his surprising result as “Calorific Rays”. Important developments in infrared science are listed chronologically below.

- 1879: Stefan–Boltzmann law formulated empirically that the power radiated by a blackbody is proportional to T^4. [59]
- 1880s and 1890s: Lord Rayleigh and Wilhelm Wien solved part of the blackbody equation, but both solutions diverged in parts of the electromagnetic spectrum. This problem was called the "Ultraviolet catastrophe and Infrared Catastrophe". [60]

- 1892: Willem Henri Julius published infrared spectra of 20 organic compounds measured with a bolometer in units of angular displacement. [61]
- 1901: Max Planck published the blackbody equation and theorem. He solved the problem by quantizing the allowable energy transitions. [62]
- 1905: Albert Einstein developed the theory of the photoelectric effect. [63]
- 1905-1908: William Coblentz published infrared spectra in units of wavelength (microns) for several chemical compounds in Investigations of Infra-Red Spectra. [64][65][66]
- 1917: Theodore Case developed the thallous sulfide detector; British scientist built the first infra-red search and track (IRST) device able to detect aircraft at a range of one mile (1.6 km).
- 1935: Lead salts – early missile guidance in World War II…

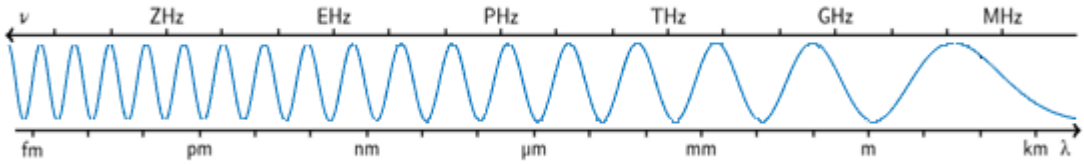
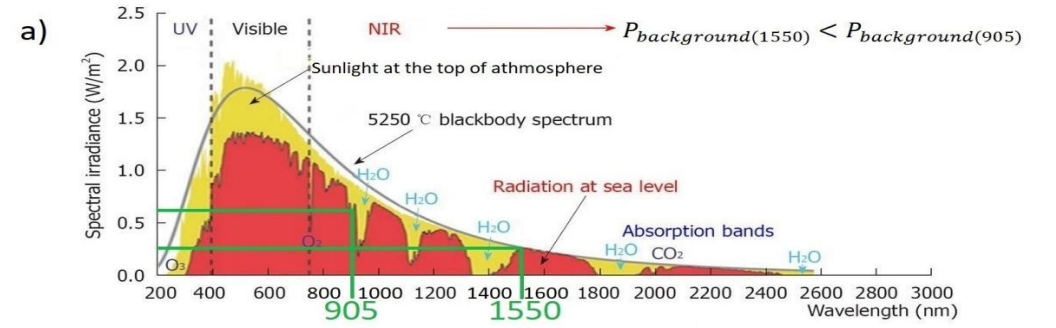
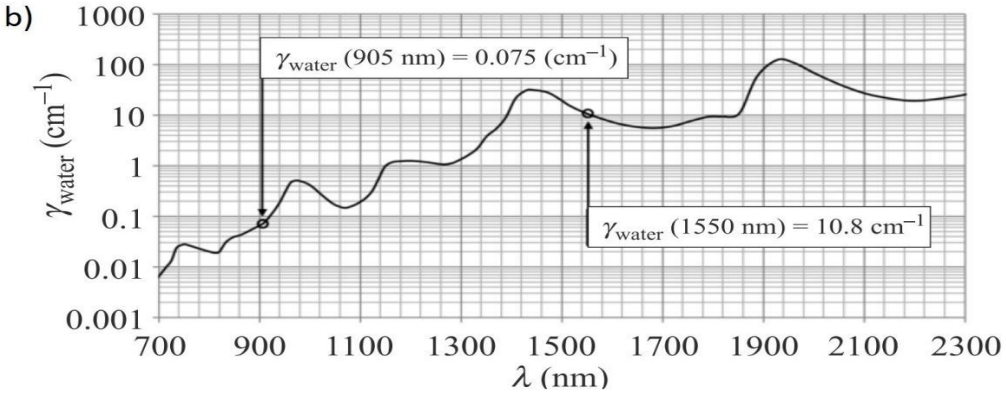


Figure 1.4 Representation of electromagnetic waves according to their frequency and wavelength.

LIDAR systems are advantageous in terms of resolution when compared to RADAR that is uses radio waves. Because LIDAR systems use shorter wavelengths, suffer less from beam divergence. On the other hand, there is a limit of usage of smaller wavelengths (such as ultraviolet) because of atmospheric scattering effects. There remains visible and infrared regime for long-range and high-resolution applications. However visible light is not suitable for everyday usage and suffers from the high solar background. There are reasons for choosing 905nm wavelength such as economic, durability of system to the environment, laser source stability etc. All of them are explained more detailed in the following.





$$H_2O_{absorption(1550)} \approx 100 * H_2O_{absorption(905)}$$

Figure 1.5 a) 5250 °C blackbody spectrum represents the solar radiation curve of Sun according to Stephen-Boltzmann Law. The y scale shows spectral irradiance in atmosphere due to sun and the x scale shows wavelength. Radiation at the top of the atmosphere is shown with yellow filled region and the radiation at sea level is shown with red filled region. Comparison of 905nm and 1550nm background radiation is given at sea level with green letter. b) H_2O Absorption of electromagnetic radiation is given in logarithmic scale. The y scale shows atmospheric attenuation coefficient the x scale shows wavelength. Captured from [30] [35].

In Figure 1.5 above a) shows the background radiation level of the atmosphere at 5250ºC (blackbody radiation curve). This is important because the background radiation is always detected by the sensor and decreases the SNR (Signal to Noise Ratio). Fig1.5. (b) show the transmission spectra of water. As can be seen the absorption coefficient of water at 1550 nm is 100 times larger when compared to

905 nm. For this reason, most LIDARs use 905 nm light in order to increase the effective range.

1.3.DIGITAL MICROMIRROR DEVICE

DMD is micro-opto-electromechanical system, see Figure 1.10 (a). It is invented by Dr. Larry Hornbeck in 1987 and developed by Texas Instrument. DMD includes millions of movable aluminum mirror can be modulated at kilohertz rate, see Figure 1.10 (b). The mirror arranged in a rectangular array named pixels can be in state ON, OFF and IDLE. Idle state's normal coincide with the whole chip alignment, ON and OFF state lie in +12 and -12 degrees to the idle state respectively as can be seen in Figure 1.10 (c). [14]. It is a robust device and used as DLP projector for many years. Although many firms produce their own projector, they all take DLP chip from Texas Instrument. DMD usage is not limited with projector technology, this is a useful tool in diverse scientific areas such as adaptive optics [15], wavefront shaping[16], compressed sensing[17] , optical tweezers [18], volumetric display[19], maskless lithography[20], holography, data storage[21] ,spectroscopy[22], microscopy and optical networking[23]. In this study, DMD is used as binary wavefront modulator and also for creating coded light.

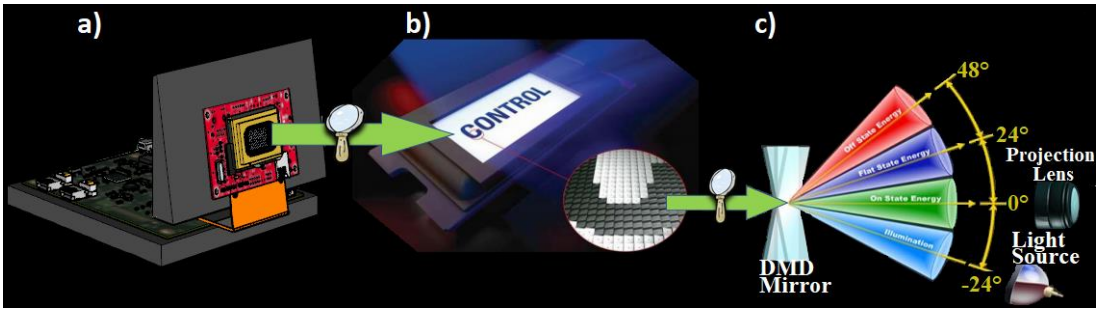


Figure 1.6 a) Digital Micro-Mirror Device with controlling board. b) Mems based movable mirror chip: black pixel shows OFF state and white pixel shows ON state. c) Single mirror position in mems chip. Light source is projected from -24°, DMD is in ON state (-12°) and reflects the light to the 0° which goes in projection lens. The IDLE state of DMD (0°) and OFF state (+12°) reflects the light to the +24° and +48° respectively. Inspired from DMD Manuel.

CHAPTER 2

OPTICAL SYSTEM DESIGN

2.1.LIDAR based on Time of flight (TOF) of laser pulses

The existence of the LIDAR principle originates from before the invention of the laser. In 1930's pioneer work is done by measuring air density profiles in the atmosphere by sensing the scattering light from the searchlight beams. In 1938, the pulsed light sources were firstly demonstrating the height of clouds[4]. The word LIDAR that employs TOF measurements was named by Middleton and Spilhaus in 1953[5]. The developments gain a huge momentum right after the invention of laser in 1960[6]. In Figure 2.1. below, TOF technique is showed.

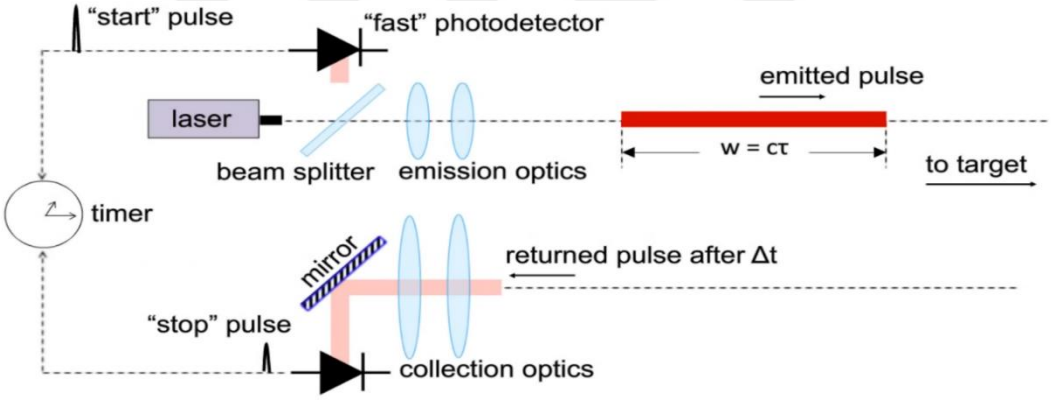


Figure 2.1 Time of flight type 3D imaging concept. Pulsed laser source emits the pulse laser beam to the beam splitter and a small portion of light starts the timer, which later calculated for the time of flight. The rest of the beam is collimated by emission optics and irradiate the target after it scatters from the target, collection optics focus the scattering light on to the photodiode and it stops the timer. Depth is reconstructed with the calculated time difference between the start and stop time of the timer. Adapted from [30].

People familiar with self-driving car technology reasonably saw rotating mechanical LIDAR on the top of these modern cars. An example of such a LIDAR

system is developed by famous Velodyne Co. Velodyne's LIDAR composed of 64 pieces of laser diode and receiver with rotating mirrors. Development in new laser systems, beam steering methods and detector technology have always been involved in LIDARs.[7] For example, development in Geiger-mode avalanche photodiode make Geiger-mode LIDAR possible: Princeton Lightwave is a newborn company invested by BMW and FORD; producing Geiger-mode LIDAR[8]. Also, the availability of frequency-modulated laser opens a new horizon for LIDAR technology. It is very similar to the chirped radar system and Blackmore Co. developed an industrial version of this system in Silicon Valley[9]. Moreover, a piece of art for beam steering as analogous of phased array antenna, large scale optical phased array (OPA) paved the way to the QUANERGY CO. for developing OPA LIDAR[10]. In this work, the pulsed time of flight LIDARs are investigated, and a new technique is shown in which DMD used as a beam steerer. Here, we develop LIDAR that is based on TOF measurements. In our approach, we use the TOF sensor together with a DMD in order to obtain a 3D map of the target scene explained more detailed in Chapter 3.

2.2.LASER RANGE FINDER

LIDAR systems based on the principle of finding the range of different points in the scene. Finding a range is simple to understand but hard to realize. Many factors affect the capability of finding the range. Some of them related to the component that is used and can be changed even with trial and error mechanism during the experiment however, environmental conditions such as time-dependent atmospheric conditions cannot be changed easily. Also, there are noises which is related to the quantum nature of light named shot noise and nothing much to with it except changing the sampling frequency. In addition, dark current fluctuations in electronics, Johnson noise, frequency and amplitude fluctuations in light source are some of the main noise sources that should be handled.

LIDAR systems based on laser range finders and range finders developed with pulsed or frequency modulated lasers. Our design based on the principle of pulsed laser time of flight calculations. To do so, selecting the laser with capable specifications are crucial. Operation wavelength is the most important parameter of laser because atmospheric conditions are time & location-dependent and vary for different operation wavelength. Conditions are changing for different region and one's laser source needs to operate well in different conditions. Another important issue is the target reflectivity. Because the atmosphere and the targets are only mediums that photons interact during the time of flight.

First, we consider the divergence of the beam. According to the Gaussian Beam Theory, there is a divergence angle for all waves depending on the aperture and the wavelength. The divergence angle formula is given by Eq. 1.In the Fraunhofer case, the Fresnel-Kirchoff relation is defined as follows:

$$U_p = C \iint \quad e^{ikr} dA \tag{1}$$

where $\boldsymbol{U_p}$ is the amplitude of wave on a diffraction plane [51].

The new expression can be derived for $\boldsymbol{U_p}$ from a circular aperture to deduce the Rayleigh criterion. Let's consider integrating over our circular aperture along the y-axis, the area element $dA = 2\sqrt{R^2 - y^2}$ and $r = r_0 + ysin\theta$. Where r_0 is the distance from the source to the aperture. $\boldsymbol{U_p}$ becomes

$$U_p = Ce^{ikr_0}\left(\int_{-R}^{R} 2e^{ikysin\theta}\sqrt{(R^2 - y^2}\, dy\right) \tag{2}$$

Let $u = \frac{y}{R}$, $\rho = kRsin\theta$, $dy = Rdu$,

$$U = Ce^{ikr_0} \int_{-1}^{1} 2\sqrt{1-u^2} e^{i\rho u} du \tag{3}$$

$$and \int_{-1}^{1} 2\sqrt{1-u^2} e^{i\rho u} du = \pi * J_1(u) \tag{4}$$

where $\boldsymbol{J_1(u)}$ is the Bessel function of the first kind.

The Rayleigh criterion tells us that "two point sources are regarded as just resolved when the principal diffraction maximum of one image coincides with the first minimum of the other"

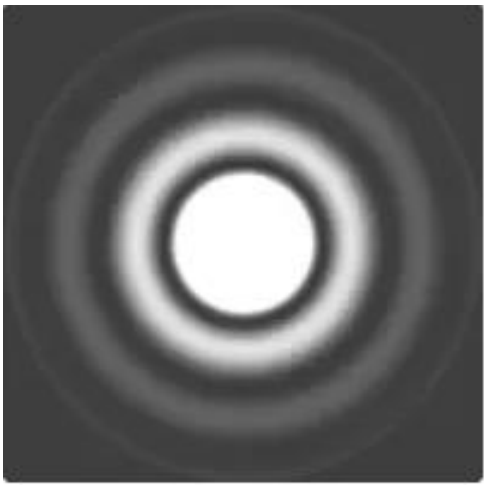


Figure 2.2 The diffraction pattern for circular aperture. Captured from[53].

And the first dark ring has an angular radius $\rho \simeq 3.832$, which corresponds to the first zero of $\boldsymbol{J_1}$. Then we get,

$$sin\theta = \frac{3.832}{kR} \simeq \frac{1.22\lambda}{D} \simeq \theta \tag{5}$$

$$k = \frac{2\pi}{\lambda}, D = 2R \tag{6}$$

$$\theta \approx \frac{1.22\lambda}{D} \tag{7}$$

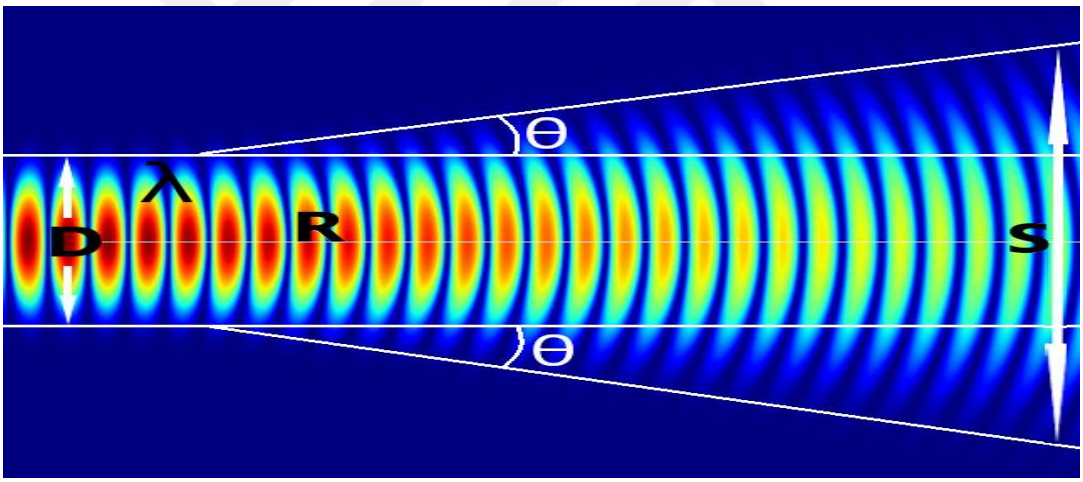


Figure 2.3 Gaussian beam expansion with θ. D is waist radius.λ is wavelength. R is distance from the waist location. θ is divergence angle. S beam radius at distance R where including intensity is $1 - e^{-2} \approx 0.865$ of total.

In Figure 2.3 divergence of ideal Gaussian beam is given. As the wave moves forward, amplitude distribution is changed with the divergence. S Minimum resolvable transverse size at distance R is given below. For comparison;

1) Radar: λ = 0.3 cm.

If D = 20 cm with divergence angle θ ≈ 1°

S ≈ 1.8 m + 0.2 m = 2 m at distance R = 100 m

2) LIDAR: λ = 1550 nm

If D = 5 mm with divergence angle θ ≈ 0.02°

S ≈ 3.7 cm @ R = 100 m

3) LIDAR: λ = 905 nm

If D = 5 mm with divergence angle θ ≈ 0.013°

S ≈ 2.2 cm @ R = 100 m

According to the values calculated above small wavelength should be chosen for small spot resulted in high resolution. 905 nm have an advantage over 1550nm. On the other hand, there are many factors that affect selection of wavelength such as cost, stability, target reflectivity, atmospheric conditions.

2.3.POWER ON DETECTOR

As well as resolution, power falls on detector is another important issue for 3D imaging. We use a pulsed time of flight technique in our LIDAR system and the power falls on its detector is given by the equation(8) below by considering the general parameter. In here, athmospheric effects has an important role for calculating the power, however, calculating and implementing it the equation is another topic that is not mentioned in this thesis.

P(R) = Optical power returned from the scene. The lowest possible resolvable level for signal to noise ratio gives us maximum measurable distance. And optical power returned from the target is given by;

$$P(R) = P_0 \rho \frac{A_0}{\pi R^2} n_0 \; exp \; exp \; (-2\gamma R) \qquad (8)[39]$$

P_0 = Peak output power of pulsed laser source,
ρ = Reflectivity coefficient of target,

A_0 = Lens aperture are in front of detector,
γ = Atmospheric attenuation coefficient,
n_0 = Receiving optics spectral transmission,
R = Target distance.

Equation (8) above is valid only if the atmospheric attenuation is nonlocal, laser spot interact with certain type of target (in spot point reflectivity remains the same), target is Lambertian, target surface normal is parallel with measurement path. It needs to be considered that change in atmospheric attenuation (also reflectivity of target) also affects the minimum detectable power. Different conditions mentioned as given in above considered by J.Wojnatowski et all and discussed briefly;

In Figure 1.5 it is easily notified that the difference between attenuation coefficients of two wavelengths is huge. That means water presence in the environment impacts much higher to 1550 nm than in 905nm. The stability of performance and cost of the detection unit is often concerned by many manufacturers. For this reason, they use 905 nm instead of using 1550 nm.

By concluding all results above, the essential property is the maximum range without any doubt. Manufacturers always give a value of maximum range under perfect visibility conditions. On the other hand, real life applications are not always in the best scenario. For example, water content has a major impact on performance of 1550nm. As different, 905nm is affected by water less than two order of magnitude. One more important factor is cost of the detection unit. 905nm rangefinders uses generally low-cost Silicon APD than rangefinders with 1550nm which uses InGaAs APD. real-lifeFor this reason, a platform based on 905 nm source has more advantages when compared to the units that operate at 1550 nm.

2.4. AVALANCHE PHOTODIODE & GEIGER MODE PHOTODIODE

LIDAR is generally a low light application since the backscattered light is detected. For this reason, a high intrinsic gain in detection unit is required detection of signal. More recent studies show that there is an alternative for APD, called Geiger mode Photodiode. APD gain is increased linearly with the increase of electric field. On the other hand, as analogous with the Silicon photomultiplier tube, Geiger mode

photon counter is not working linearly. They have a dynamic range that is related to the pixel number.

In the pulsed LIDAR system signal is not used directly for imaging. Signal is binarized and thresholded for calculating range data. Because of the high gain, it is advantageous for LIDAR system to use Geiger mode photodiodes under very low light conditions. In figure 2.4 we see the importance of optical gain when electronic readout noise occurs.

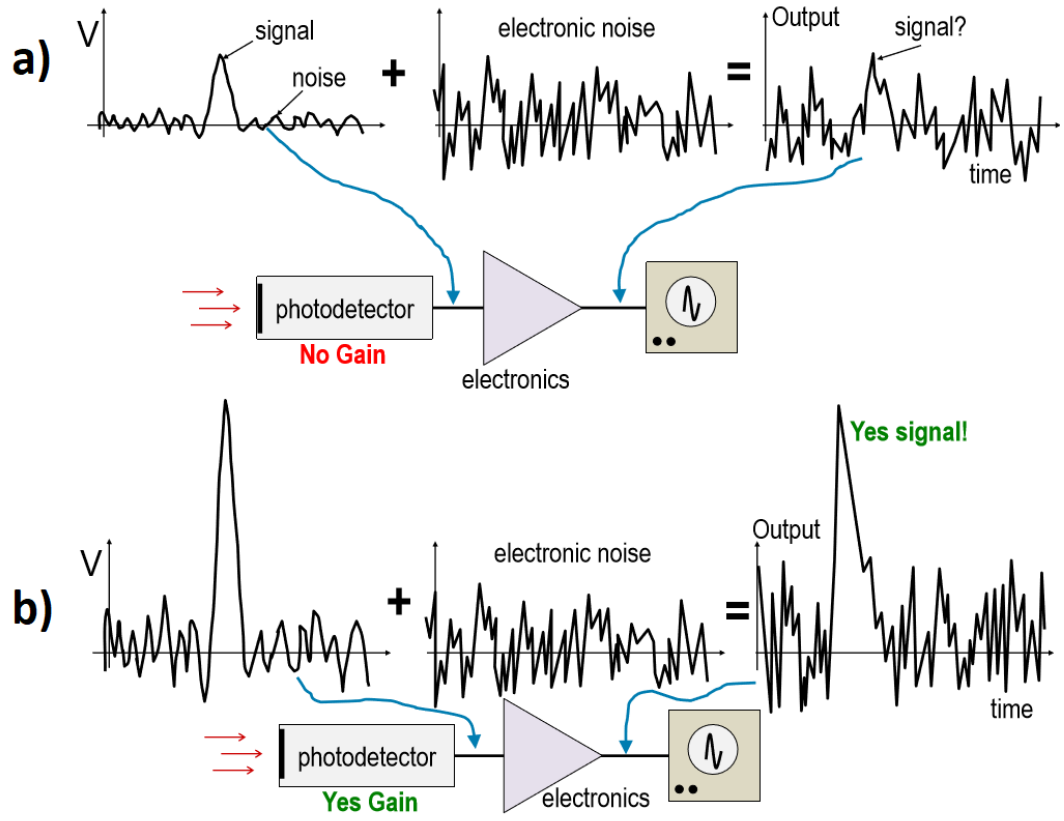


Figure 2.4 a) Signal and electronic noise added together without gain. Resulted signal shows that initial photo detector signal is lost when electronic noise level is comparable with optical signal. b) Signal with gain added to the electronic noise and resulted signal shows that electronic noise can be eliminated by appropriate gain level. Adapted from [30].

