Programmable Aperture Photography

An investigation into applications and methods

Presented by:
Ashill Chiranjan
CHRASH006

Prepared for:
Prof. Fred Nicolls
Department of Electrical Engineering
University of Cape Town

Co-Supervisor : Ritesh Kanjee (CSIR)

Submitted to the Department of Electrical Engineering at the University of Cape Town in fulfilment of the academic requirements for a Master of Science degree in Electrical Engineering

August 7, 2018

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.

3. This report is my own work.

4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

5. I have not previously, in its entirety or in part, submitted this dissertation for obtaining any qualification.

6. Parts of this work have been published as:

    a) A. Chiranjan, B. Duvenhage and F. Nicolls, *"Implementation of Adaptive Coded Aperture Imaging using a Digital Micro-Mirror Device for Defocus Deblurring"*, in 2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), Stellenbosch, 2016.

Signature:... Signed by candidate
Ashill Chiranjan at Pretoria

Date:..... 12/09/2018 .....

Acknowledgements

I would like to express my sincere gratitude to:

- My supervisor, Prof. Fred Nicolls, who provided the necessary guidance and support throughout the project.
- The Optronic Sensor Systems group within the Council for Scientific and Industrial Research for their support in terms of access to facilities, equipment, knowledge and funding.
- My friends and family for their continued support and motivation.

Abstract

The fields of digital image processing (DIP) and computational photography are ever growing with new focuses on coded aperture imaging and its real-world applications. Research has shown that coded apertures are far superior to traditional circular apertures for various tasks. A variety of coded aperture patterns have been proposed and developed over the years for use in various applications such as defocus deblurring, depth estimation and light field acquisition. Traditional coded aperture masks are constructed from static materials such as cardboard and cannot be altered once their shapes have been defined. These masks are then physically inserted into the aperture plane of a camera-lens system which makes swapping between different patterned masks difficult. This is undesirable as optimal aperture patterns differ depending on application, scene content or imaging conditions and thus would need to be changed quickly and frequently.

This dissertation proposes the design and development of a programmable aperture photography camera. The camera makes use of a liquid crystal display (LCD) as a programmable aperture. This allows one to change the aperture shape at a relatively high frame rate. All the benefits and drawbacks of the camera are evaluated. Firstly the task of performing deblurring and depth estimation is tested using existing and optimised aperture patterns on the LCD. A light field is then captured and used to synthesise virtual photographs and perform stereo vision. Thereafter, exposure correction is performed on a scene based on various degrees of illumination.

The aperture pattern optimised online based on scene content outperformed generic coded apertures for defocus deblurring. The programmable aperture also performed well for depth estimation using an optimised pattern and existing coded apertures. Using the captured light field, refocused photographs were constructed and stereo vision performed to accurately calculate depth. Finally, the aperture could adjust to the different levels of illumination in the room to provide the correct exposure for image capture. Thus the camera provided all the advantages of traditional coded aperture imaging systems but without the disadvantage of having a static aperture in the aperture plane.

Contents

List of Figures

List of Tables

CCD charge-coupled device

CMOS complementary metal oxide semiconductor

DIP digital image processing

DLP digital light processing

DMD digital micro-mirror device

DOF depth of field

EV exposure value

gigE gigabit ethernet

GUI graphical user interface

LCD liquid crystal display

LCoS liquid crystal on silicon

LED light emitting diode

MAP maximum a posteriori

MEMS micro-electro-mechanical system

MTF modulation transfer function

MURA modified uniformly redundant array

PSF point spread function

SDK software development kit

SLM spatial light modulator

SLR single-lens reflex

SNR signal-to-noise ratio

USB universal serial bus

Chapter 1

Introduction

This chapter first presents a background to the study. Thereafter the motivation and objectives of the work are outlined. Finally, the layout of the dissertation is provided.

1.1 Background to the study

The field of digital image processing (DIP) has very wide applications in numerous environments and almost every technical field today is impacted by DIP [1]. Associated closely with DIP is computational photography. This field of research comprises techniques in computational imaging to improve images that are taken using digital photography. Computational photography is a highly interdisciplinary field which utilises concepts and principles from engineering, physics, optics, mathematics, computer vision and image processing.

In a world that is three dimensional in nature traditional photography captures only two dimensions. This means that a great detail of depth information is lost. Digital cameras have limited depth of field and thus the parts of the image away from the plane of focus appear blurred. The use of advanced camera systems allows one to either capture an all-focus image, i.e. one that has a large depth of field (DOF), or extract depth from the image for the purposes of tracking or ranging. To record both simultaneously requires more extensive hardware [2]. There are many methods used to obtain or extract depth from an all-focus image but the one that will be investigated is the insertion of an optical element into the aperture of a lens. This alters the way in which light passes through the lens. The method has been termed coded aperture imaging.

1.2 Motivation for the study

Coded aperture imaging is the insertion of a patterned occluder within the aperture of a lens. This alters the incident light array so that the image captured by the sensor is not the final desired image but is coded in a way that requires further post computations. This facilitates the extraction of more information than if it had not been coded at all [2]. Coded apertures have been around for several years, with uses primarily being for depth estimation and defocus deblurring. Over the years, numerous aperture patterns have been designed and developed to help aid the image recovery process. Several patterns exist that have proven to be good for either defocus deblurring or depth estimation. Despite much research in the field, there is still no single pattern that has proven to be exceptionally good for both applications.

Most patterns are constructed from static materials, such as cardboard, and cannot be altered once their shapes have been defined. This means that prior selection of the pattern is important as it would need to be constructed before being used. The construction process of each pattern is a time consuming and laborious task. It is also not possible to construct certain patterns with cardboard e.g. concentric circles. The interchange of patterns within the lens is extremely time consuming with someone