The demand for high gain is increasing every day for low light level applications. Increasing demand in this area results in the technological development of high gain photodiode. Avalanche photodiode and Geiger mode photodiode are

examples of high intrinsic gain photodiodes. In a conventional photodiode, incident photons are absorbed by the junction and create electron-hole pairs. Under reverse bias condition, mainly dark current and diffusion currents are suppressed.

The gain mechanism of Geiger-Mode type diode is impact ionization. In this phenomenon high reverse bias voltage is applied and result in high **E** field in junction. High **E** field accelerates the electron in junction and electrons that collide with electron hole pair and creates new electron participate in the photocurrent. In Figure 2.5 avalanche gain is showed.

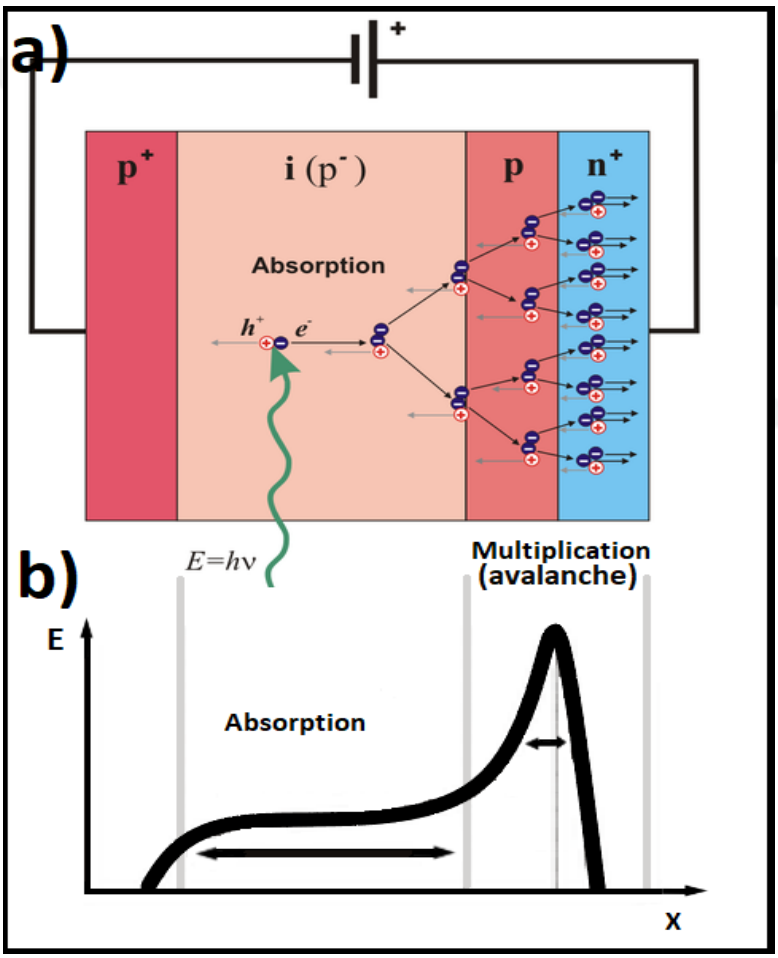


Figure 2.5 Gain mechanism: Avalanche multiplication. a)APD junction design. Light absorbed by the junction creates electron-hole pair. Under high reverse bias voltage electrons are accelerated so that their kinetic energy level increases and impact ionization occurs. Same process goes on after impact ionization and this

phenomenon named as avalanche multiplication. b)Electric field in junction. Inspired from [45].

As mentioned before APD works linearly but with low gain. Different from APD, Geiger mode photodiodes work under condition $V_{bias} > V_{breakdown}$. Slightly higher reverse bias than breakdown voltage of make junction works in Geiger mode and reaches gain level $\sim 10^6$. Low noise level Geiger mode photodiodes first manufactured by Hamamatsu [30] named with Multi Pixel Photon Counter (MPPC). The fundamental activity of that MPPC can be clarified in three modes as charge, release and quenching. [40] It uses a self-quenching resistor for not going too over from the Breakdown Voltage level. Even it is analogous of photomultiplier tube, it has an advantage over PMT which relatively low bias level is. This kind of junction also resistant to the external magnetic field and has very high gain. This junction called MPPC is not a mature technology, however, the society accepts this as a robust technology. [41] When the photon incidence on junction avalanche process start in MPPC, and the current will start to stream. The presence of extinguishing quenching resistor limits the MPPC and decrease $V_{op} - V_{br}$ and avalanche process stops. The working scheme is given in Figure 2.6

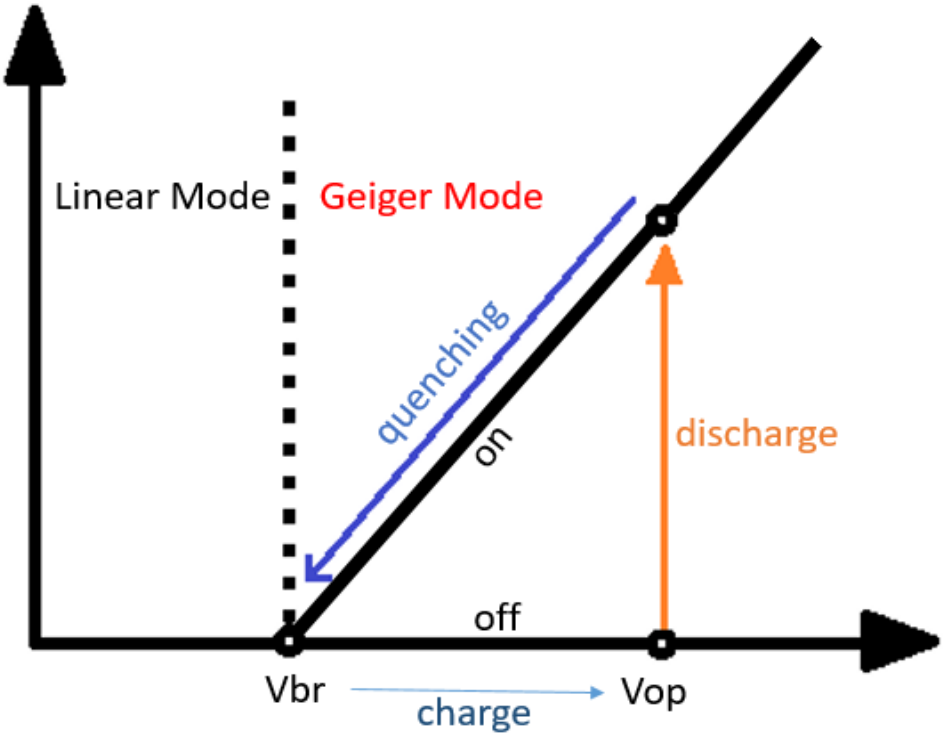


Figure 2.6 The lifecycle of the avalanche process in Geiger mode. The difference between linear mode and Geiger mode is the bias voltage level. If Vbias lower than the breakdown voltage junction works in the linear regime. If it is higher than breakdown, junction works in Geiger regime. The junction is charged to the operating voltage which is above the breakdown level. Photons start the avalanche process and quenching starts to decrease voltage level for stopping avalanche process for the next cycle.

The working principle of APD and Geiger-mode APD are mostly the same. The difference is the bias voltage level. Geiger-mode APD uses a bias with above the breakdown voltage of the junction, on the other hand, APD uses a bias with below the breakdown voltage. Both need a passive quenching resistor to stop the avalanche process. In both, most of the dark current occurs because of the thermally generated carriers and cooling is an option to suppress the noise. High gain (10^6) is achieved in Geiger mode and that value makes single-photon level detection possible. These features make MPPC suitable for many applications such as MRI-PET, TOF-PET, high energy physics calorimeter, astronomy, fluorescence measurement, DNA BIO-chip sequencer. [41][42][43]

2.5.TIME TO DIGITAL CONVERTER

Measurements of very short time intervals (~ps) is an important requirement in RADAR&LIDAR systems. Many such systems, in their distance calculations, depend heavily on time interval resolutions between transmission and reception of certain signals. Our LIDAR system is based on the time of flight principle in which a light beam is sent to a target object by a pulsed laser source and scattered back to some receiver. Since the velocity of light is known, the time difference between the transmission of the original pulse and the reception of scattered light may be used to calculate depth. A Time-to-Digital Converter (TDC) with high precision is used to digitize the time difference between two pulses. Figure 2.7 shows a high-level schematic of an ordinary laser range finder.

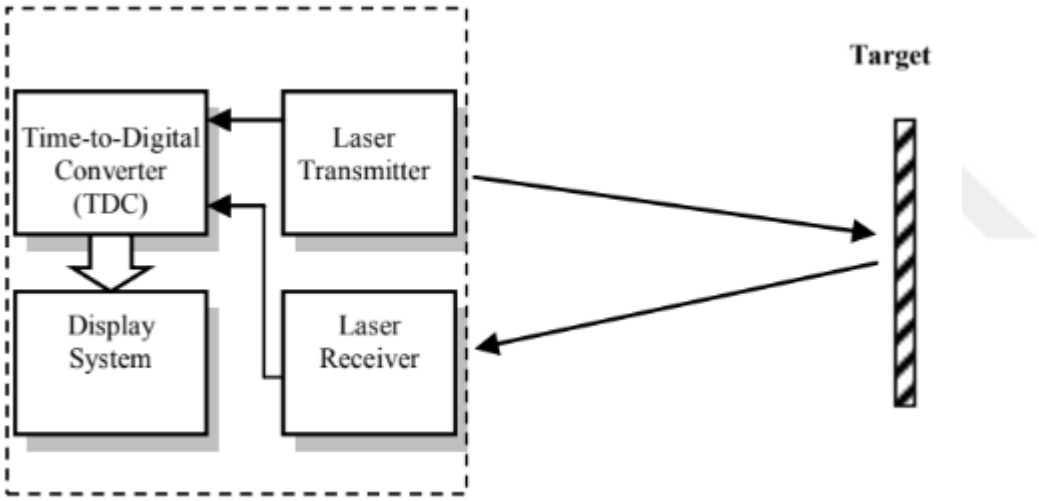


Figure 2.7 High-level schematic of a laser range finder. A light pulse is transmitted from a Laser Transmitter to some target object, then, a Laser Receiver is used to intercept the pulse scattered by the object. Signals are sent to a Time-to-Digital converter (TDC) upon transmission and interception of the pulse respectively. Time difference between these signals is then digitized by the TDC to be displayed or used in further calculations. Captured from [44].

When measuring the time difference between two signals with high precision, TDC's are the only inexpensive option. For example, achieving 1ns resolution using

an Analog-to-Digital Converter (ADC) would require a GHZ clock rate and such ADCs are around one hundred times more expensive than TDCs. The time interval measurement of a TDC is triggered to start with the emitted signal and stop with the backscattered signal. TDCs are thresholded based on incoming signal level to eliminate background noise.

Simply put, a TDC is a high frequency counter where the count starts with a trigger, increases with every clock cycle and stops with a trigger. When the counter is triggered to stop, the number in the counter multiplied by the length of a clock cycle gives us the time interval. Consequently, in such a TDC, the time interval is quantized in clock cycles. In order to achieve higher precision, a smaller clock cycle length is required. An alternative method is developeded to achieve higher precision without having to decrease clock cycle length.

As explained above, the time interval measurement of a TDC using a standard counter depends on the number of clock cycles. Thus, the measurement error is determined by quantization error and resolution is limited by the clock rate. In the alternative method, the idea is to overcome this limitation using a Vernier method with two controllable, slightly different frequencies $T_{slow}\ and\ T_{fast}$. This method employs two clocks, one slow and one fast, and a phase detector to detect coincidence between the two clocks. The difference between the frequencies $T_{slow}\ and\ T_{fast}$ replaces clock cycles as the time step.

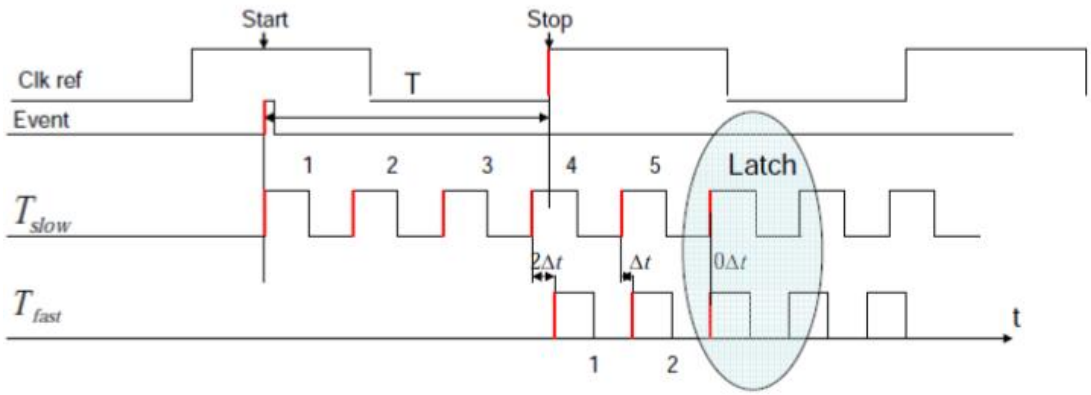


Figure 2.8 The Vernier Method two-step TDC as explained in [44]. The slow clock is triggered by the start event and the fast clock is triggered by the stop event. Counter counts both clocks until a latch occurs, providing higher time resolution. Captured from [44].

The two-step Vernier method for TDCs is shown in Figure 2.11. The start event triggers the slow clock and the stop event triggers the fast clock. A phase detector is used to detect coincidence between the two clocks. Theoretically, using this method, TDC resolution can reach the following level:

$$r = T_{fast} - T_{slow}. \quad (9)[44]$$

We have discussed the basics of TDCs and their significance to our work. It should be noted that there are different types of TDCs besides the basic one-step TDC and the Vernier two-step TDC explained in this section. Also, many different algorithms have been developed to increase time resolution. However, the Vernier two-step TDC is found suitable for many applications due to its relatively low price and complexity as well as high resolution. Further details and in depth comparative analyses are omitted as they are beyond the scope of this study.

2.6.BEAM EXPANDER DESIGN WITH ZEMAX

In ZEMAX, ray tracing is a simulation method for an optical beam which preserves its wave properties. There are two modes in ZEMAX, namely, “sequential” and “non-sequential”. Sequential mode is generally used for lens design in order defined surfaces. Whereas non-sequential mode is used for simulation and analysis. In non-sequential mode, the ray is traced independently of the surface order. This mode requires more processing power as it simulates all possible orders. We design the beam expander in our LIDAR with DMD system using the sequential mode of ZEMAX.

In our design, we use 905nm monochromatic light. It is decided that 905nm is more stable under different environmental conditions and Silicon APDs can be used for high gain detection with low cost. A LIDAR system designed at 905nm can operate under different conditions. LIDAR systems working at 1550nm require design changes depending on operating conditions and work with expensive detectors. Our design uses a 905nm laser, the complete table of ZEMAX parameters For angular magnification calculation, a small deviation from the optical axis is required in ZEMAX. The deviation is determined to be 0.001 degrees. Table 2.1 shows the "Field Data" window used in ZEMAX to set deviation from the optical axis.

Table 2.1 The tilt between the optical axis and the incoming wave. It must be non-zero for angular magnification calculation (this is specific to the ZEMAX).

Field Data

Type	Angle				
Field Normalization	Radial	-	-	-	-
Use	X-Field	Y-Field	Weight	VDX	VDY
1	0	0.001	1	0	0

In ZEMAX, the diameter of an aperture is set to the corresponding “aperture value". Taking the DMD dimensions into consideration, 10 millimeters is chosen as the aperture value for our system. Table 2.2 shows the "General" window in ZEMAX (all units are in mm).

Table 2.2 General settings in ZEMAX. Incoming beam parameter is in millimeters (appropriate for the DMD dimensions ~10mm). Image space has to be chosen "Afocal" when designing the beam expander.

General Settings

Aperture Type	Pupil Diameter
Aperture Value(mm)	10
Apodization Type	Uniform
Image Space	Afocal

The Merit Function Editor is an optimization tool in ZEMAX. In order to achieve 3x angular magnification, we added an AMAG (angular magnification) Merit Function using the editor. Table 2.3 shows the Merit Function Editor in ZEMAX. 3x magnification is targeted with "0.334" under the "Target" column.

Table 2.3 Merit Function Editor in ZEMAX, AMAG stands for angular magnification. Target is set to 0.334 which corresponds to 3X magnification.

Merit Function Editor

	Operation	Type	Target	Weight	Value	%Contribution
1	BLNK	BLNK	-	-	-	-
2	AMAG	AMAG	0.334	1	0.334	100

A special lens can be designed by changing the radius parameter into a variable. However, since it is difficult to produce special lenses, we designed our beam expander using commercially available lenses from THORLABS. For an appropriate infrared design, B coated (650-1050nm) lenses are used.

Table 2.4 THORLABS lenses LC1715-B and LA1417-B are chosen with variable distance between them.

Lens Data
Editor

	Surface: Type	Comment	Radius	Thickness	Glass	Semi-Diameter
Object						
1	Standard		Infinity	20		20
2	Standard	LC1715-B	-25.730	3.5	**BK-7**	12.7
3	Standard		Infinity	98.788(V)		12.7
4	Standard	LA1417-B	77.260	7.290	**BK-7**	25.4
5	Standard		Infinity	100		25.4

Designs created in ZEMAX can be visualized using tools such as “design layout” and “3D shaded model”. Figure 2.9 shows the ZEMAX design layout and Figure 2.10 shows the 3D shaded model in sequential mode.

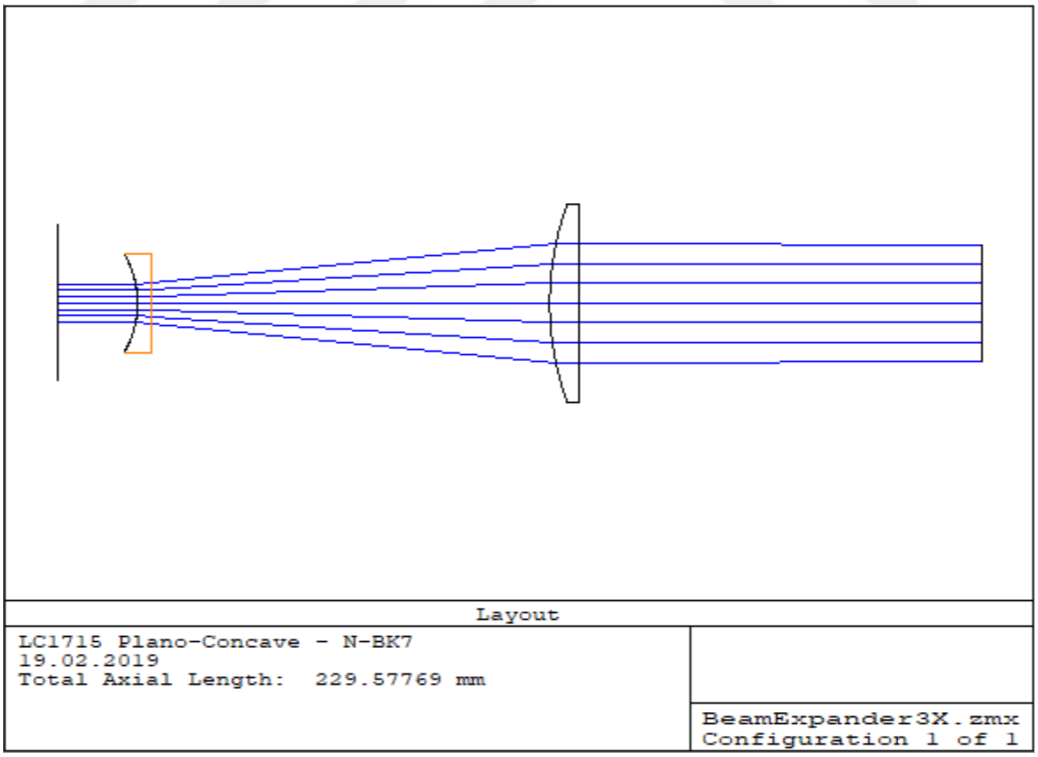


Figure 2.9. 2D layout of the 3X beam expander design.

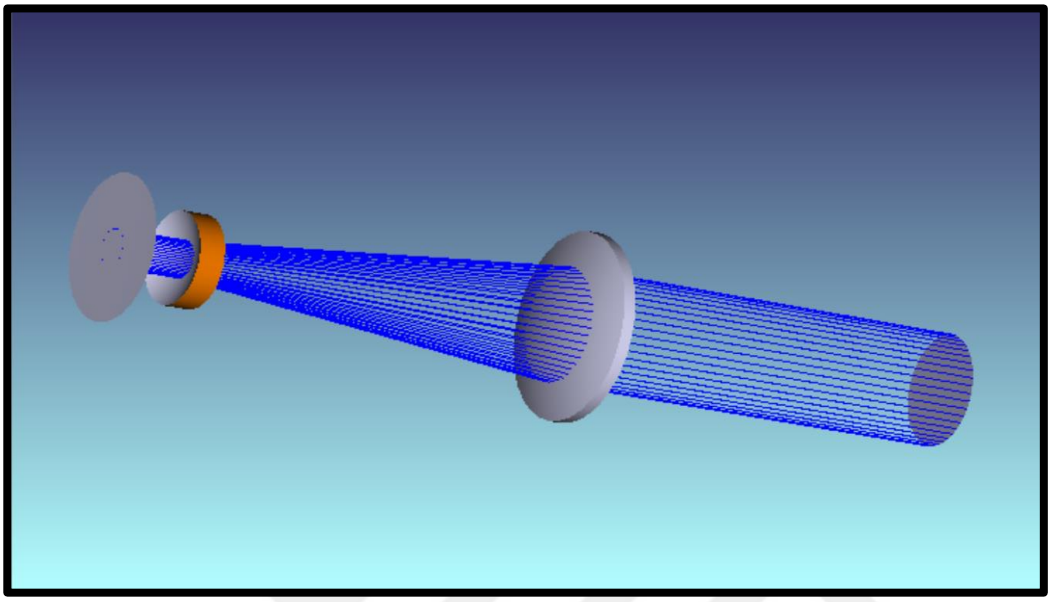


Figure 2.10 3D layout of the 3x beam expander design.

The beam expander is used after Fourier filtering of the output light from the DMD as can be seen in Figure 3.9. The THORLABS code and related technical specifications of the lenses are provided in the table below.

Table 2.5 Lens data from THORLABS.

Lens Name	Diameter	Focal Length	Diopter	Radius of Curvature	Center Thickness	Edge Thickness	Back Focal Length
LC1715-B	1”	-50mm	-20	-25.7mm	3.5mm	6.9mm	-52.3mm
LA1417-B	2”	150mm	+6.7	77.3mm	7.3mm	3.0mm	144.7mm

After simulating our lenses in ZEMAX, we conclude our beam expander design. We orientate our lenses so that 97.8mm spacing between them for achieve 3x angular magnification. We need 3x angular magnification for increasing object space. Our

purpose is to resolve 5cm difference between two object that is shown in Figure.3.12. By doing so we used 1920*1080 pixels DMD in front of the beam expander.

2.7.STRUCTURED LIGHT and THE TRIANGULATION METHOD

Triangulation techniques are known to have been used by ancient Babylonians and Egyptians as early as 5,000 years ago. Euclid and Archimedes laid the mathematical foundations of trigonometry about 2,500 years ago and Snell von Rojen studied optical triangulation laws in the 17th century. Practical 3D scanners based on the triangulation principle were developed almost 25 years ago [50]. With the help of state-of-the-art structured light, camera and image processing technologies [11], triangulation techniques have now reached depth resolutions in the micrometers.

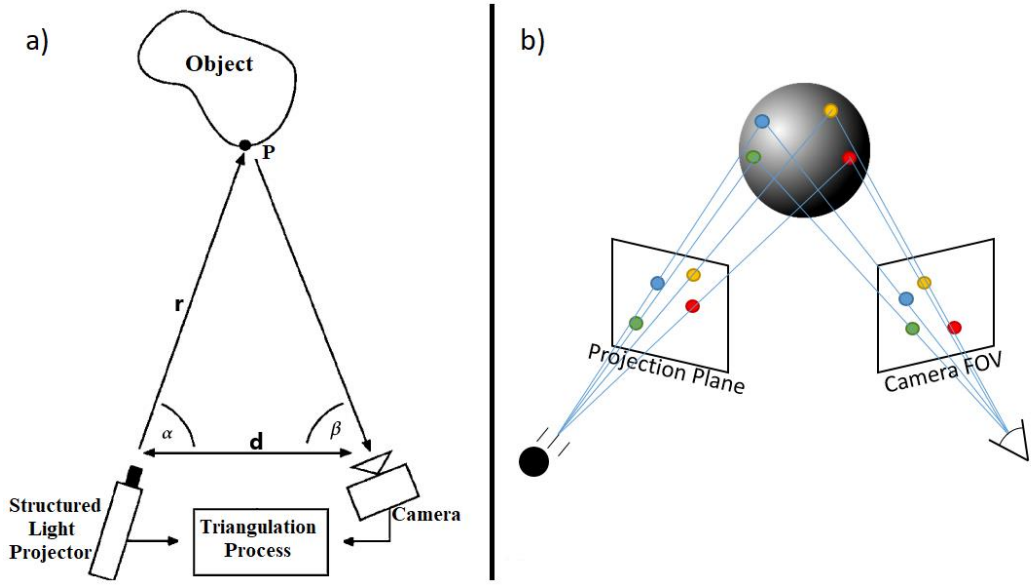


Figure 2.11 a) The basic triangulation principle (angle-angle-side). Some structured light pattern from the projector illuminates the object and its image is captured by the camera. Depth can be reconstructed given the distance "d", the angle "α"and the angle "β". b) 3D representation of the structured light projector and camera system in a).

In the context of 3D imaging systems, triangulation is the process of determining the depth of a point on some target object by forming triangles to it from

a set of known points. Triangulation is a tool for solving triangle. To demonstrate, in Figure 2.12 a), the point P on the target object corresponds to different pixel coordinates in the Projection Plane and the Camera Plane. This difference in pixel coordinates is used to compute the angles α and β. Then, the angles, as well as the known distance d between the camera and the projector, are used to calculate the distance r between the projector and the point P. Further, Figure 2.12 b) provides a 3D visualization of this process. Rays that intercept the Projection Plane are drawn between the projector and each colored point on the sphere. Similarly, rays that intercept the Camera Plane are drawn between the camera and each colored point on the sphere. The two rays corresponding to some target point intercept their respective planes at different angles. These angles are computed and used in triangulation to construct the depth of the target point. By repeatedly finding depths for different points on the object using triangulation, the entire sphere can be modeled in 3D.

Triangulation is a useful technique currently used in many different fields to solve a diverse range of problems. However, the roots of triangulation go far deeper than state-of-the-art technologies of our day or even theorems established 500 years before the beginning of our calendar. Indeed, it is possible to observe the use of triangulation in many natural organisms as well. For instance, the human visual system is, in its simplest form, a stereo vision system which uses triangulation to perceive depth. The distance between the two eyes is known, and the rest is image processing in our brain for 3D sensing. It can be seen that a system analogous to our own visual system can be built by simply exchanging the projector in the sytem from Figure 2.12 with a second camera.

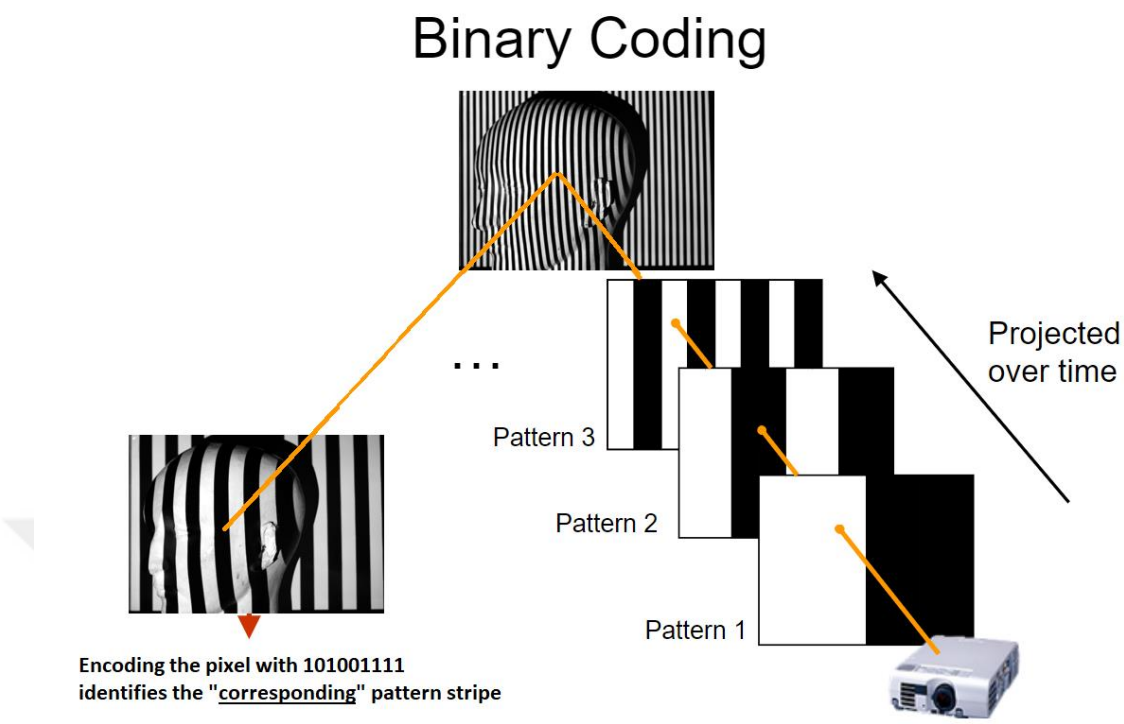


Figure 2.12 Encoding of a scene using binary structured light; structured patterns consisting of alternating black and white vertical stripes are projected on a mannequin and the resulting scenes are captured by a camera. Points on the mannequin are encoded based on light intensity for each pattern for further processing.

Figure 2.12 shows a real life application of the triangulation process described in Figure 2.11 a) which uses a method called Binary Coding to calculate the angles α and β. In this method, different light patterns, each consisting of a gradually increasing number of vertical stripes alternating between black and white, are repeatedly projected onto the scene. In each pattern, any given point in the scene will correspond to either a white -encoded with a 0- or a black -encoded with a 1- stripe. Therefore, any point in the scene can be encoded by a bitsring whose length is equal to the number of different patterns. Each such bitstring is assigned predetermined angle values. In summary, in the Binary Coding method for depth estimation, relevant points in some scene are encoded by bitstrings, then the encodings are converted into angles and finally the angles are used in triangulation to calculate depth.

Many argue that development of technologies similar to our natural faculties is due to our baser instinct of copying ourselves. It follows from this argument that we should develop 3D imaging systems as indeed depth perception and 3D vision are undoubtedly some of our foremost sensory capabilities. It should be no surprise then, that 3D imaging systems have been researched and developed with ever increasing speed and motivation and adapted by many different applications. Apple's Face ID is one of the most popular examples of our day and uses the principles discussed in this section with infrared dots as structured light patterns [12]. Similar technologies are used in Microsoft's Kinect and HoloLens to sense the 3D world [13]. Besides these popular brands, 3D imaging systems are widely used in automated robots, driverless cars, micrometer resolution dental imaging systems and countless other applications in the current industry trend.

2.8. IMAGE PROCESSING

Digital image processing is a multidisciplinary field which involves mathematics and informatics as well as computer science and engineering. It can be defined as the analysis, synthesis and modification of digital images through the use of intelligently designed algorithms. Various digital image processing techniques are used to detect interesting components and patterns such as edges, corners or more complex shapes in digital images. Additional applications of digital image processing may be listed as noise removal, texture mapping, image filtering, increasing storage efficieny and many others. In this work, we make use of noise removal, adaptive thresholding and texture mapping techniques.

In image processing, "noise" is a subjective term defined as an undesirable pattern or anomaly in an image. Thus, noise removal is the process of clearing undesirable patterns or anomalies from an image. Many noise removal techniques have been developed over the years. However, due to the nature of the task at hand, no highly successful generalized noise removal method exists. Existing successful methods are specialized for certain noise types. Therefore, before deciding on a noise

removal technique, it is important to classify noise type. It should also be noted that success or failure of a noise removal algorithm is difficult to quantify due to the main criteria being “having a pleasant appearance”.

We picked our noise type to be “salt and pepper”, which is encountered in the form of randomly scattered pixels or small sets of pixels with sharp color differences with their neighbors, as it is seems to be the most appropriate for our issue [24]. Figures 2.13.b and 2.14.b are examples of images with salt and pepper noise.

Basic smoothing algorithms attempt to make images smoother by applying filters that modify pixel values based on neighboring pixels. We observe that smoothing algorithms are not useful for removal of salt and pepper noise as seen in Figures 2.13.c, 2.13.d, 2.14.c and 2.14.d. In contrast, median filtering, which is a non-linear digital filtering technique that preserves edges while removing noise, is found to perform reasonably well as can be seen in Figures 2.13.e and 2.13.f. It can also be observed from Figure 2.14, that median filtering removes salt and pepper noise with increasing success as kernel size increases. However, an increase in kernel size also results in a more blurry image, creating a trade-off. Therefore, picking the right kernel size is an important factor when using a median filter for salt and pepper noise removal.

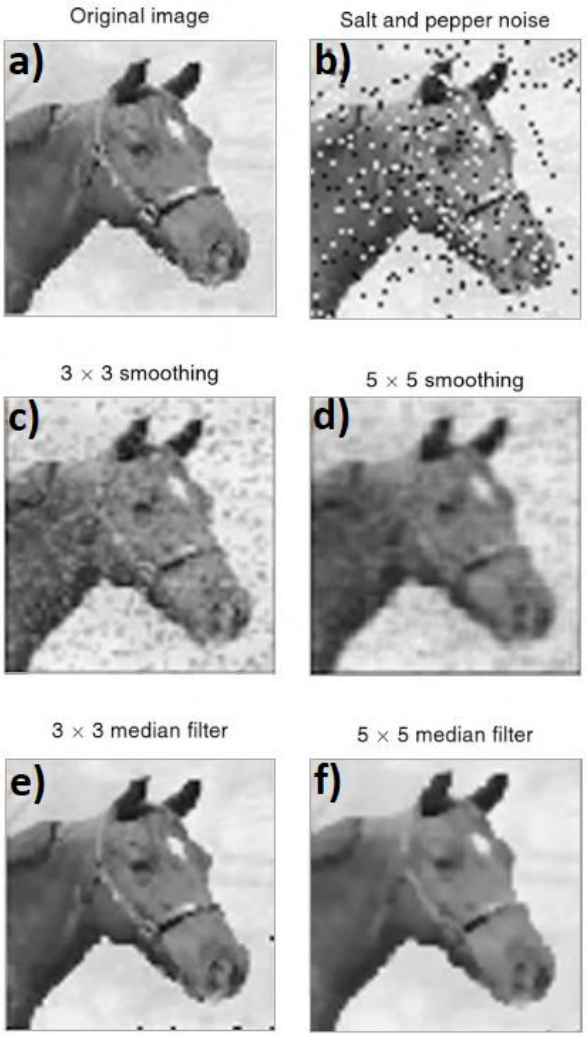


Figure 2.13 a) Original image. b) Original image with salt and pepper noise. c) Noisy image after smoothing using an average filter with a 3x3 kernel. d) Noisy image after smoothing using an average filter with a 5x5 kernel. e) Noisy image after median filtering with a 3x3 kernel. f) Noisy image after median filtering with a 5x5 kernel. Taken from [36].

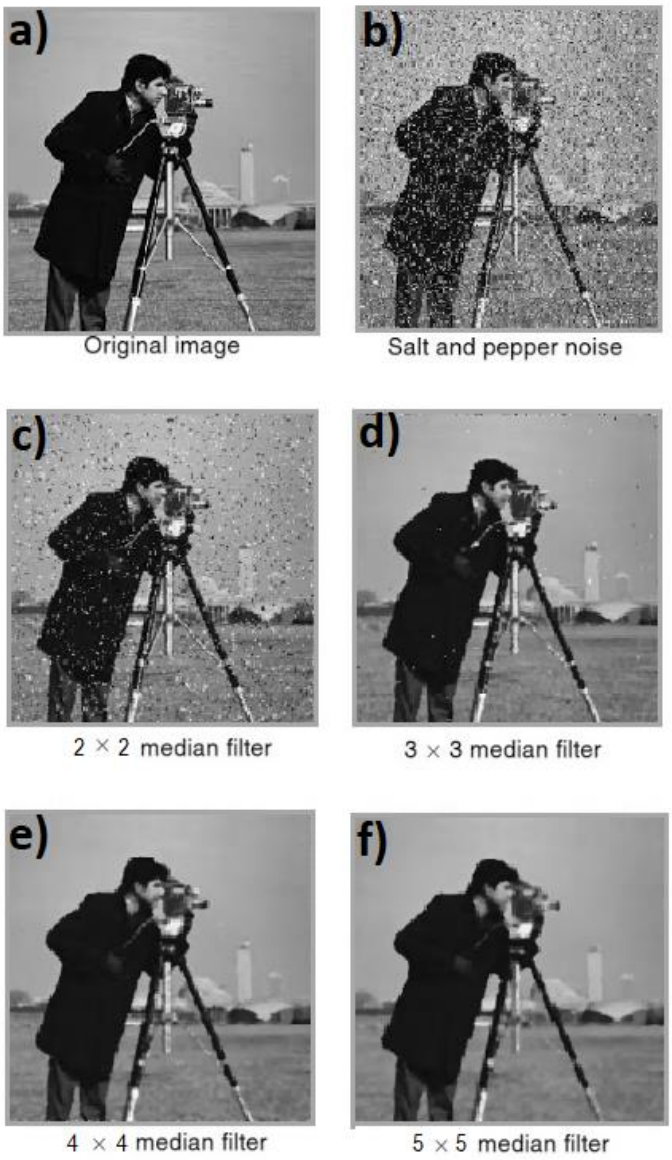


Figure 2.14 The original image, a). The original image with salt and pepper noise, b). The noisy image after median filtering with a 2x2 kernel (c), 3x3 kernel (d), 4x4 kernel (e) and 5x5 kernel (f). Taken from [36].

In digital image processing, thresholding is a simple method of image segmentation, or the partitioning of an image into multiple segments. Image segmentation is widely used to extract and highlight useful information in images. Adaptive thresholding is another digital image processing method in which an image

is divided into several parts and different thresholding techniques are applied to each part separately. It is shown that adaptive thresholding is suitable for highlighting and extracting information from images with varying brightness and low contrast [25]. Figure 2.15 provides examples of different thresholding techniques applied on an image. It can be seen in 2.15.a that the original image has gradually increasing brightness from left to right. Figure 2.15.b shows that when simple hard thresholding is applied to the image, almost half of the information in the image is lost. Hard thresholding is a method in which pixels are assigned black or white values based on some predetermined threshold. However, when an adaptive thresholding method is used, where the image is divided into segments and each segment is thresholded based on a different value, a more complete and readable result is achieved in Figure 2.15.c.

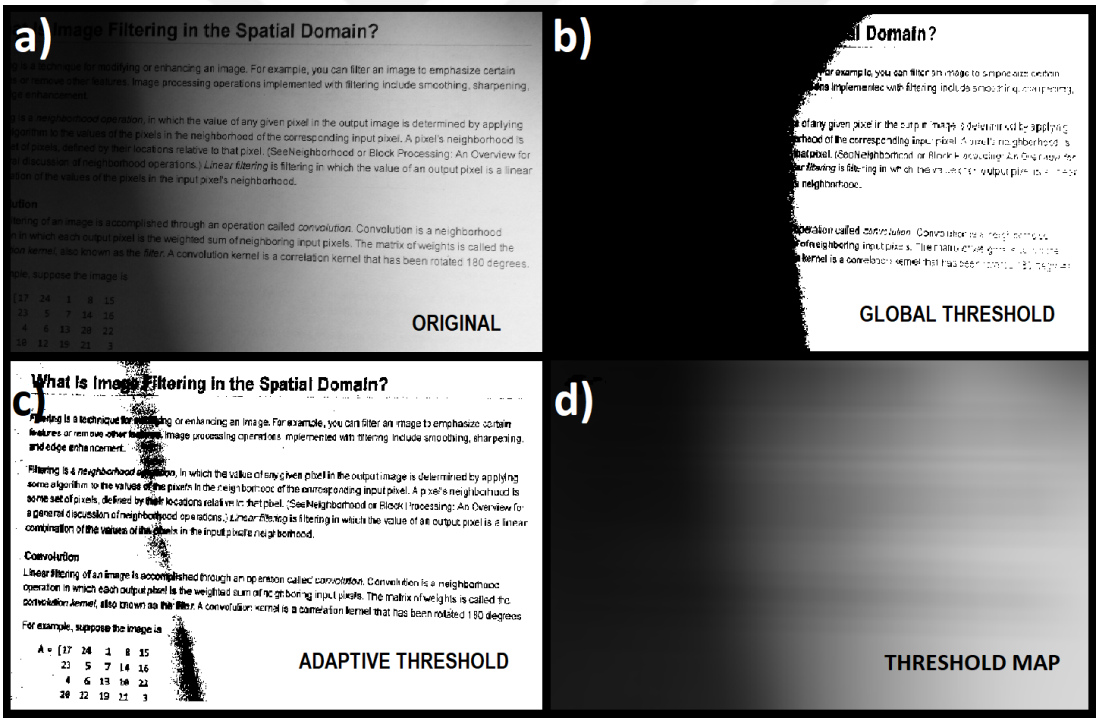


Figure 2.15 a) The original image. b) A global hard threshold is used to convert the image into binary. c) An adaptive thresholding method is used to construct the image with minimum loss. d) The shading level of the original image used by adaptive thresholding to set threshold levels for different parts of the image. Inspired from[54].

Texture mapping, in digital image processing and graphic design, is a process in which a 2D image, called a surface map, is wrapped around a 3D object. Creating more detailed and complete 3D point cloud data by adding color information is an increasingly demanded task for countless applications. One of the widely used techniques for creating textured 3D models works by adding pixelwise depth information to a 2D image and wrapping it on to a point cloud [26]. Figure 2.16 provides an example for this method. First a 3D point cloud of the target object is created using a LIDAR sensor and meshed to close off small holes on any of the surfaces. Then, color information captured from the same angle of view is used to wrap the point cloud and create the texture-mapped 3D model seen below. We make use of this technique in this thesis and our results are provided in Chapter 4.

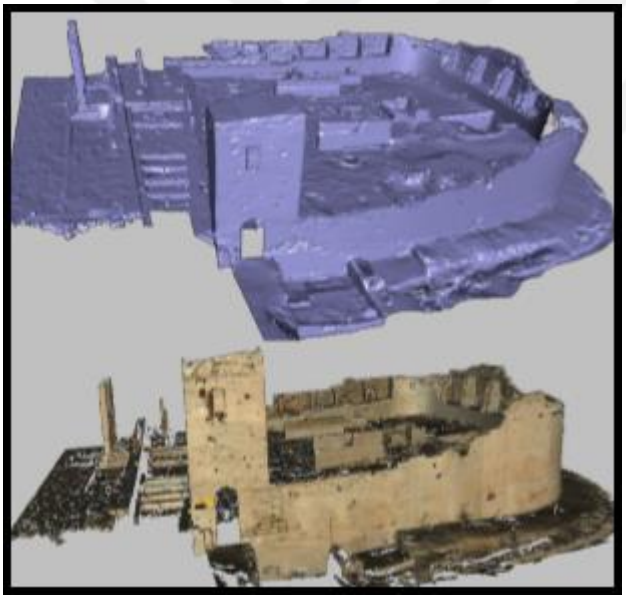


Figure 2.16 The image above shows the 3D point cloud of a scene. A color camera is used to capture the color information of the scene from the same FOV. Then, the color information is mapped onto the 3D point cloud, resulting in the image below. Taken from [33].

CHAPTER 3

LIDAR SYSTEMS (TIME OF FLIGHT)

3.1. TIME OF FLIGHT CALCULATIONS

Time of flight LIDAR systems are mainly divided into two groups. One of them uses a pulsed laser source and the other uses frequency modulated continuous waves. Our study is based on the pulse principle. In this method; short laser pulses transmit through the target and are reflected back. An ADC can be used as a decoder and depth information can be constructed from the time difference between the sent and reflected pulses. This principle is shown in Figure 3.1.

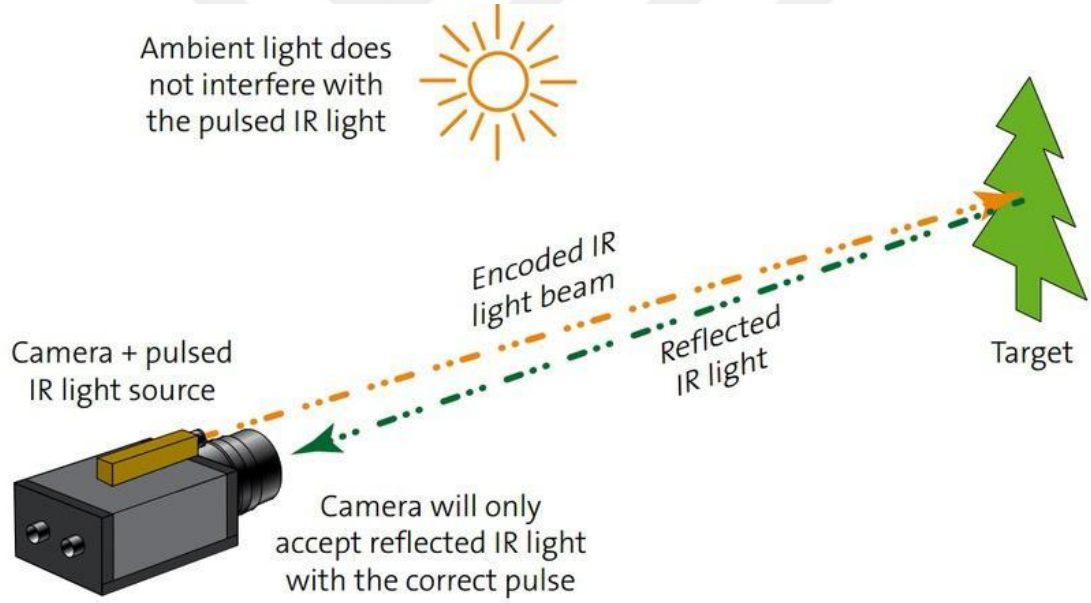


Figure 3.1 Schematic diagram of a typical laser rangefinder. A coded infrared pulse is transmitted to the target and the camera detects the back scattering signal from the object. A timer is triggered to start with the initial pulse and stop when the reflected pulse is sensed by the detector. Here, the detector is a camera as a substitute for an APD array. Captured From [55]

As a similar principle to the figure above, we start our systems by measuring distance. In order to do so, we use two silicon detectors (DET10A) with 1 nanosecond rise time which is a crucial parameter for depth resolution. We also use a pulsed laser source with a wavelength of 905nm and a beam splitter to divide the pulse into two. Finally we make use of collection optics for the reflected pulse and an oscilloscope for analyzing signals from the two detectors. In Figure 3.2 the schematic diagram of the time of flight measurement principle is given.

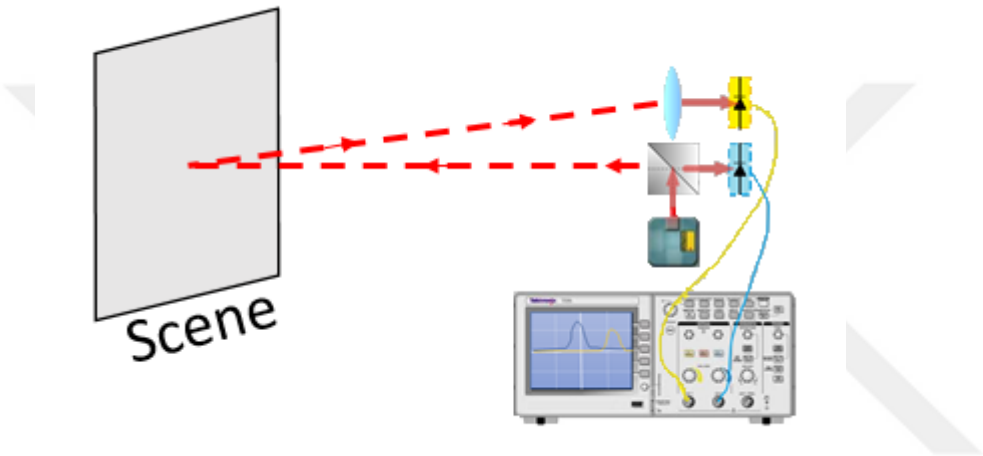


Figure 3.2 Two photodiodes are connected to the oscilloscope. Pulse laser beam coming from the laser source is divided to trigger the photodiode and illuminate the screen. Laser pulse flight to the screen and reflected pulse are collected by optics on the second photodiode. Then, the oscilloscope processes the signals coming from the photodetectors to calculate depth values.

The oscilloscope is triggered with the starting signal in channel 1 and the split beam is projected onto photodiode 2. In this setup depth is directly related to the time difference. As mentioned before, the rise time of the detector is important because the sharp increase in the signal is easy to threshold. Our detector is a THORLABS DET10A and its parameters are provided below.

Table 3.1 THORLABS DET10A detector parameters.

DET10

A

Active Area	Wavelength Range	Rise Time	Bandwidth	NEP $W/\sqrt{HZ}$	Dark Current	Junction Capacitance	Bias Voltage
$0.8mm^2$	200-1100nm	1 ns	350 MHz	$5*10^{14}$	0.3 nA	6pF	10V

After receiving the signal, a linear fit function is applied to it in MATLAB. The result is given in Figure 3.3. When measuring the data, we used the full length of the optic table, 280 centimeters, as depth. The recorded time of flight was 9.64 nanoseconds which corresponds to a distance of 289 centimeters. There is an error of 9 centimeters, which corresponds to 0.3 nanosecond. If we consider that the oscilloscope's sampling rate is 50 Megahertz, the error level is quite acceptable. The data processed in MATLAB can be seen in Figure 3.5. We used a 3-stage linear fit on our data.

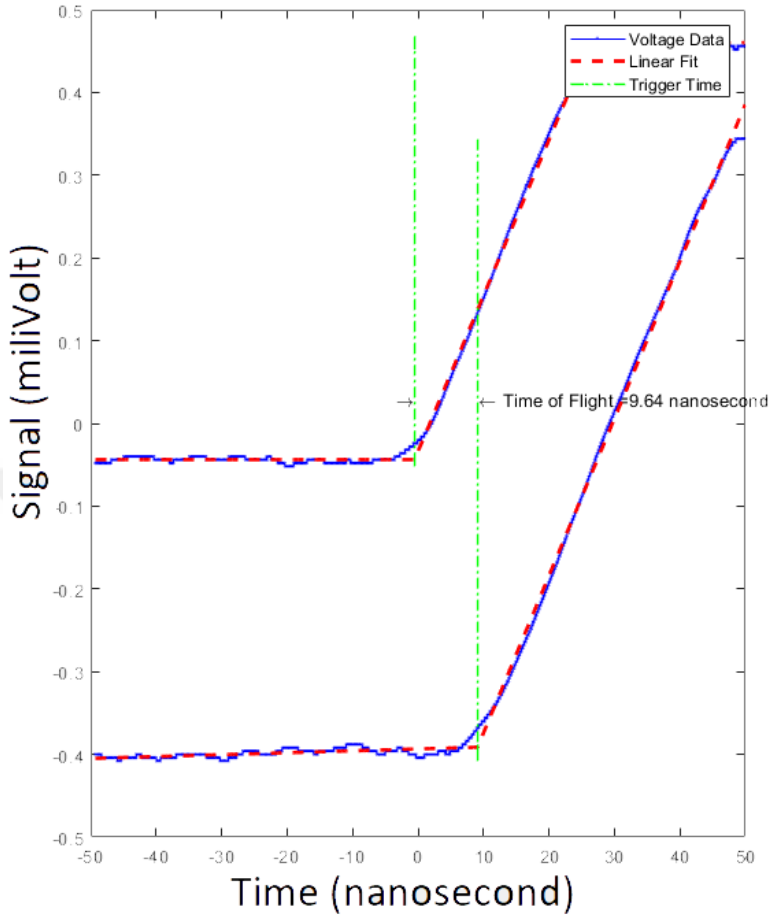


Figure 3.3 3-stage linear fit function is used to determine time of flight. There are additional techniques to measure the time difference between two signals of the same type such as rise time, fall time or peak point techniques.

After analyzing the single data point, we move on to multiple points by changing the distance between the detectors. The multiple-point measurements in Figure 3.4 were made by changing the distance by equal amounts each time.

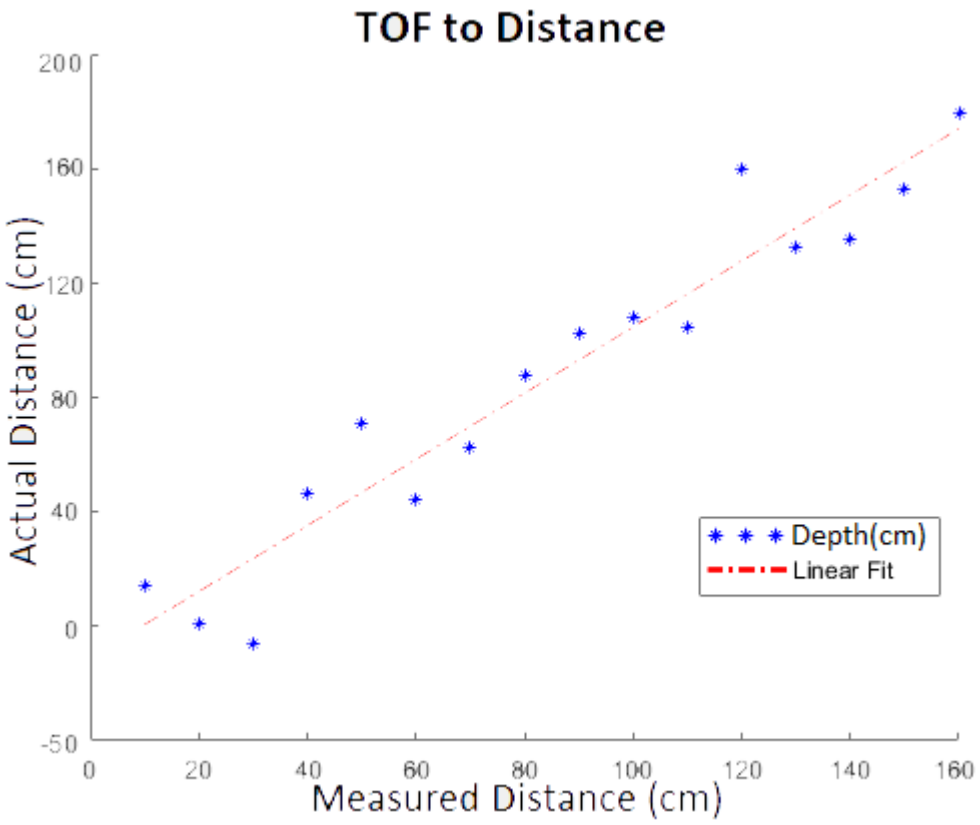


Figure 3.4 Actual Distance vs Measured Distance for 16 points with different depths. By using the same technique used to obtain the results in Figure 3.3, depth data is measured from the time of flight. The reference detector is kept at the same point and the target is gradually moved from 10cm to 160cm with a step size of 10cm.

This section provides a proof of concept for depth measurement using LIDAR time of flight. Subsequently, a DMD will be integrated into the LIDAR time of flight setup to act as a beam steerer. The next chapter is a detailed explanation of the optical design of the DMD LIDAR Setup. When looking at the experimental setup, the following question may arise: How is it possible to work on the nanosecond scale with a 50MHz sampling rate? The answer is the method known as equivalent time sampling. In this method, the scope gathers the necessary number of samples across with the same signal trigger. In Figure 3.5. equivalent time sampling is shown. Gray vertical bar is sampling period of the signal. Each equivalent sampling time shown with numbers(1,2,3,etc). All numbers refers the different part

of the signal and signal is reconstructed. The input signal must be repetitive because each time, a different part of the signal is gathered in equivalent-time sampling.

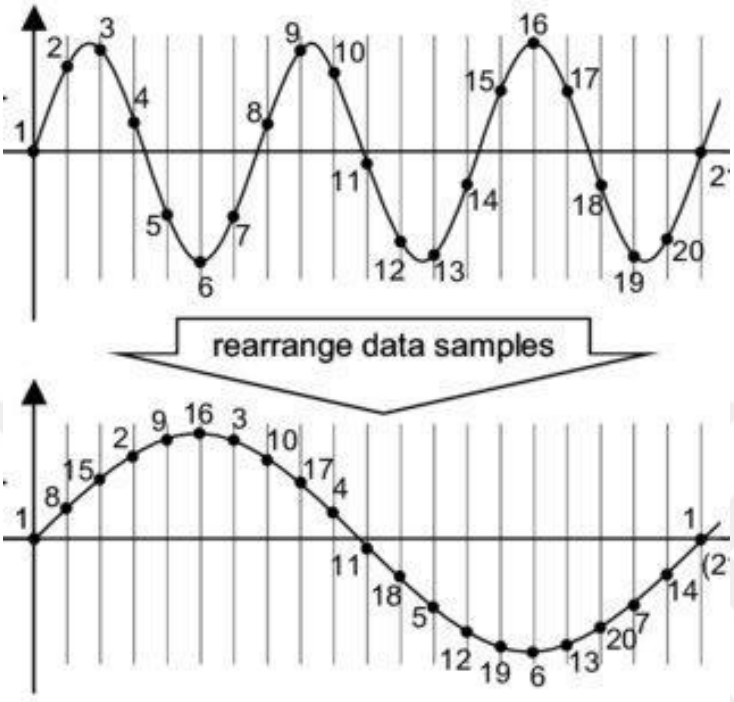


Figure 3.5 Schematic diagram of the equivalent time sampling method. The representations of the data acquired from an oscilloscope with the equal time difference method and the reconstructed signal. Maximum resolving power is used for sampling and the actual signal is resolved after a set of measurements. Captured from [52].

3.2.DMD LIDAR DESIGN & EXPERIMENTAL SETUP

Beam steering is a crucial part of a LIDAR system's design, various techniques are employed in the creation of a homogeneous scan pattern. The three main scan techniques used in airborne LIDAR systems are shown in Figure 3.6 below.

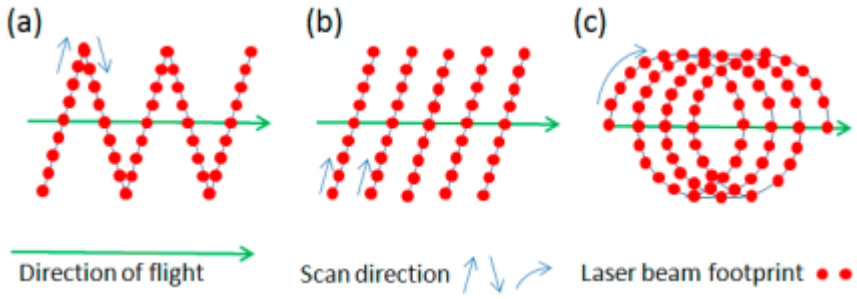


Figure 3.6 The three main scanning techniques. a) Oscillating Mirror. b) Rotating Polygon. c) Palmer Scan Captured from [46]

All of these systems depend on moving parts to perform scans, whereas a DMD has no moving parts. Thus, it is a more robust and novel method for beam steering. Once the proof of concept for the LIDAR time of flight system detailed in the previous section is complete, we change our setup by integrating a COTS Laser Range Finder Lightwave SF30/C to our setup as a time of flight calculator. We created an optical design for the DMD LIDAR system and drew a 3D model of it which can be seen in Figure 3.7.

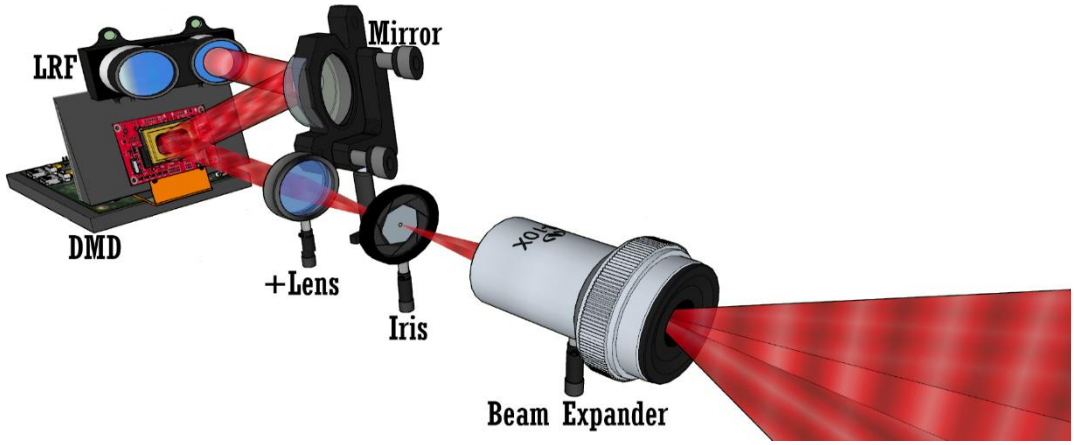


Figure 3.7 The method and required apparatus for the innovative LIDAR design. The laser range finder's pulsed laser source illuminates the DMD chip and the DMD pattern scans the scene. The first lens with the iris makes Fourier filtering of higher orders of the diffracted beam from the DMD. The beam expander adjusts the field of view of the system. Finally, the depth information is reconstructed using time of flight calculations.

After designing our system in SketchUp, we built the experimental setup shown in Figure 3.8.

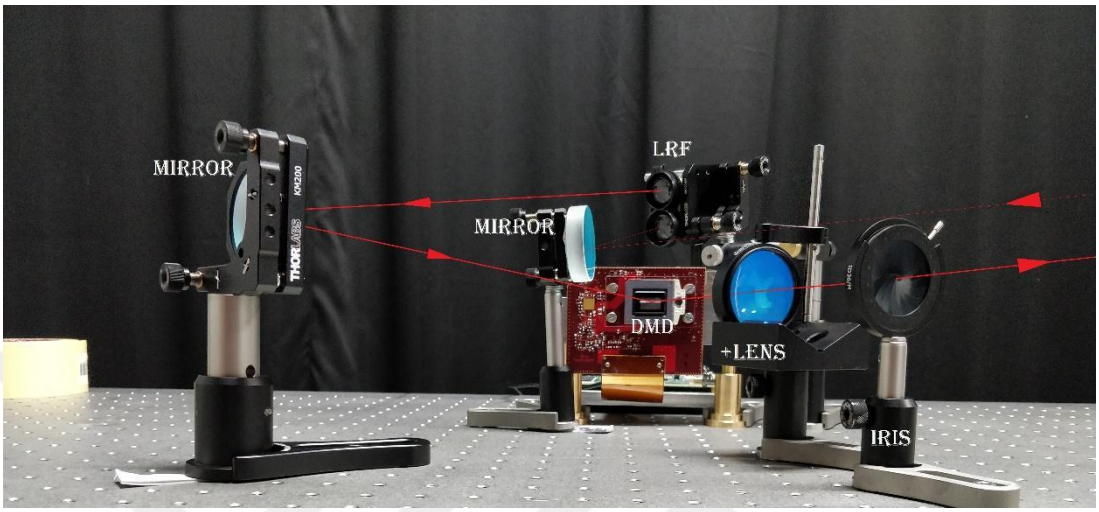


Figure 3.8 The experimental setup of the DMD LIDAR system whose SketchUp was provided in figure 3.7.

The LRF sends the 905nm pulse to the mirror which is then reflected by the mirror onto the DMD. The DMD can either be controlled by its own interface or 3rd party programs written in Python, Matlab etc. through a USB interface. In our experiments, we used a Python script for creating the desired patterns on the DMD. After programming the DMD, the light is scattered through the positive lens. Because we use a coherent light source and a DMD, a diffraction pattern occurs. The DMD is a diffractive optical element and thus unwanted diffraction orders need to be removed. By eliminating higher diffraction orders with the IRIS, we ensure that a clear pattern occurs on the screen. Being that the laser is pulsed, time of flight measurement of the related point of the scene can be measured. Different parts of a scene are reconstructed pixel by pixel by changing the pattern on the DMD and measuring time of flight each time.

3.3.IMAGE RECONSTRUCTION & RESULTS

After installing the experimental setup, we connected it to a PC through the USB interface and began capturing pixel-wise depth information using a Python script. The script uses pre-stored DMD patterns to illuminate the screen and the SF/30C measures the depth information for each pattern. The related embedded code for controlling the DMD written in Python is provided in the Appendices. When controlling the DMD pattern, time of flight needs to be measured in real time. In order to do so, we acquire the time of flight data from the Laser Range Finders own interface between consecutive DMD patterns. In Figure 3.11 LRF technical specifications and laser parameters taken from the LIGHTWARE Terminal program are provided.

Table 3.2 LIGHTWARE Terminal with firmware interface. SF/30C setting options.

LIGHTWARE Terminal (SF30/C Firmware)	
Output Type	Distance over USB
Exposure Type	799.600us
USB port Output Rate	39.08 / sec
Serial Port Baud Rate	115200
Serial Port Output Rate	39.08 / sec
Analog Range (0V to 2.048V)	256m
Analog Port Output Rate	39.08 / sec
Alarm Activation Distance	2.00m
Alarm Latch	OFF

Table 3.3 SF30/C technical specifications.

SF30/C General Properties	
Range	100m (natural targets) - 175m (reflective target)
Resolution	1cm
Update Rate	20010 readings per second (max)
Accuracy	+- 0.1 meters (%70 reflective target@20°C)
Power Supply Voltage	5V DC
Power Supply Current	250mA
Output & Interface	USB, serial & analog
Dimension & Weight	30 * 56.5 * 50 millimeters
Connections	Plug & socket, micro USB
Laser Power	20W(peak), <18mW(average), Class 1M
Optical Aperture	51 millimeters
Beam Divergence	0.2°
Operating Temperature	0-40°C

Table 3.4 Laser parameters used in SF/30C

SF30/C Laser Specifications	
Laser Wavelength	905nm
Pulse Width	<20ns
Pulse Frequency	<36kHZ
Peak Power	<10W
Average Power	<0.6mW
Average Energy Per Pulse	<300nJ

We use 3 different raster scan patterns with the DMD. Each of them scans the screen with differently sized pixel groups. For example, the first experiment is done with an 8x2-fold scanning. This means we divided the DMD pixels into groups of size 240x540 which act like a single large pixel.

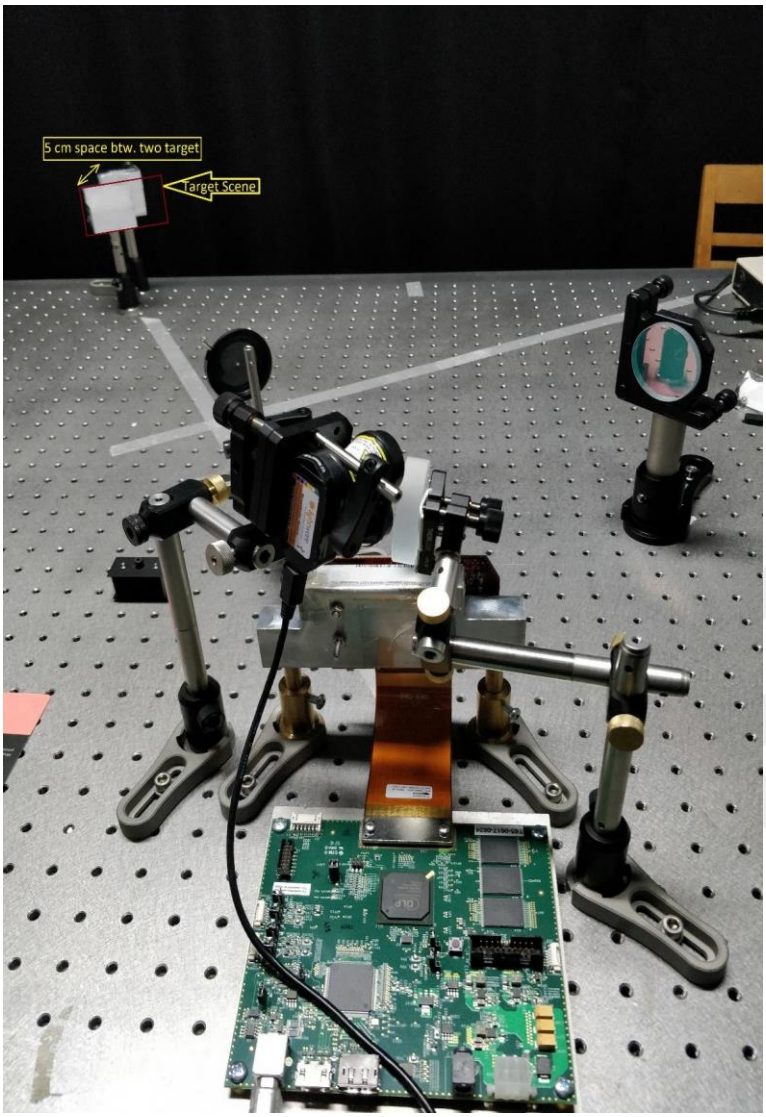


Figure 3.9 The experimental setup of the DMD LIDAR design given in Figure 3.7. The target scene is shown by the red rectangle. There is 5cms of space between the two targets.

In Figure 3.9, the target scene is shown with a red rectangle and Figure 3.10.b shows the reconstructed result of the depth scanningThe distance between two objects at the target scene is set to 5 cm.

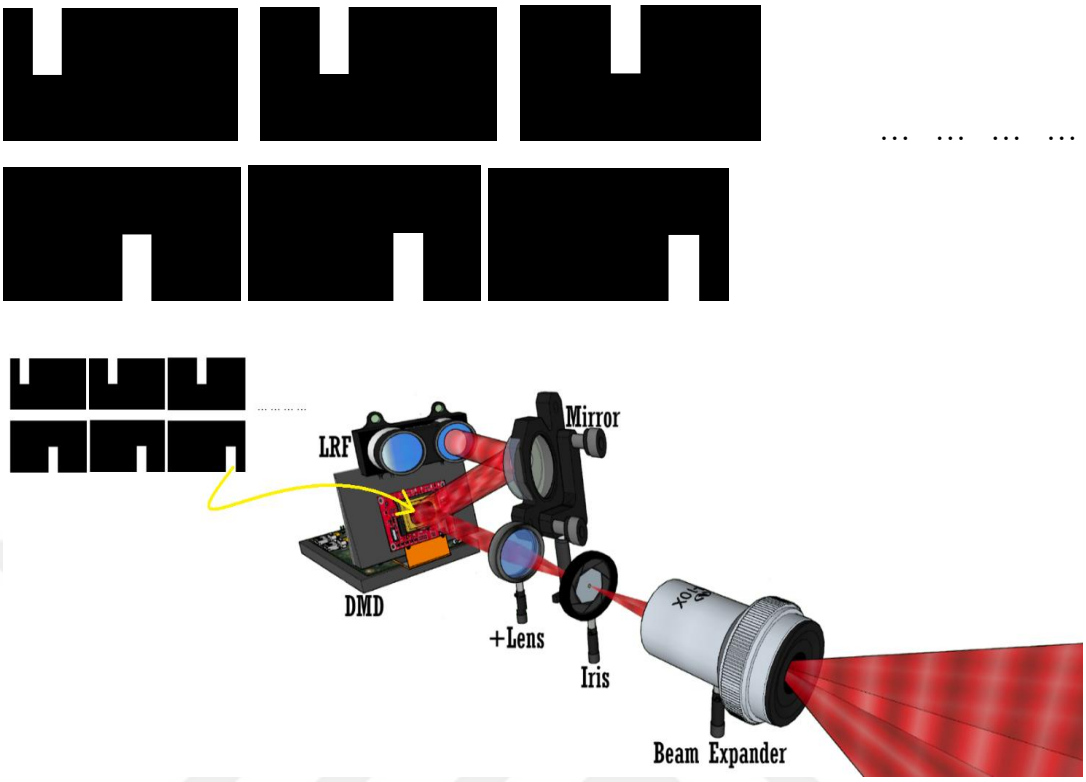


Figure 3.10 a. 8x2 scanning pattern (pixels are divided into 240*540 groups). Each of the 32 patterns is loaded onto the DMD respectively and the range value is logged with SF/30C using a Python script through the USB interface.

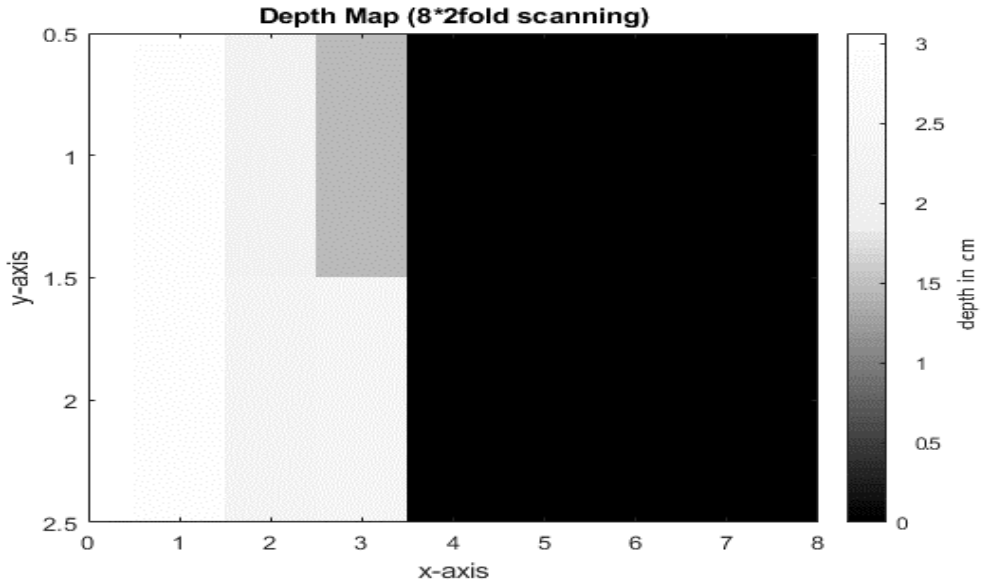


Figure 3.10. b. Reconstruction results for the 8*2 depth scanning of the scene.

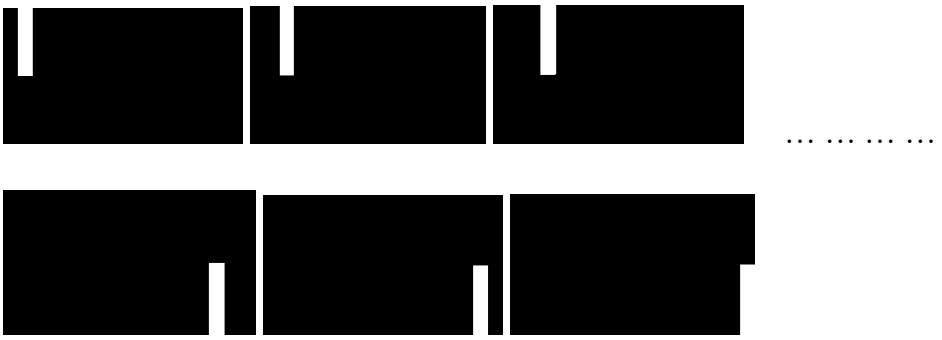


Figure 3.10. c. 16x2 scanning of the scene. (pixels are grouped into 120*540 clusters). Each of the 32 patterns is loaded onto the DMD respectively and the range value is logged with SF/30C using a Python script through the USB interface.

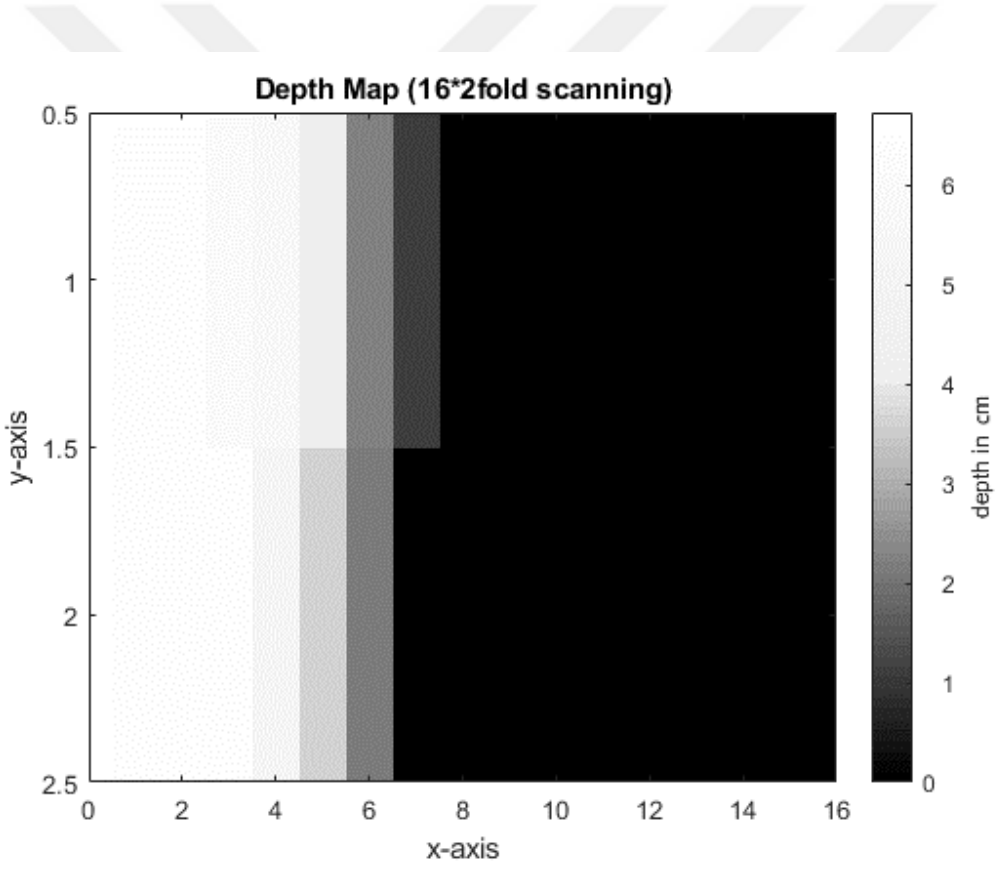


Figure 3.10. d. Reconstruction result of 16x2 depth scanning of the scene.

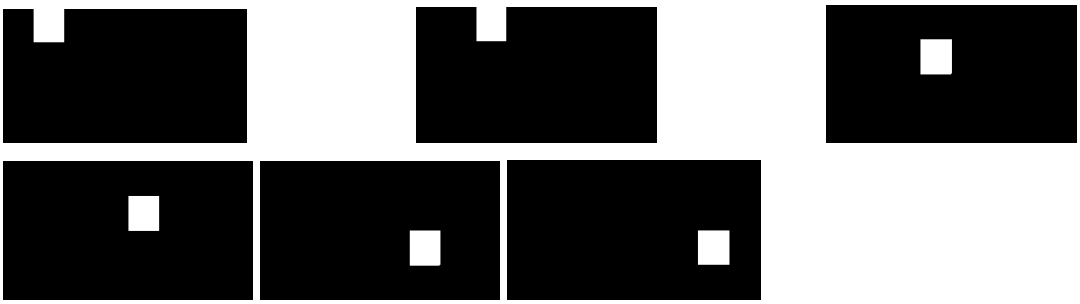


Figure 3.10. e. 8x4 scanning of the scene (pixels are grouped into 240*270 clusters). Each of the 32 patterns is loaded onto the DMD respectively and the range value is logged with SF/30C using a Python script through the USB interface.

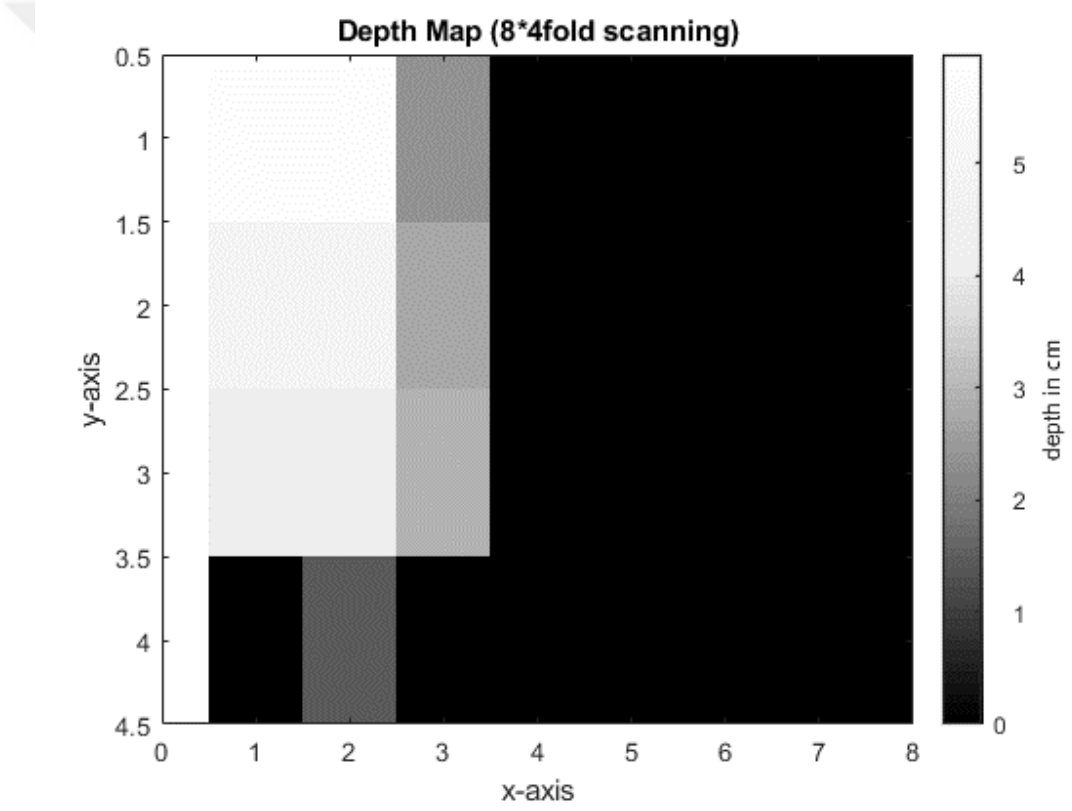


Figure 3.10. f. The reconstruction result of the 8x4 depth scanning of the scene.

As shown in Figure 3.13, the difference in depth between the two targets is resolved. However, the reconstructed depth between them differs based on the pixel clustering size. As mentioned before, the constant trigger level in SF/30C causes this measurement problem. It is clearly seen, even same depth on the scene may result in

different depth value after postprocessing. Here in the Table 3.5. below performance comparison of different resolutions are given.

Table 3.5 Performance and scanning time comparison of different resolutions on DMD.

Resolution	16x2	8x2	8x4
Performance	Best	Good	Bad
Scaning Time	Slow	Fast	Slow

We expect that the lowest resolution option results in best performance because of the relative homogeneity of laser beam. However, that was not the case. Because of the intensity of our laser beam is not homogeneous everywhere as seen in Figure3.12.. Unhomogeneous distribution of the beam cause different trigger time on TDC and this results in different depth value after postprocessing. The algorithm and the depth offset due to the fixed trigger level are explained in chapter 3.4 and Figure 3.11 in detail.

3.4.FUTURE WORK WITH DMD LIDAR

The experiment is done successfully with a DMD. The alignment, diffraction and light intensity problems were very difficult to handle. We use an alignment laser for aligning the detector FOV to the scene. In addition, because we use a coherent light source; the some of the diffracted beam from the DMD surface falls on the scene unintentionally. We have used a light dump and a Fourier Filter to deal with this issue. Another problem we faced was the intensity. When we measure the same depth point for a different material (with a different reflectivity at 905nm), we get different intensity readings. In the figure below it is clearly seen that, if the algorithm uses a fixed trigger level for the time of flight calculations, it miscalculates the TOF. However, when a Lightware SF/30C rangefinder is used, the algorithm works fine

with a fixed trigger level. The TOF changes for the same depth when the material changes. Two solutions to achieve higher resolution with a DMD scan are proposed.

The first one is changing the fixed trigger level of SF/30C to get the same depth result from a fixed point when different amounts of intensity are reflected.

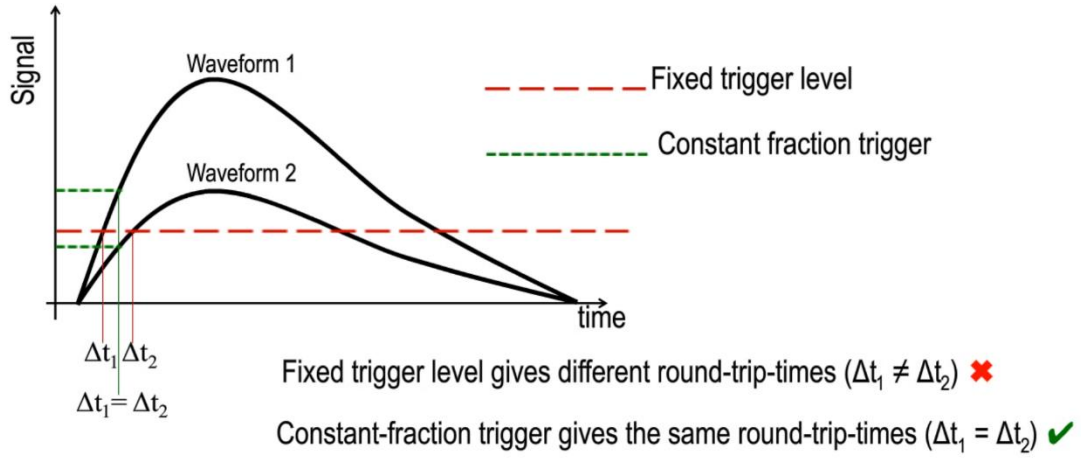


Figure 3.11 Trigger type of the given signal. A fixed trigger level is misleading when the intensity level of the reference signal and the reflected signal is high. Constant fraction trigger gives better results when the intensity level difference is high. Adapted from [30]

However, we have faced the same problem on the DMD surface. Our laser was not perfectly uniform and its intensity distribution is provided in the figure below.

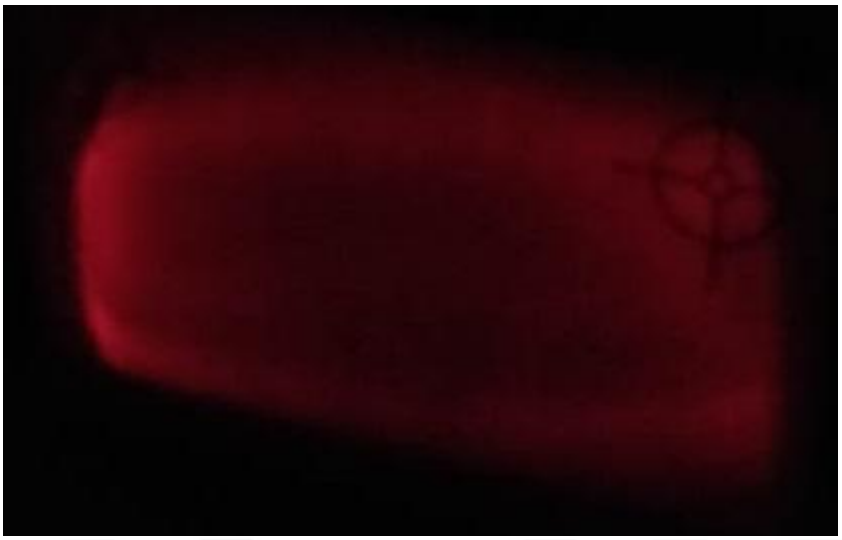


Figure 3.12 a. Laser beam intensity on IR card after DMD. Intensity differences on the beam are easily seen on the IR card which results in misleading reading of the depth values in SF/30C.

Let's assume that the scene is flat and every point has the same depth, then the DMD mirror is partitioned and the scene is scanned with the pattern below.

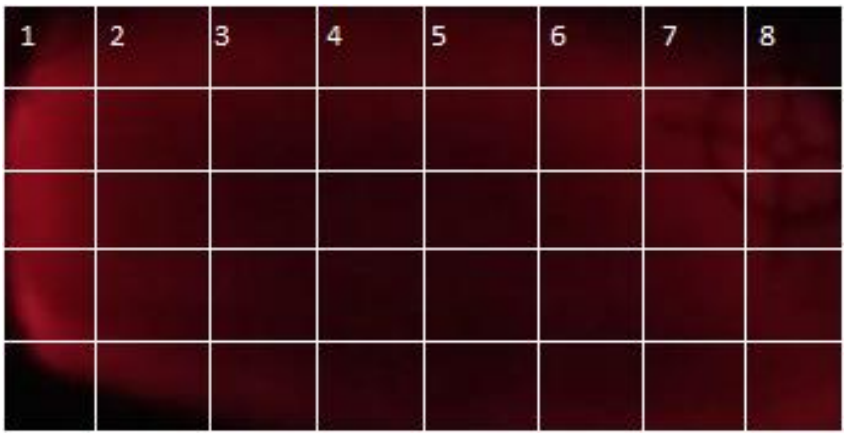


Figure 3.12. b. Portioned laser beam intensity on IR card after DMD.

By changing the trigger level algorithm in SF/30C, these problems can be solved. Because of the intensity changes in the pixels grouped into clusters 2 and 7; the fixed trigger level algorithm in the SF/30C yields different depth data.

Another solution to this problem is focusing the light in the diffraction order of the DMD. When the light is focused using all pixels on the DMD, the intensity gets higher and a small amount of change in the intensity level of focus is ignored when it is scanned through the scene. In the figure below, computer generated holograms are used to focus the light in a far field with DMD.

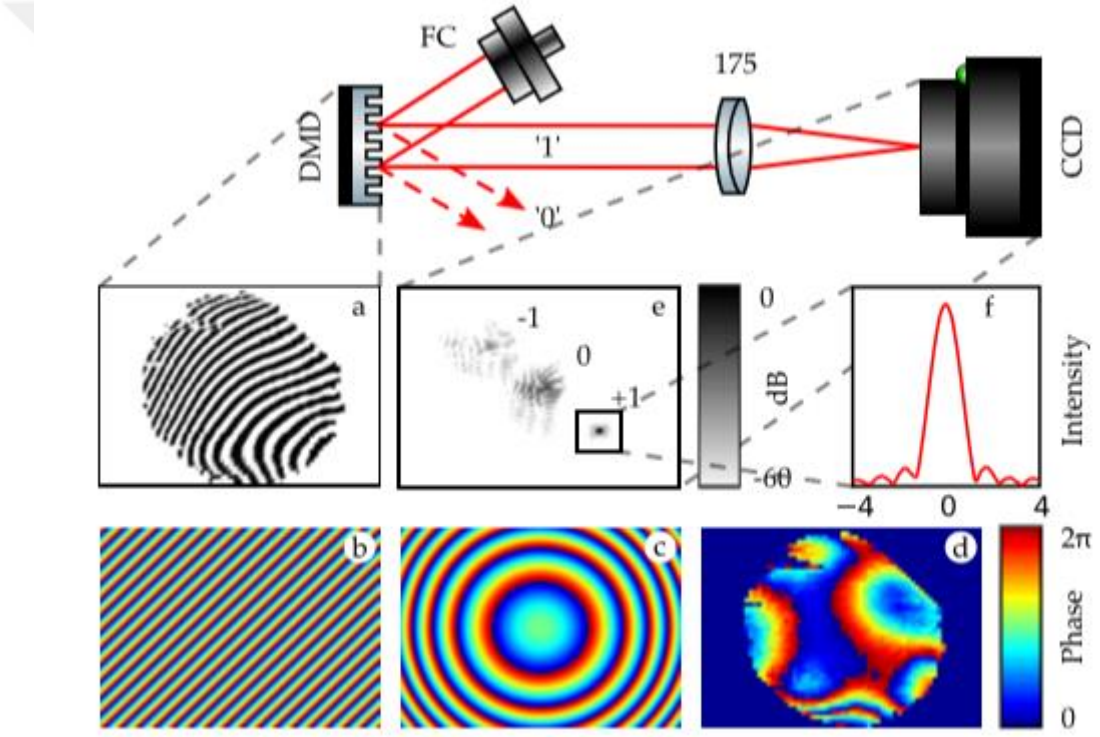


Figure 3.13 The experimental setup used for testing the holographic optical tweezers. A collimated beam of laser light at 785 nm is delivered from a firer coupler (FC) and is incident on the DMD. The reflected light from mirrors in state '1' is focused using an achromatic doublet lens (f = 175 mm). The traps are produced in the focal plane of this lens and imaged with a CCD camera. (a) The binary hologram on the DMD, formed from (b) a linear phase ramp, (c) a quadratic lens and (d) an arbitrary phase correction map. (e) The intensity measured by the CCD camera. (f) A line profile through the desired trap showing diffraction-limited performance (the x-axis is in

dimensionless units of λD/f where D is the diameter of the limiting aperture). Adapted from [28].

In [28] coherent light sources illuminate the DMD. A computer generated hologram is created and loaded as the pattern on the DMD and the first order diffraction is captured with the CCD. When specific patterns are projected on the DMD, it is observed that a focus point appears at certain locations on the CCD.

Both of these solutions can be used separately to deal with this problem, however, to achieve the best results, these two solutions should be employed simultaneously, that is, light should be focused by using diffraction (to increase intensity) and the trigger algorithm should be changed from a fixed trigger level to a constant fraction trigger level algorithm (insensitive to intensity.

CHAPTER 4

STRUCTURED LIGHT AND TRIANGULATION FOR MACHINE VISION

4.1.TRIANGULATION

According to its definition in trigonometry, triangulation is the process of determining an unknown value of a triangle (can be a side or an angle) from its known sides and angles. The human visual system uses this principle as in stereo vision. Our visual system is evolved in such a way that we have a 3D perception of the world around us. We know the distance between our eyes and the angles are calculated in our brains to reconstruct (predict) depth values from these data. In this thesis, triangulation is done by removing one of the eyes (cameras) and replacing it with a light projector. In the figure below, we see the triangle made by the projector, the camera and the object. Before starting the triangulation process, pixel difference and calibration methods are used in the experimental setup for depth prediction. Later we show that in our triangulation method; the distance B is fixed, the θ matrix is developed with related patterns and α values are taken by mapping camera pixel values to the angles through image processing. These parts of the work and the image processing code used in this thesis is given as a Matlab script in the Appendices.

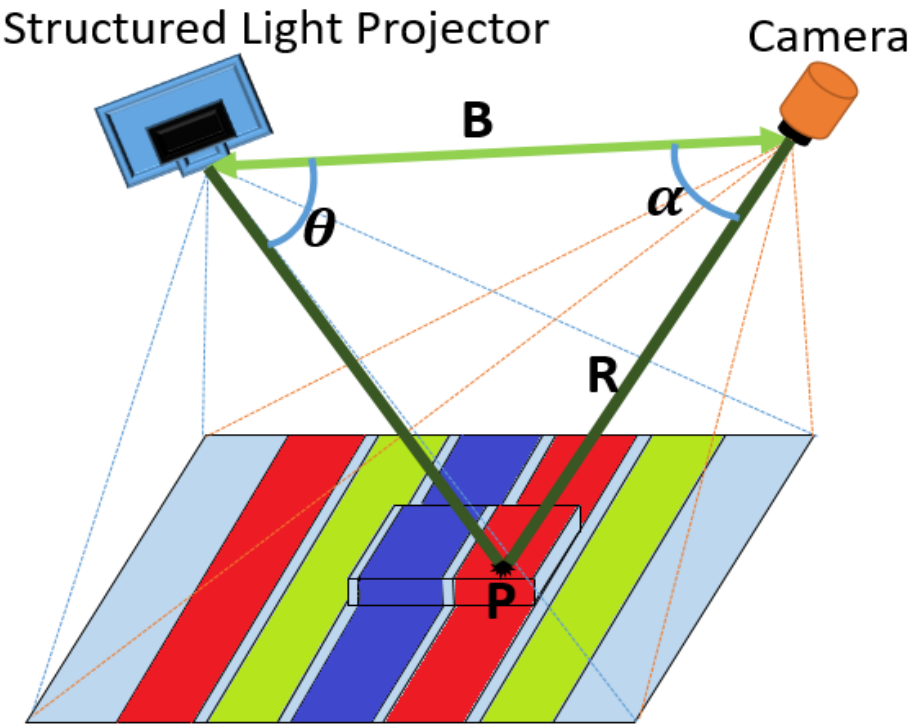


Figure 4.1 Basics of the triangulation method are shown. The distance between the structured light projector and the camera is given with B. R is the depth value of the point P with respect to the camera. Alpha and Theta are the angles of the point P with respect to camera and projector.

The most frequently used method for 3D imaging using triangulation is done by actively illuminating the screen using structured light with spatially changing intensity patterns. As seen in 4.1, the structured light pattern illuminates the screen by the help of a projector or a light source modulated with a DMD. The light intensity on the camera sensor is converted to a binary signal after the image is processed. A DSLR camera is used to get a 2D image of the scene illuminated by structured light. If there is nothing on the scene (planar surface), the pattern acquired by the camera is similar to the image of the light source modulated with the DMD. However, when there is an object in the scene, because of the geometrical shape of the object, the acquired structured pattern is distorted by the object. The principle is to create the triangulation rule for each pixel on the camera and get depth information from the distorted pattern. There are many algorithms using different types of structured light

for reconstructing a scene. The main principle, triangulation, remains the same for all techniques. The geometric relation between the 3 points (the light projector, the camera, and the point P) is demonstrated with the formula,

$$R = B\frac{sin(\theta)}{sin(\alpha + \theta)} \tag{9}$$

4.2.STRUCTURED LIGHT

Structured light is the modification of light intensity or the phase values to direct light anywhere in space with the desired intensity and with phase values. By the help of structured light you can overcome the correspondence problem and you can easily reconstruct 3D data with less processing power. In 2013, Brunao et al.[47] made a comparative study between structured light and passive stereo vision. They classified stereo matching algorithms into two groups called local and global algorithms. These algorithms are based on some constraints such as the epipolar constraint, the similarity constraint, the continuity constraint, the uniqueness constraint, and the ordering constraint to solve the correspondence problem. They show that the local algorithms are much faster than the global ones, however, global approaches solve the correspondence problem better than local ones. Structured light is much more efficient than passive methods because little image processing is required to solve the correspondence problem. The structured light approach involves a projection of the structured-light patterns in some order to code the screen. By using this method, the matching of the related points is done very efficiently, because each point on the object is uniquely determined by a designated code.

These systems are generally composed of a camera and a light projector. The light projector can be white light, laser line or speckle dot pattern. The triangulation principle is used between the camera, the light projector and the object.

As shown in the figure below, the projector projects a known coded pattern onto the scene, generally a set of light planes or scanning laser lines where both the structured light and the camera pixels are known. Once the algorithm determines the same point on both the camera pixels and the projected pattern, the image is reconstructed.

There are different structured light techniques in literature such as binary code(time multiplexing), gray code, continious coding, phase shift, rainbow method, varying color code, color coded stripes, De Bruijn Sequence, pseudo random dot speckle, color coded grid etc. [48]. In this thesis 8 sets of binary coding patterns are used to encode the objects.

Time multiplexing methods such as binary, gray or color coded stripes use codewords created by projecting a predetermined set of patterns in a given order onto target objects. The codeword corresponding to a specific pixel or group of pixels on a given image is obtained only when all patterns have been projected.

Each new pattern corresponds to a new bit (or digit) in the codeword, resulting in a direct relationship between precision and the number of patterns. In most methods, patterns projected earlier correspond to more significant bits in the codewords. Also, codeword basis is kept small to minimize noise.

Continuous coding methods such as varying color code or the rainbow method use periodic or absolute patterns with varying intensity or color throughout target axes.

Phase shifting methods; when a sinusoidal grating is projected onto a 2D surface, a unique phase value can be assigned to all points along lines parallel to the coding axis. Any 3D object will cause deformations in the form of bending or shifting of lines parallel to the coding axis, resulting in differences between projected and recorded patterns. These differences contain depth information. Object shapes can be recovered by matching points on projected and recovered patterns.

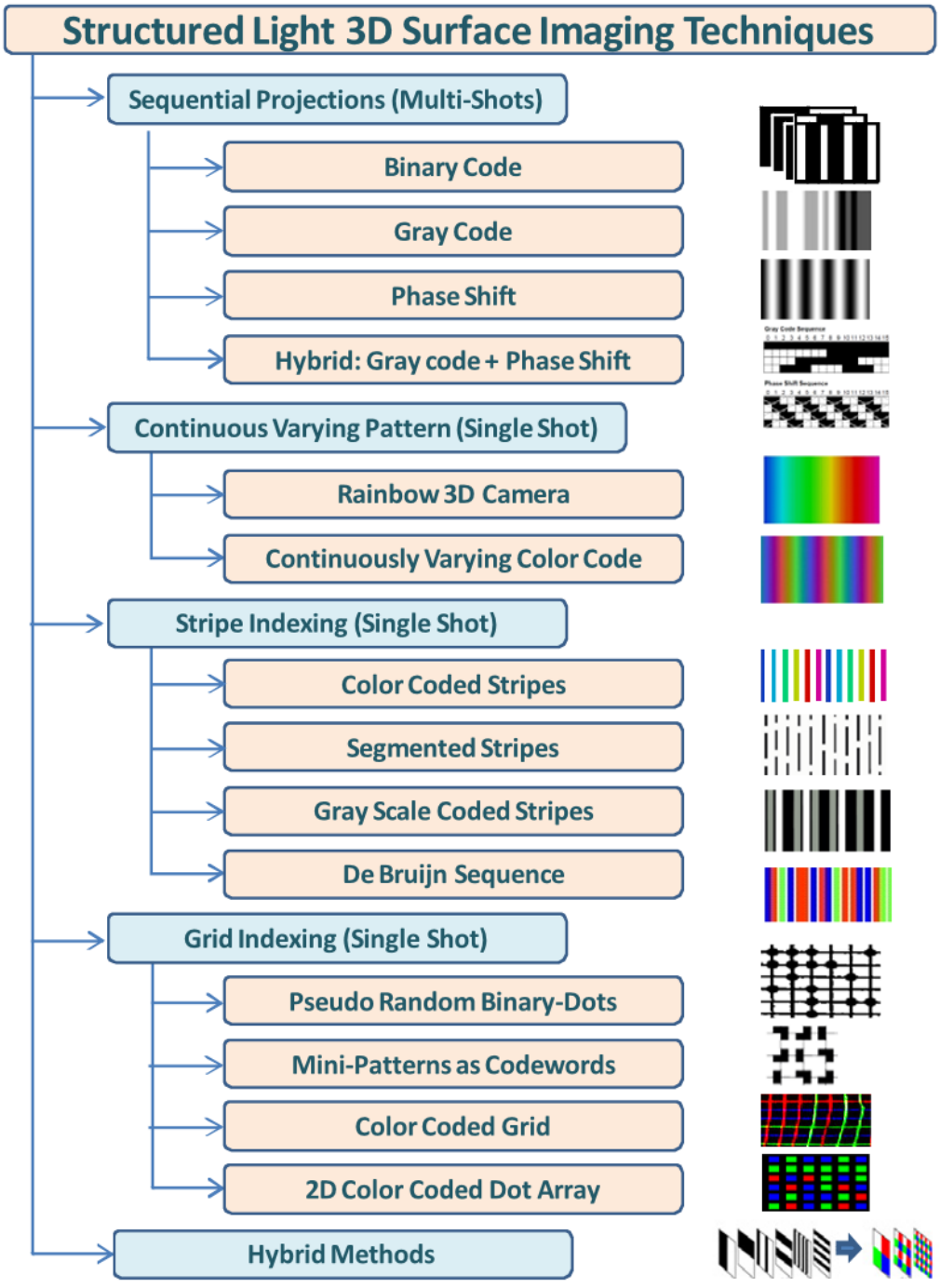


Figure 4.2 Different structured light techniques are given. In this thesis, binary code is used as the sequential projection. On the left, the name of the technique is written. On the right, the projected pattern on the screen is shown. Captured from [4].

For example, a Rainbow 3D camera uses colors to encode the screen. In the figure below, the object is illuminated with a structured light named the “Rainbow Pattern”.

4.3.DMD MACHINE VISION DESIGN & EXPERIMENTAL SETUP

We used an 800x600 resolution light projector and eight sets of patterns to illuminate a mannequin. A 1920x1080 camera is used for capturing images.

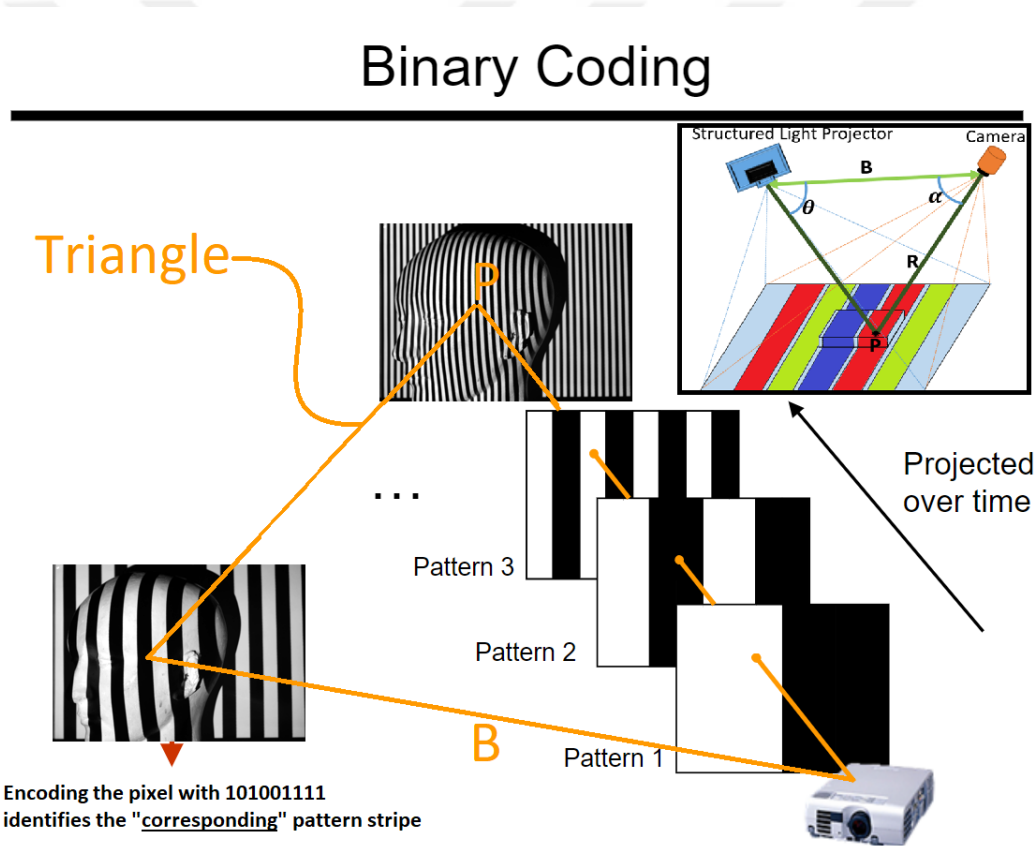


Figure 4.3 Patterns are projected onto a mannequin sequentially. Point P is encoded with a binary intensity value. The scene is then processed with image processing techniques to extract the depth information hidden in this encoded pixel.

In our first try, we use the pixel difference method and calibrate it by measuring the true distance between camera and mannequin. After we get a point cloud by using adaptive threshold and reconstruction algorithms, we post processed the data with median filtering. In chapter 4.4 our results are given.

4.4.IMAGE RECONSTRUCTION & RESULTS

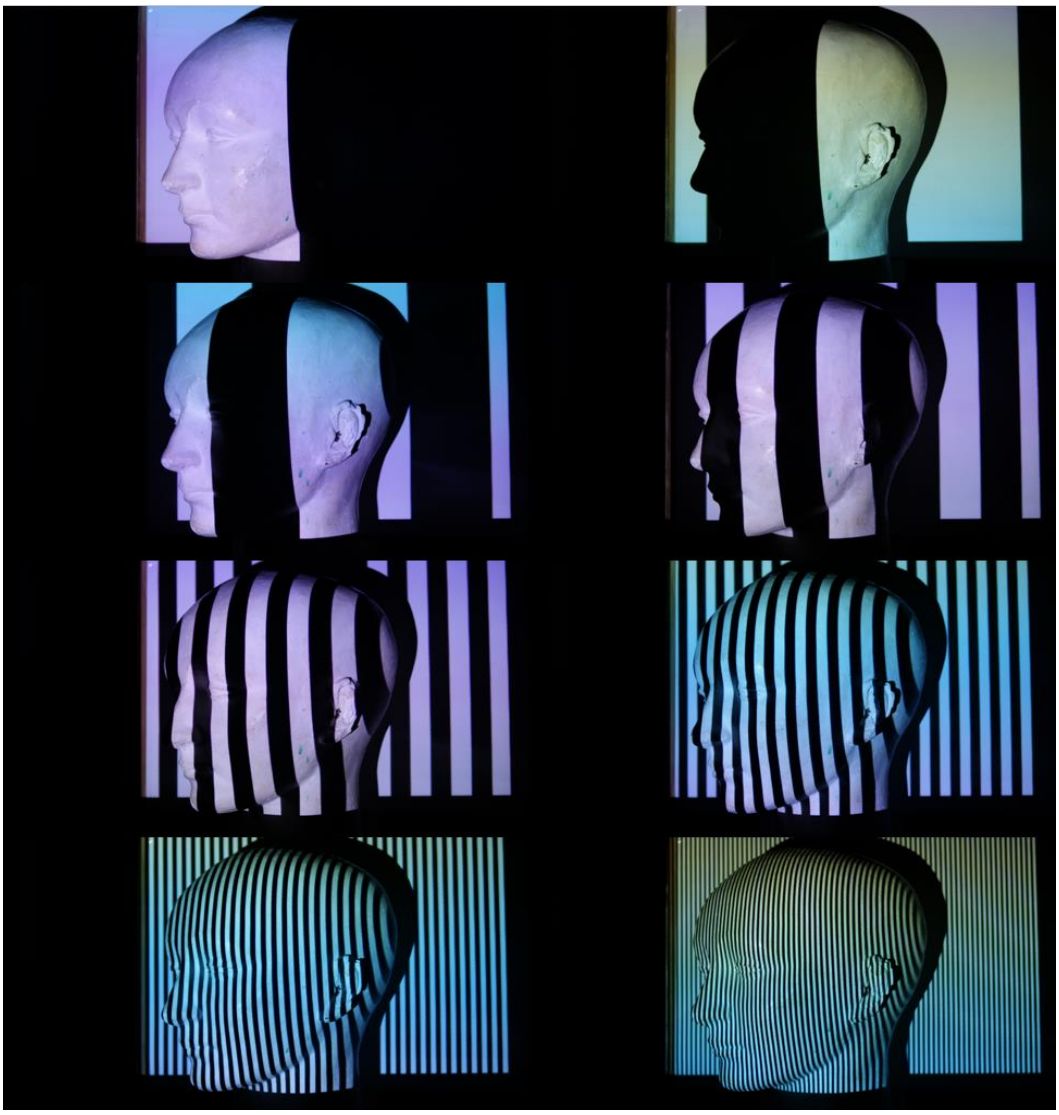


Figure 4.4 Binary pattern projected on a mannequin sequentially with 2^n stripes. n = 8 for our case.

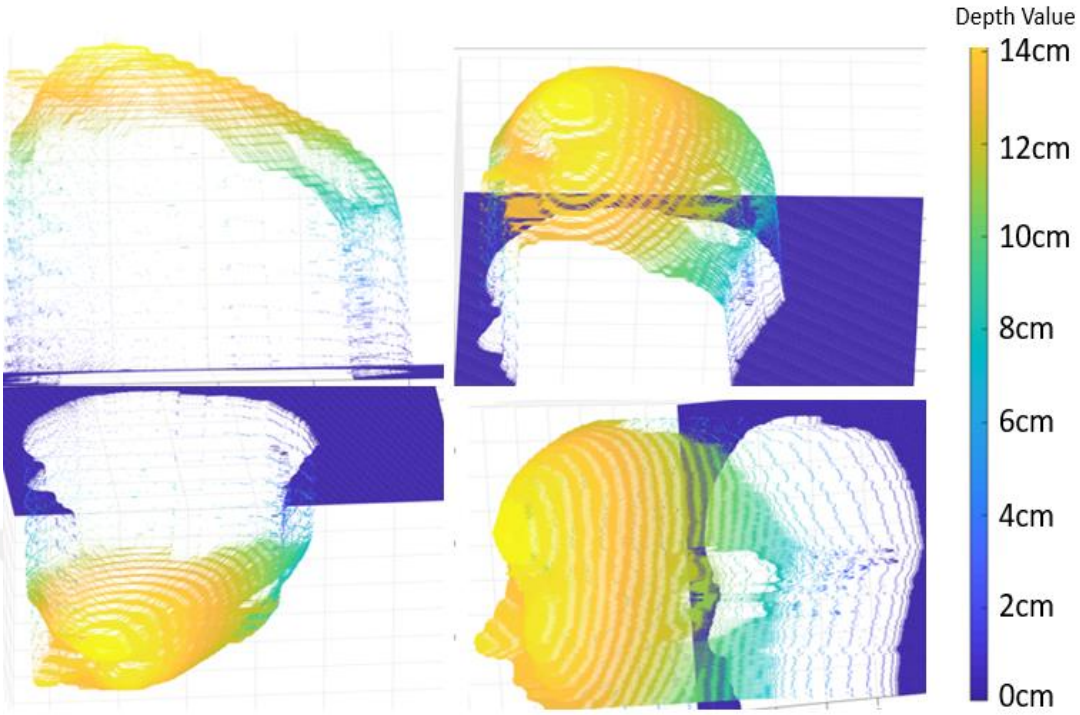


Figure 4.5 Depth value of the mannequin is given in centimeter resolution. Color bar describes the depth.

Pixel difference method takes place in mannequin case. We used 2^n stripes where $n = 8$. This means that our scene is divided into 256 horizontal sampling spaces. Besides, we placed object(mannequin) in resolution of 800x600 pixel at image space. Our object is positioned 2 meters away from the projector and camera. Each horizontal point is reconstructed below centimeter resolution. Reconstruction detail is given in the Appendix B as a MATLAB Code with explanatory comments.

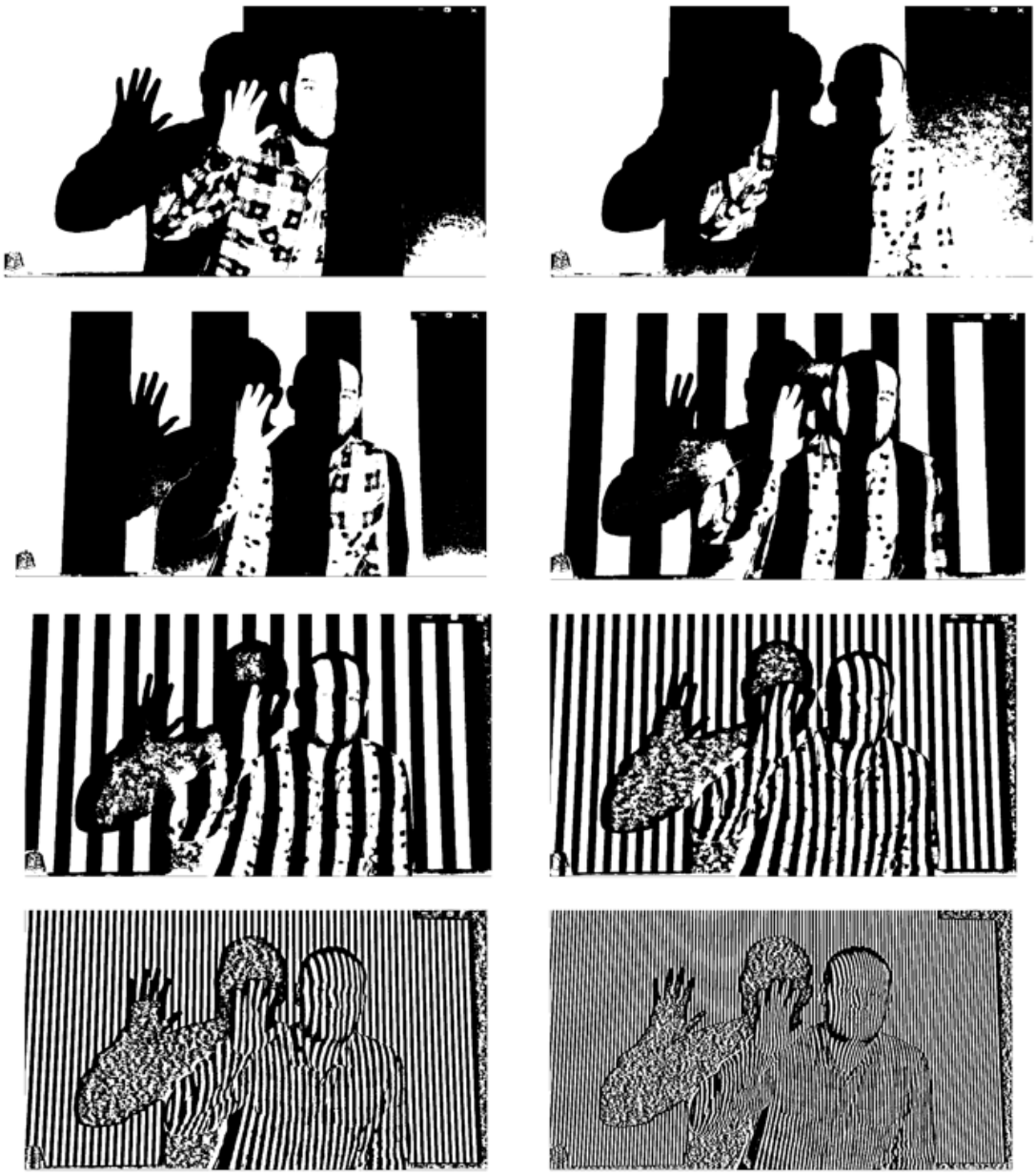


Figure 4.6 Binary pattern projected on author (Çağdaş Anıl YÜKSEL) sequentially with 2^n stripes. n = 8 for our case.

Triangulation takes place in second part (3D representation of author) according to equation (9). Here again 256 stripes are used. In addition, object(me) is captured in resolution of 1920x1080 pixel at image space. Our object is positioned 4 meters away from the projector and camera. Each horizontal point is reconstructed below centimeter resolution.

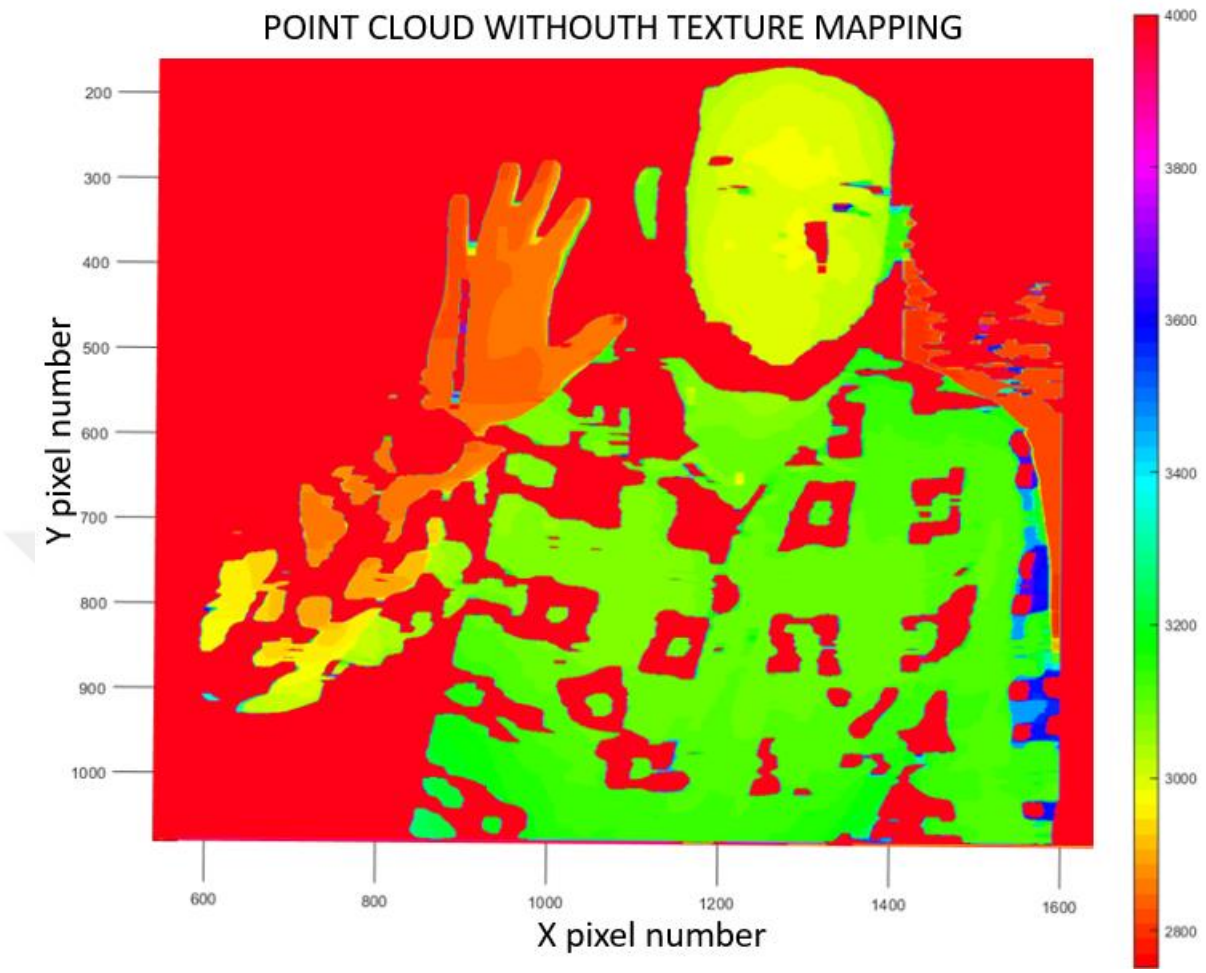


Figure 4.7 Depth value for the author is given in millimeter resolution. The color bar describes depth.

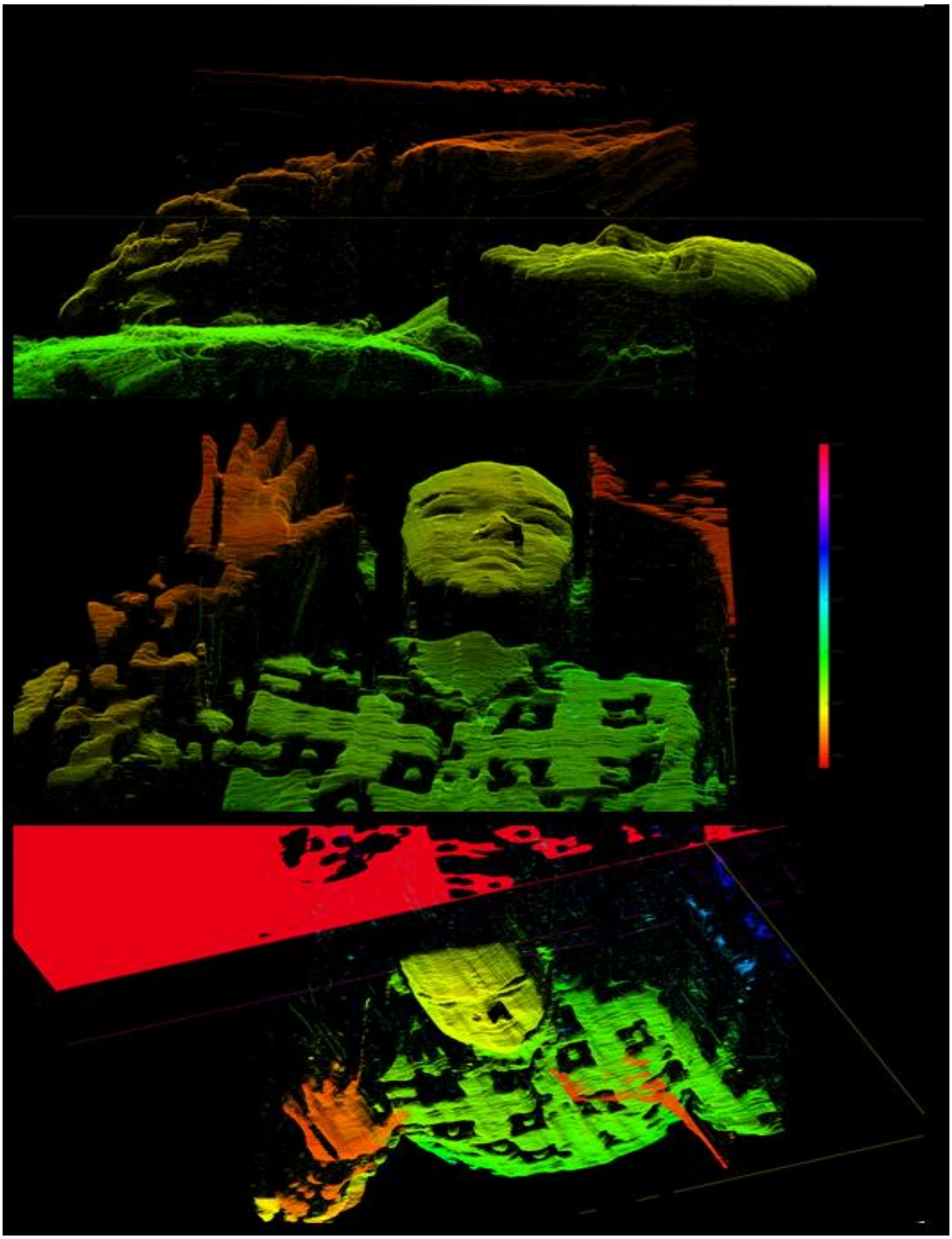


Figure 4.8 3D depth information is visualized in MATLAB. Different angle of view of the author is given.

Table 4.1 Performance and scanning time of structured light technique.

Resolution for DMD	1080(y)	1920(x)	-
Resolution for Camera	1080(y)	1920(x)	-
Resolution For System	1080(y)***	1920(x)***	< cm(z)*
Scan Range	-	-	4 meter(Lab Conditions)
Scaning Time	<second**	<second**	~2 seconds

*Behind the author, in Figure 4.6-7-8-9, there is a wall. Wall depth is measured by hand. It is 4 meter at the center of the camera. Depth resolution is calculated by comparing pre-known distance with reconstructed one.

**The time calculated for 8 shots of photos for the 8 stripes. Z calculations depens on the hardware. We have used İntel i5 2470k and got a result of 2 seconds.

*** 2^8 =256 stripes is used for each pixel. However reconstruction resolution is not only depend on the stipe size but, DMD, Camera and # of stripes.

As 3D data capturing technologies such as laser scanning, photogrammetry and various computer-assisted methods become more efficient and widely available, the importance of visualizing 3D objects increases. Many methods for (preferably photo-realistically) displaying 3D objects exist in computer graphics. One such method, texture-mapping, is the process of superimposing or projecting a digital photo, called a texture, onto a 3D object. Depending on texture resolution, texture mapping can also be used to create more realistic representations of 3D models with little detail.

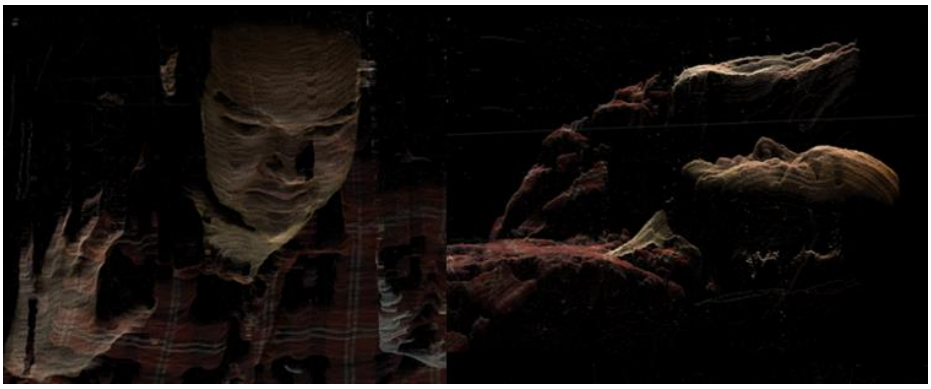


Figure 4.9 Point cloud without and with texture mapping, respectively.

In Figure 4.10 and 4.12 we use 512 stripes(n=9) for increasing x axis resolution. In Figure 4.11 and 4.13 results of resolution study are given.

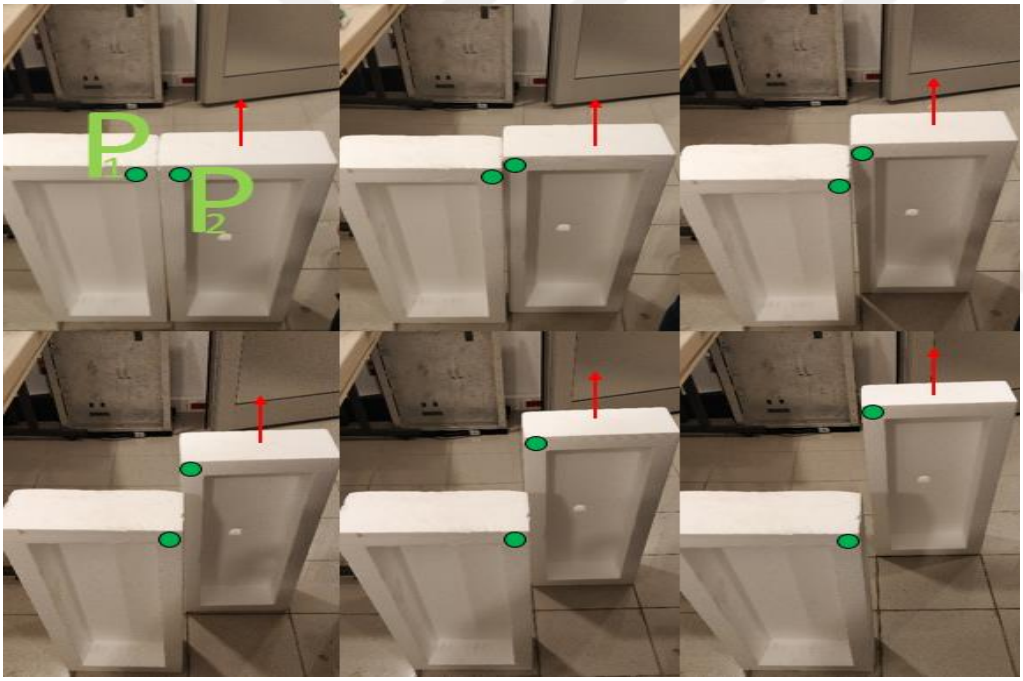


Figure 4.10 Specific two points is measured. Point P1 is fixed and P2 move apart through in parallel to the Camera axis. Each time the depth values and differences are measured by hand and compared with reconstructed result.

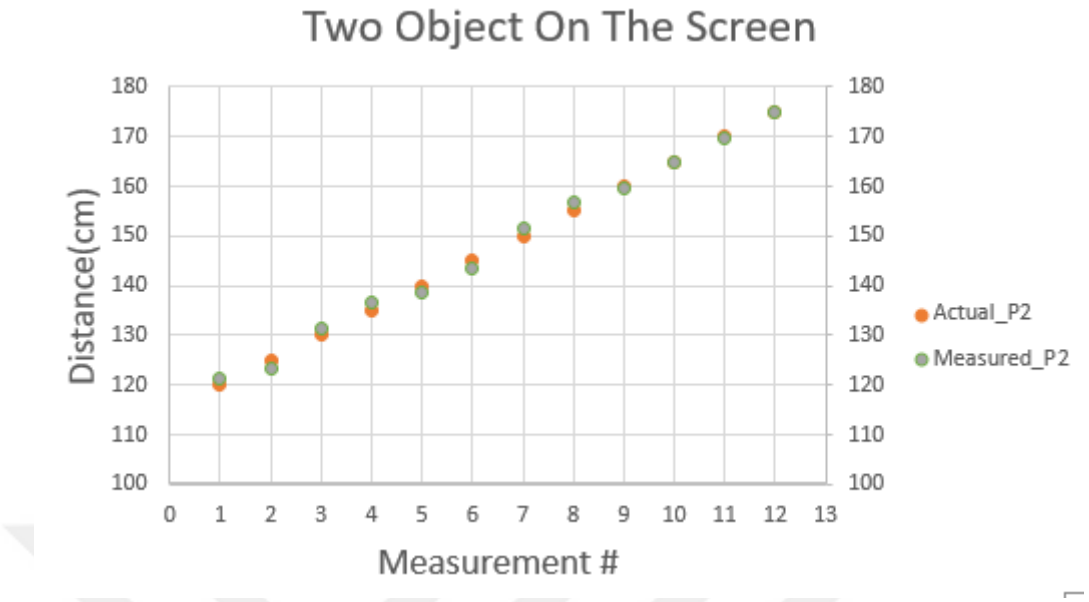


Figure 4.11a. Actual (by hand) and measured distance comparison of the point P1 and P2 given in Figure 4.10.

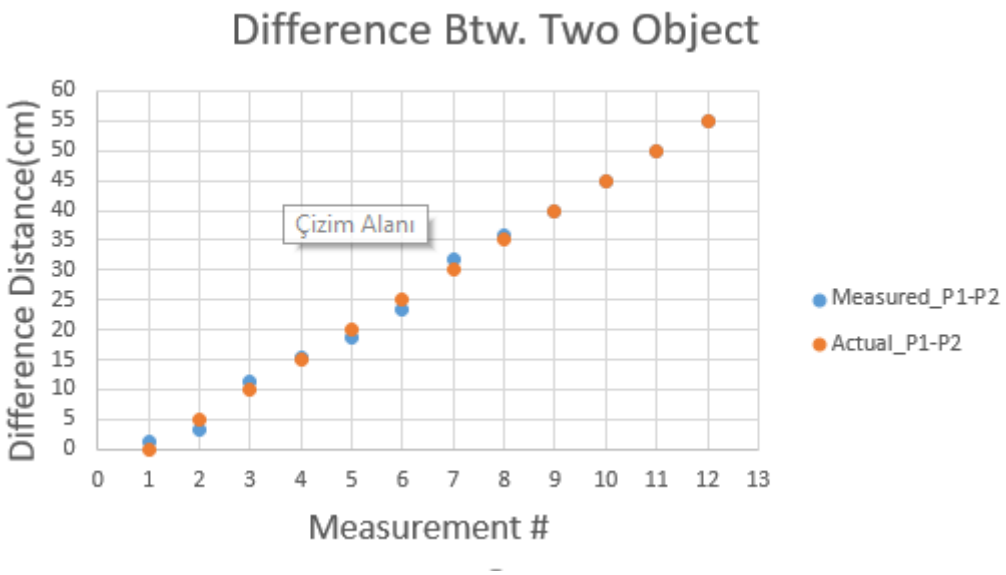


Figure 4.11.b Actual(by hand) and measured distance comparison of the point P1 and P2 given in Figure 4.10.

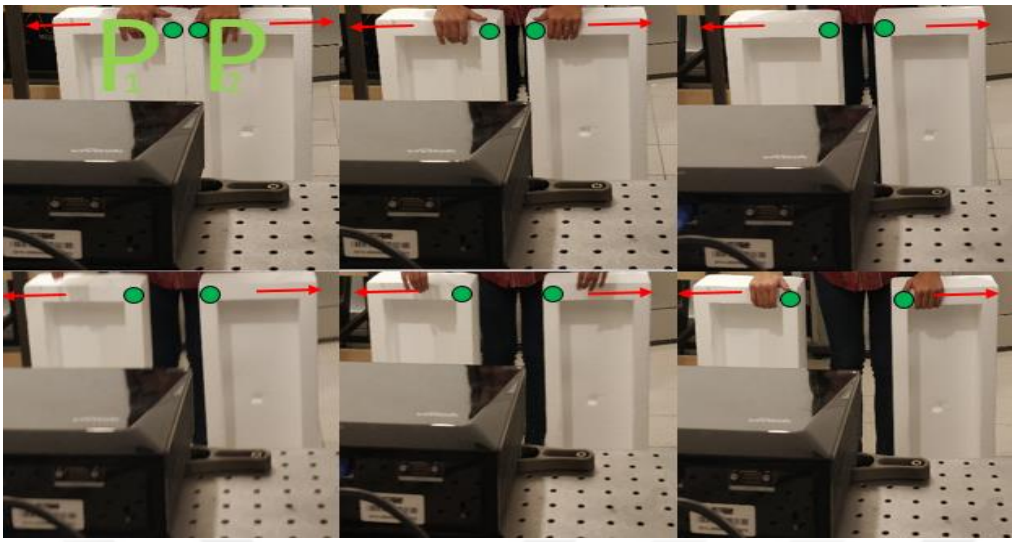


Figure 4.12 Specific two points is measured. Point P1 and P2 move apart from each other through in perpendicular to the Camera axis. Each time the depth values and differences are measured by hand and compared with reconstructed result.

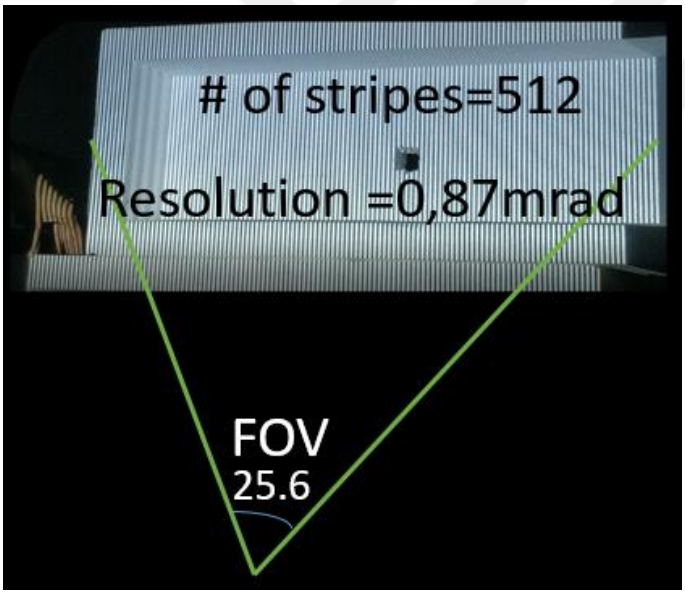


Figure 4.13 Theoritical resolution and measured resolution are detemined with the number of stripes and field of view of the projector. Here resolution in angle space is given. (Measured Resolution in X axis compatible with the actual Resolution ~0,87mrad).X-axis-distance resolution is equal with the angle resolution times distance.

CONCLUSION

In the first part, we build an experimental setup and measure distance of an object using Time-of-Flight measurement. After that, we expanded the measurements from a single point to 2D raster scan using a DMD chip. Using the DMD we succesffuuly obtain 3D image of a target scene. The appropriate wavelength for the project is found by simulating different atmospheric conditions. We provide a comparison of the atmospheric effects of two wavelengths in this range (905nm and 1550nm). After considering cost related issues, 905nm is chosen for our work. We continue with an investigation of 905nm photodiode technologies and conclude that a Geiger-Mode avalanche photodiode would yield better results, however it is a state-of-the-art technology, thus, currently difficult to supply. Therefore, we decided to use a linear mode avalanche photodiode in this work as the receiver unit.

To conclude this part, we cover Digital Micromirror Devices (DMD) by explaining the properties of the micro-mirrors such as modes, angular positions, resolutions, speeds etc. We state an innovative solution for a LIDAR scanning mechanism using a DMD which is designed for the experimental setup in ZEMAX. After successfully conducting a range finding experiment, we install the DMD setup and make 8x2, 16x2 and 8x4 fold scans to demonstrate that a 5cm distance between two obstacles can be detected by a DMD. Finally, a theoretical discussion of using diffraction order to achieve high intensity at desired spots is provided, however, it remains as future work.

In the second part, for high resolution we used “Structured Light” technique as a Machine Vision experiment. After a brief explanation of triangulation, the necessary calculations for the experiment are made by keeping the distance between the DMD and the camera fixed and solving for the angles of the triangle created by the object, the camera and the DMD. A 1080p camera is used to receive the data containing depth information which is post processed with some image processing techniques such as adaptive thresholding and median filtering. The reconstruction

algorithm, pattern creator and angle calculator functions are implemented in MATLAB. The texture data is also received with the same camera sensor and mapped to a point in the data cloud. Finally, we generate a 1920x1080 resolution depth map with texture mapping in under 5 seconds.

In conclusion, the utilization of a DMD as a scanner for a LIDAR sensor is experimentally demonstrated. The measured depth values and possible future works are provided. Also, a DMD is used as a structured light projector to create texture mapped 3D data clouds using Machine Vision. Satisfactory results are achieved by using both techniques.

REFERENCES

[1] M. Sun *et al.*, "time-based depth resolution," *Nature Communications*, vol. 7, no. May, pp. 1–6, 2016.

[2] "LIDAR drives forwards," *Nature Photonics*, vol. 12, no. 8, p. 441, 2018.

[3] "TEXASinstrumentLIDARpatentappUS20170003392A1.pdf." .

[4] "1 2 3 4," vol. 26, no. 7, p. 96864, 1987.

[5] J. F. Potter, "Two-frequency LIDAR inversion technique," no. 1.

[6] R. B. Smith and A. Carswell, "Differences arising in the determination of the atmospheric extinction coefficient by transmission and target reflectance measurements," 1986.

[7] M. Communications, "Mapping the world in 3D," vol. 4, no. July, pp. 3–4, 2010.

[8] "PrincetonLightwareGeigerUS20170350967A1.pdf." .

[9] D. Digital and A. Synthesis, "(12) Patent Application Publication (10) Pub. No.: US 2016 / 0377724 A1," vol. 1, no. 19, 2016.

[10] T. Yu, T. Yu, and A. Pacala, "(12) Patent Application Publication (10) Pub. No.: US 2016/0161600 A1," vol. 1, no. 19, 2016.

[11] H. Uhxfnpdqq, "< HDUV RI + LJK ' HILQLWLRQ ' 6FDQQLQJ + LVWRU \ 6WDWH RI WKH $ UW 2XWORRN," 1826.

[12] "Faceid_AppleUS20190041909A1.pdf." .

[13] "memsmirror_microsoftpatentUS20180172994A1.pdf." .

[14] G. L. Amidon *et al. , United States Patent (19) Fig-1*, no. 19. 1999.

[15] B. Jerome, "Application of Digital Micromirror Devices to Atmospheric LIDAR Measurement and Calibration," *PhD*, 2018.

[16] S. A. Goorden, A. P. Mosk, and J. Bertolotti, "device," pp. 1–9.

[17] M. Fornasier and H. Rauhut, "Compressive sensing," *Handbook of Mathematical Methods in Imaging: Volume 1, Second Edition*, vol. 24, no. July, pp. 205–256, 2015.

[18] D. Stuart and A. Kuhn, "Single-atom trapping and transport in DMD-controlled optical tweezers," *New Journal of Physics*, vol. 20, no. 2, 2018.

[19] D. Miyazaki, T. Honda, K. Ohno, and T. Mukai, "Volumetric display system

using a digital micromirror device based on inclined-plane scanning," *IEEE/OSA Journal of Display Technology*, vol. 6, no. 10, pp. 548–552, 2010.

[20] D. H. Lee, "Optical system with 4 μm resolution for maskless lithography using digital micromirror device," *Journal of the Optical Society of Korea*, vol. 14, no. 3, pp. 266–276, 2010.

[21] L. Hesselink, S. S. Orlov, and M. C. Bashaw, "Holographic data storage systems," *Proceedings of the IEEE*, vol. 92, no. 8, pp. 1231–1280, 2004.

[22] P. Nelson, "DLP ® Technology for Spectroscopy," no. August, pp. 1–9, 2016.

[23] L. A. Yoder *et al.*, "<title>DLP technolgy: applications in optical networking</title>," no. November 2001, pp. 54–61, 2001.

[24] S. K. Sathua, A. Dash, and A. Behera, "Removal of Salt and Pepper noise from Gray-Scale and Color Images : An Adaptive Approach," vol. 5, no. 1, pp. 117–126, 2017.

[25] D. Bradley and G. Roth, "Adaptive Thresholding using the Integral Image," *Journal of Graphics Tools*, vol. 12, no. 2, pp. 13–21, 2007.

[26] T. P. Kersten and D. Stallmann, "Automatic Texture Mapping of Architectural and Archaeological 3D Models," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXIX-B5, no. February, pp. 273–278, 2012.

[27] I. M. Vellekoop and A. P. Mosk, "Focusing coherent light through opaque strongly scattering media," vol. 32, no. 16, pp. 2309–2311, 2007.

[28] D. Stuart, O. Barter, and A. Kuhn, "Fast algorithms for generating binary holograms," vol. 211001, no. 2008, pp. 24837–24846, 2014.

[29] Jim Anderson, September 1, 2016, A Closer Look at 3D Imaging, Quality Magazine, <https://www.qualitymag.com/articles/93544-a-closer-look-at-3d-imaging>

[30] Slawomir Piatek, 18 September 2018, Hamamatsu LIDAR Webinar <https://hub.hamamatsu.com/us/en/webinar/hamamatsu-LIDAR-webinar/index.html>

[31] Page D, Koschan A, Abidi M. Reverse Engineering (2007) pp. 11-32 Published by Springer London

[32] <http://www.ti.com/dlp-chip/getting-started.html>

[33] Kersten, T. P. Stallmann, D., July 2012, Automatic Texture Mapping of Architectural and Archaeological 3D Models, ISPRS, vol. XXXIX-B5. <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXIX-B5/273/2012/>

[34] What is Infrared? , Retrieved from <https://www.flir.com/discover/what-is-infrared/>
[35] Tanaka Y. , Impact of near-infrared radiation in dermatology, World Journal of Dermatology (2012) 1(3) 30, DOI: 10.5314/wjd.v1.i3.30
[36] SD. Kaleem,8 August 2015, Removal Of Salt And Pepper Noise In Images Using Wavelet Transform And Median Filter, IJATES, Vol No.3. <http://www.ijates.com/images/short_pdf/1439132881_P77-89.pdf>
[37] Steve Eddins,25 July 2016, Adaptive thresholding for binarization. <https://blogs.mathworks.com/steve/2016/07/25/adaptive-thresholding-for-binarization/>
[38] M.C. Amann, T. Bosch, M. Lescure, R. Myllyla, and M. Rioux, "Laser ranging: a critical review of usual techniques for distance measurement", Opt. Eng. 40, 10–19 (2001).
[39] H.N. Burns, C.G. Christodoulou, and G.D. Boreman, "Sys− tem design of a pulsed laser rangefinder", Opt. Eng. 30, 323–329 (1991).
[40] J. Šíp, "Quenching circuits for Single Photon Avalanche Photodiodes", Proceedings of 8th conference Student EEICT 2002, Volume 3, VUT Brno FEEC 2002.
[41] V. Golovin, V. Saveliev, "Novel type of avalanche photodetector with Geiger mode operation", NIM A 518 (2004) 560.
[42] V. Saveliev, "The recent development and study of silicon photomultiplier", NIM A 535 (2004) 528.
[43] A. N. Otte et al., "Prospects of Using Silicon Photomultipliers for the Astroparticle Physics Experiments EUSO and MAGIC", IEEE Trans. Nucl. Sci. 53 (2006) 636
[44] M. Science, J. Marteau, D. Gibert, and K. Jourde, "arXiv : 1310. 4281v1 [physics. ins-det] 16 Oct 2013 Implementation of sub-nanoseconds TDC in FPGA : applications to time-of-flight analysis in muon," no. October, 2013.
[45] <https://www.semiconductorforu.com/what-is-avalanche-photodiode/.>
[46] Juan Carlos Fernandez-Diaz 1,2, William E. Carter 1,2, Ramesh L. Shrestha 1,2, and Craig L. Glennie, Remote Sens. 2014, 6, 9951-10001; doi:10.3390/rs6109951
[47] F. Brunoa, G. Biancoa, M. Muzzupappaa, S. Baroneb, A. V. Razionaleb "Experimentation of structured light and stereo vision for underwater 3D reconstruction." ISPRS Journal of Photogrammetry and Remote Sensing 00 (2010) 1–18
[48] Salvi, J.; Fernandez, S.; Pribanic, T.; Llado, X. A state of the art in structured light patterns for surface profilometry. Pattern Recognit. 2010, 43, 2666–2680.
[49] Christian Beder, Bogumil Bartczak and Reinhard Koch." A Comparison of PMD-Cameras and Stereo-Vision for the Task of Surface Reconstruction using Patchlets" Computer Science Department Universiy of Kiel Hermann-Rodewald-Str.3 24118 Kiel, Germany

[50] Jason Geng."Structured-light 3D surface imaging: a tutorial." Received September 22, 2010; revised December 10, 2010; accepted December 20,2010; published March 31, 2011 (Doc. ID 134160)
[51] Grant R. Fowles,2005,"Introduction to Modern Optics"
[52] A.Richardson. “Test of A/D Converters”. Test and Design-for-Testability in Mixed-Signal Integrated Circuits (pp.213-234)
[53] https://pinholeworks.com/wp/goerge-airy-vs-lord-rayleigh/
[54] http://www.mathworks.com/
[55] https://www.stemmer-imaging.com/fr-ch/donnees/cameras-3d-time-of-flight-cameras/

APPENDICES

A. Python Code for Controlling DMD

```
import pycrafter6500
import numpy
import PIL.Image
import time

# Enter Project Name

project_name = "anilpng"

# Enter Starting Point and Numer of Photos

starting_point = 1
number_of_photos = 400

# Enter Exposure Time
exposure = [1050]*number_of_photos
dark_time = [0]*number_of_photos
trigger_in = [False]*number_of_photos
trigger_out = [1]*number_of_photos

##################################################

project_size = "s_" + project_name

images = []
Name = "_Binary.bmp"

for counter in range(starting_point,starting_point + number_of_photos):
   images.append(numpy.asarray(PIL.Image.open("anil" + str(counter) + ".png")))
##    images.append(numpy.asarray(PIL.Image.open(str(counter) + Name)))
dlp = pycrafter6500.dmd()

dlp.stopsequence()

dlp.changemode(3)

##################################################
```

```
# Preprocessing Mode
##start = time.time()
##dlp.savesequence(images,exposure,trigger_in,dark_time,trigger_out,1,project_na
me,project_size)
##end = time.time()
##print(end - start)

# Uploading Mode

start = time.time()
dlp.usesequence(images,exposure,trigger_in,dark_time,trigger_out,0,project_name,
project_size)
end = time.time()
print(end - start)

dlp.startsequence()


import usb.core
import usb.util
import time
import numpy
import sys


##function that converts a number into a bit string of given length

def convlen(a,l):
    b=bin(a)[2:]
    padding=l-len(b)
    b='0'*padding+b

    return b

##function that converts a bit string into a given number of bytes

def bitstobytes(a):
    bytelist=[]
    if len(a)%8!=0:
        padding=8-len(a)%8
        a='0'*padding+a
    for i in range(len(a)/8):
        bytelist.append(int(a[8*i:8*(i+1)],2))

    bytelist.reverse()
```

```
    return bytelist

##function that encodes a 8 bit numpy array matrix as a enhanced run lenght encoded string of bits

def mergeimages(images):
    mergedimage=numpy.zeros((1080,1920,3),dtype='uint8')

    for i in range(len(images)):
        if i<8:
            mergedimage[:,:,2]=mergedimage[:,:,2]+images[i]*(2**i)
        if i>7 and i<16:
            mergedimage[:,:,1]=mergedimage[:,:,1]+images[i]*(2**(i-8))
        if i>15 and i<24:
            mergedimage[:,:,0]=mergedimage[:,:,0]+images[i]*(2**(i-16))

    return mergedimage

def encode(image):


## header creation
    bytecount=48
    bitstring=[]

    bitstring.append(0x53)
    bitstring.append(0x70)
    bitstring.append(0x6c)
    bitstring.append(0x64)

    width=convlen(1920,16)
    width=bitstobytes(width)
    for i in range(len(width)):
        bitstring.append(width[i])

    height=convlen(1080,16)
    height=bitstobytes(height)
    for i in range(len(height)):
        bitstring.append(height[i])


    total=convlen(0,32)
    total=bitstobytes(total)
```

```
for i in range(len(total)):


   bitstring.append(total[i])

for i in range(8):
   bitstring.append(0xff)

for i in range(4):    ## black curtain
   bitstring.append(0x00)

bitstring.append(0x00)

bitstring.append(0x02) ## enhanced rle

bitstring.append(0x01)

for i in range(21):
   bitstring.append(0x00)



n=0
i=0
j=0

while i <1080:
   while j <1920:
      if i>0 and numpy.all(image[i,j,:]==image[i-1,j,:]):
         while j<1920 and numpy.all(image[i,j,:]==image[i-1,j,:]):
            n=n+1
            j=j+1

         bitstring.append(0x00)
         bitstring.append(0x01)
         bytecount+=2

         if n>=128:
            byte1=(n & 0x7f)|0x80
            byte2=(n >> 7)
            bitstring.append(byte1)
            bitstring.append(byte2)
            bytecount+=2

         else:
```

```
                bitstring.append(n)
                bytecount+=1
            n=0
        else:
            if j<1919 and numpy.all(image[i,j,:]==image[i,j+1,:]):
                n=n+1
                while j<1919 and numpy.all(image[i,j,:]==image[i,j+1,:]):
                    n=n+1
                    j=j+1
                if n>=128:
                    byte1=(n & 0x7f)|0x80
                    byte2=(n >> 7)
                    bitstring.append(byte1)
                    bitstring.append(byte2)
                    bytecount+=2

                else:
                    bitstring.append(n)
                    bytecount+=1

                bitstring.append(image[i,j-1,0])
                bitstring.append(image[i,j-1,1])
                bitstring.append(image[i,j-1,2])
                bytecount+=3

                j=j+1
                n=0

            else:
                if j>1917 or numpy.all(image[i,j+1,:]==image[i,j+2,:]) or
numpy.all(image[i,j+1,:]==image[i-1,j+1,:]):
                    bitstring.append(0x01)
                    bytecount+=1
                    bitstring.append(image[i,j,0])
                    bitstring.append(image[i,j,1])
                    bitstring.append(image[i,j,2])
                    bytecount+=3

                    j=j+1
                    n=0

                else:
                    bitstring.append(0x00)
                    bytecount+=1
```

```
                toappend=[]


                while numpy.any(image[i,j,:]!=image[i,j+1,:]) and
numpy.any(image[i,j,:]!=image[i-1,j,:]) and j<1919:
                    n=n+1
                    toappend.append(image[i,j,0])
                    toappend.append(image[i,j,1])
                    toappend.append(image[i,j,2])
                    j=j+1

                if n>=128:
                    byte1=(n & 0x7f)|0x80
                    byte2=(n >> 7)
                    bitstring.append(byte1)
                    bitstring.append(byte2)
                    bytecount+=2

                else:
                    bitstring.append(n)
                    bytecount+=1

                for k in toappend:
                    bitstring.append(k)
                    bytecount+=1
                n=0
        j=0
        i=i+1
        bitstring.append(0x00)
        bitstring.append(0x00)
        bytecount+=2
    bitstring.append(0x00)
    bitstring.append(0x01)
    bitstring.append(0x00)
    bytecount+=3


    while (bytecount)%4!=0:
        bitstring.append(0x00)
        bytecount+=1

    size=bytecount

    print size
```

```
    total=convlen(size,32)


    total=bitstobytes(total)
    for i in range(len(total)):
        bitstring[i+8]=total[i]

    return bitstring, bytecount



##a dmd controller class

class dmd():
    def __init__(self):
        self.dev=usb.core.find(idVendor=0x0451 ,idProduct=0xc900 )

        self.dev.set_configuration()

        self.ans=[]

## standard usb command function

    def command(self,mode,sequencebyte,com1,com2,data=None):
        buffer = []

        flagstring=''
        if mode=='r':
            flagstring+='1'
        else:
            flagstring+='0'
        flagstring+='1000000'
        buffer.append(bitstobytes(flagstring)[0])
        buffer.append(sequencebyte)
        temp=bitstobytes(convlen(len(data)+2,16))
        buffer.append(temp[0])
        buffer.append(temp[1])
        buffer.append(com2)
        buffer.append(com1)

        if len(buffer)+len(data)<65:

            for i in range(len(data)):
                buffer.append(data[i])
```

```
            for i in range(64-len(buffer)):
                buffer.append(0x00)

            self.dev.write(1, buffer)

        else:
            for i in range(64-len(buffer)):
                buffer.append(data[i])

            self.dev.write(1, buffer)

            buffer = []

            j=0
            while j<len(data)-58:
                buffer.append(data[j+58])
                j=j+1
                if j%64==0:
                    self.dev.write(1, buffer)

                    buffer = []

            if j%64!=0:

                while j%64!=0:
                    buffer.append(0x00)
                    j=j+1


                self.dev.write(1, buffer)




        self.ans=self.dev.read(0x81,64)

## functions for checking error reports in the dlp answer

    def checkforerrors(self):
        self.command('r',0x22,0x01,0x00,[])
        if self.ans[6]!=0:
            print self.ans[6]
```

```
## function printing all of the dlp answer

    def readreply(self):
        for i in self.ans:
            print hex(i)

## functions for idle mode activation

    def idle_on(self):
        self.command('w',0x00,0x02,0x01,[int('00000001',2)])
        self.checkforerrors()

    def idle_off(self):
        self.command('w',0x00,0x02,0x01,[int('00000000',2)])
        self.checkforerrors()

## functions for power management

    def standby(self):
        self.command('w',0x00,0x02,0x00,[int('00000001',2)])
        self.checkforerrors()

    def wakeup(self):
        self.command('w',0x00,0x02,0x00,[int('00000000',2)])
        self.checkforerrors()

    def reset(self):
        self.command('w',0x00,0x02,0x00,[int('00000010',2)])
        self.checkforerrors()

## test write and read operations, as reported in the dlpc900 programmer's guide

    def testread(self):
        self.command('r',0xff,0x11,0x00,[])
        self.readreply()

    def testwrite(self):
        self.command('w',0x22,0x11,0x00,[0xff,0x01,0xff,0x01,0xff,0x01])
        self.checkforerrors()

## some self explaining functions

    def changemode(self,mode):
        self.command('w',0x00,0x1a,0x1b,[mode])
        self.checkforerrors()
```

```
    def startsequence(self):
        self.command('w',0x00,0x1a,0x24,[2])
        self.checkforerrors()

    def pausesequence(self):
        self.command('w',0x00,0x1a,0x24,[1])
        self.checkforerrors()

    def stopsequence(self):
        self.command('w',0x00,0x1a,0x24,[0])
        self.checkforerrors()


    def configurelut(self,imgnum,repeatnum):
        img=convlen(imgnum,11)
        repeat=convlen(repeatnum,32)

        string=repeat+'00000'+img

        bytes=bitstobytes(string)

        self.command('w',0x00,0x1a,0x31,bytes)
        self.checkforerrors()


    def
definepattern(self,index,exposure,bitdepth,color,triggerin,darktime,triggerout,patin
d,bitpos):
        payload=[]
        index=convlen(index,16)
        index=bitstobytes(index)
        for i in range(len(index)):
            payload.append(index[i])

        exposure=convlen(exposure,24)
        exposure=bitstobytes(exposure)
        for i in range(len(exposure)):
            payload.append(exposure[i])
        optionsbyte=''
        optionsbyte+='1'
        bitdepth=convlen(bitdepth-1,3)
        optionsbyte=bitdepth+optionsbyte
        optionsbyte=color+optionsbyte
        if triggerin:
```

```
        optionsbyte='1'+optionsbyte
    else:
        optionsbyte='0'+optionsbyte

    payload.append(bitstobytes(optionsbyte)[0])

    darktime=convlen(darktime,24)
    darktime=bitstobytes(darktime)
    for i in range(len(darktime)):
        payload.append(darktime[i])

    triggerout=convlen(triggerout,8)
    triggerout=bitstobytes(triggerout)
    payload.append(triggerout[0])

    patind=convlen(patind,11)
    bitpos=convlen(bitpos,5)
    lastbits=bitpos+patind
    lastbits=bitstobytes(lastbits)
    for i in range(len(lastbits)):
        payload.append(lastbits[i])



    self.command('w',0x00,0x1a,0x34,payload)
    self.checkforerrors()



def setbmp(self,index,size):
    payload=[]

    index=convlen(index,5)
    index='0'*11+index
    index=bitstobytes(index)
    for i in range(len(index)):
        payload.append(index[i])


    total=convlen(size,32)
    total=bitstobytes(total)
    for i in range(len(total)):
        payload.append(total[i])

    self.command('w',0x00,0x1a,0x2a,payload)
```

```
        self.checkforerrors()

## bmp loading function, divided in 56 bytes packages
## max  hid package size=64, flag bytes=4, usb command bytes=2
## size of package description bytes=2. 64-4-2-2=56

    def bmpload(self,image,size):

        t=time.clock()

        packnum=size/504+1

        counter=0

        for i in range(packnum):
            if i %100==0:
                print i,packnum
            payload=[]
            if i<packnum-1:
                leng=convlen(504,16)
                bits=504
            else:
                leng=convlen(size%504,16)
                bits=size%504
            leng=bitstobytes(leng)
            for j in range(2):
                payload.append(leng[j])
            for j in range(bits):
                payload.append(image[counter])
                counter+=1
            self.command('w',0x11,0x1a,0x2b,payload)


            self.checkforerrors()
        print time.clock()-t


    def savesequence(self,images,exp,ti,dt,to,rep,encodedimagesname,sizesname):

        self.stopsequence()

        arr=[]

        for i in images:
            arr.append(i)
```

```
    num=len(arr)

    encodedimages=[]
    sizes=[]

    for i in range((num-1)/24+1):
       print 'merging...'
       if i<((num-1)/24):
          imagedata=mergeimages(arr[i*24:(i+1)*24])
       else:
          imagedata=mergeimages(arr[i*24:])
       print 'encoding...'
       imagedata,size=encode(imagedata)
       numpy.savetxt( str(i) + encodedimagesname,imagedata,fmt='%d')
       sizes.append(size)

       encodedimages.append(imagedata)

    numpy.savetxt(sizesname,sizes,fmt='%d')
    numpy.save(encodedimagesname, encodedimages )

def usesequence(self,images,exp,ti,dt,to,rep,encodedimagesname,sizesname):

    self.stopsequence()

    arr=[]

    for i in images:
       arr.append(i)

    num=len(arr)

    encodedimages=[]
    sizes=[]

    sizes = numpy.loadtxt(sizesname, dtype=numpy.int32)

    for i in range((num-1)/24+1):

       if i<((num-1)/24):
          for j in range(i*24,(i+1)*24):
             self.definepattern(j,exp[j],1,'111',ti[j],dt[j],to[j],i,j-i*24)
       else:
          for j in range(i*24,num):
```

```
            self.definepattern(j,exp[j],1,'111',ti[j],dt[j],to[j],i,j-i*24)

encodedimages = numpy.load(encodedimagesname + ".npy")

self.configurelut(num,rep)

for i in range((num-1)/24+1):

    self.setbmp((num-1)/24-i,int(sizes[(num-1)/24-i]))

    print 'uploading...'
    self.bmpload(encodedimages[(num-1)/24-i],sizes[(num-1)/24-i])
```

B. Pixel Difference Method In MATLAB

```
clear all
clc

%% reading image from file,make it gray, apply theshold for binary
value
tic
B=zeros(450,653);
D=zeros(450,653);
for n=1:9
   jpgFileName =
strcat('C:\Users\User\Desktop\3dfotograflar\Reference3\_',
num2str(n), '.jpg');
   a(:,:,:) = imread(jpgFileName);
   p2=(a(:,:,3));
   p2=imresize(p2,0.25);

   p2=p2(1:450,268:920);

   c=double((2^(9-n)*7));
   d=double(2*c);
   p2=(bradley(p2, [c d], 0));
   A=p2*2^(9-n);
   B=B+A;



   jpgFileName =
strcat('C:\Users\User\Desktop\3dfotograflar\Photo3\_', num2str(n),
'.jpg');
   b(:,:,:) = imread(jpgFileName);
   p2=(b(:,:,3));
   p2=imresize(p2,0.25);
   p2=p2(1:450,268:920);

   p2=(bradley(p2, [c d], 0));

   C=(p2)*2^(9-n);
   D=D+C;

end



%%
Z=B-D;
Z=abs(Z);
Z=Z.*10;
```

```
Z(Z>500)=0;
Z=flipdim(Z,2);

for i=1:20
Z=medfilt2(Z,[1 16]);
end

%%

U=1:1:653;
V=1:1:450;

[X,Y]=meshgrid(U,V);
%%
Matrix(:,:,1)=X;
Matrix(:,:,2)=Y;
Matrix(:,:,3)=Z;
pcshow(Matrix);
time=toc;

colorbar('Ticks',[0,50,100,150,200,250,300,350],...

'TickLabels',{'0cm','2cm','4cm','6cm','8cm','10cm','12cm','14cm'})

     %%
```

C. Adaptive Threshold Method In MATLAB

1)

```
function bw=adaptivethreshold(IM,ws,C,tm)
%ADAPTIVETHRESHOLD An adaptive thresholding algorithm that
seperates the
%foreground from the background with nonuniform illumination.
%  bw=adaptivethreshold(IM,ws,C) outputs a binary image bw with the
local
%   threshold mean-C or median-C to the image IM.
%  ws is the local window size.
%  tm is 0 or 1, a switch between mean and median. tm=0
mean(default); tm=1 median.
%
%  Contributed by Guanglei Xiong (xgl99@mails.tsinghua.edu.cn)
%  at Tsinghua University, Beijing, China.
%
%  For more information, please see
%  http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm
if (nargin<3)
    error('You must provide the image IM, the window size ws, and
C.');
elseif (nargin==3)
    tm=0;
elseif (tm~=0 && tm~=1)
    error('tm must be 0 or 1.');
end
IM=mat2gray(IM);
if tm==0
    mIM=imfilter(IM,fspecial('average',ws),'replicate');
else
    mIM=medfilt2(IM,[ws ws]);
end
sIM=mIM-IM-C;
bw=im2bw(sIM,0);
bw=imcomplement(bw);
```

```
clear;close all;
im1=imread('page.png');
im2=imread('tshape.png');
bwim1=adaptivethreshold(im1,11,0.03,0);
bwim2=adaptivethreshold(im2,15,0.02,0);
subplot(2,2,1);
imshow(im1);
subplot(2,2,2);
imshow(bwim1);
subplot(2,2,3);
imshow(im2);
subplot(2,2,4);
imshow(bwim2);
```

2)

```
%BRADLEY local thresholding.
%   BW = BRADLEY(IMAGE) performs local thresholding of a two-
dimensional
%   array IMAGE with Bradley method. The key idea of the algorithm
is that
%   every image's pixel is set to black if its brightness is T
percent
%   lower than the average brightness of surrounding pixels in the
window
%   of the specified size, otherwise it is set to white.
%
%   BW = BRADLEY(IMAGE, [M N], T, PADDING) performs local
%   thresholding with M-by-N neighbourhood (default is 15-by-15).
The
%   default value for T is 10 and can be set in range 0..100. To
deal with
%   border pixels the image is padded with one of PADARRAY options
(default
%   is 'replicate').
%
%   Example
%   -------
%   imshow(bradley(imread('eight.tif'), [125 125], 10));
%
%   See also PADARRAY, RGB2GRRAY.

%   Contributed by Jan Motl (jan@motl.us)
%   $Revision: 1.1 $  $Date: 2015/04/19 17:03:01 $

function output = bradley(image, varargin)
% Initialization
numvarargs = length(varargin);      % only want 3 optional inputs
at most
if numvarargs > 3
    error('myfuns:somefun2Alt:TooManyInputs', ...
     'Possible parameters are: (image, [m n], T, padding)');
end

optargs = {[15 15] 10 'replicate'}; % set defaults

optargs(1:numvarargs) = varargin;   % use memorable variable names
[window, T, padding] = optargs{:};


% Convert to double
image = double(image);

% Local mean
mean = averagefilter(image, window, padding);
```

```
% Initialize the output to white color
output = true(size(image));

% Set a pixel to black if the image brightness
% is below (100-T)% of the average neighbourhood brightness
output(image <= mean*(1-T/100)
```

TEZ ŞABLONU ONAY FORMU
THESIS TEMPLATE CONFIRMATION FORM

16. Şablonda verilen yerleşim ve boşluklar değiştirilmemelidir.
17. Jüri tarihi Başlık Sayfası, İmza Sayfası, Abstract ve Öz'de ilgili yerlere yazılmalıdır.
18. İmza sayfasında jüri üyelerinin unvanları doğru olarak yazılmalıdır. Tüm imzalar mavi pilot kalemle atılmalıdır.
19. Tezin son sayfasının sayfa numarası Abstract ve Öz'de ilgili yerlere yazılmalıdır.
20. Bütün chapterlar, referanslar, ekler ve CV sağ sayfada başlamalıdır. Bunun için **kesmeler** kullanılmıştır. **Kesmelerin kayması** fazladan boş sayfaların oluşmasına sebep olabilir. Bu gibi durumlarda paragraf (¶) işaretine tıklayarak kesmeleri görünür hale getirin ve yerlerini **kontrol edin**.
21. Figürler ve tablolar kenar boşluklarına taşmamalıdır.
22. Şablonda yorum olarak eklenen uyarılar dikkatle okunmalı ve uygulanmalıdır.
23. Tez yazdırılmadan önce PDF olarak kaydedilmelidir. Şablonda yorum olarak eklenen uyarılar PDF dokümanında yer almamalıdır.
24. **Bu form aracılığıyla oluşturulan PDF dosyası arkalı-önlü baskı alınarak tek bir spiralli cilt haline getirilmelidir.**
25. Spiralli hale getirilen tez taslağınızdaki ilgili alanları imzalandıktan sonra, Tez Juri Atama Formu ile birlikte bölüm sekreterliğine teslim edilmelidir.
26. Tez taslağınız bölüm sekreterliğiniz aracılığıyla format ve görünüm açısından kontrol edilmek üzere FBE'ye ulaştırılacaktır.
27. FBE tarafından kontrol işlemleri tamamlanan tez taslakları, öğrencilere teslim edilmek üzere bölüm sekreterliklerine iletilecektir.
28. Tez taslaklarının kontrol işlemleri tamamlandığında, bu durum öğrencilere METU uzantılı öğrenci e-posta adresleri aracılığıyla duyurulacaktır.
29. Tez taslakları bölüm sekreterlikleri tarafından öğrencilere iletileceği için öğrencilerimizin tez taslaklarını enstitümüzden elden alma konusunda ısrarcı olmamaları beklenmektedir.
30. Tez yazım süreci ile ilgili herhangi bir sıkıntı yaşarsanız, **Sıkça Sorulan Sorular (SSS)** sayfamızı ziyaret ederek yaşadığınız sıkıntıyla ilgili bir çözüm bulabilirsiniz.

16. Do not change the spacing and placement in the template.
17. Write defense date to the related places given on Title page, Approval page, Abstract and Öz.
18. Write the titles of the examining committee members correctly on Approval Page. Blue ink must be used for all signatures.
19. Write the page number of the last page in the related places given on Abstract and Öz pages.
20. All chapters, references, appendices and CV must be started on the right page. **Section Breaks** were used for this. **Change in the placement** of section breaks can result in extra blank pages. In such cases, make the section breaks visible by clicking paragraph (¶) mark and **check their position.**
21. All figures and tables must be given inside the page. Nothing must appear in the margins.
22. All the warnings given on the comments section through the thesis template must be read and applied.
23. Save your thesis as pdf and Disable all the comments before taking the printout.
24. **Print two-sided the PDF file that you have created through this form and make a single spiral bound.**
25. Once you have signed the relevant fields in your thesis draft that you spiraled, submit it to the department secretary together with your Thesis Jury Assignment Form.
26. Your thesis draft will be delivered to the GSNAS via your department secretary for controlling in terms of format and appearance.
27. The thesis drafts that are controlled by GSNAS, will be sent to the department secretary to be delivered to the students.
28. This will be announced to the students via their METU students e-mail addresses when the control of the thesis drafts has been completed.
29. As the thesis drafts will be delivered to the students by the department secretaries, we are expecting from our students no to insist about getting their theses drafts from the Institute.
30. If you have any problems with the thesis writing process, you may visit our **Frequently Asked Questions (FAQ)** page and find a solution to your problem.

☒ **Yukarıda bulunan tüm maddeleri okudum, anladım ve kabul ediyorum. / I have read, understand and accept all of the items above.**

Name :
Surname :
E-Mail :
Date :
Signature : _____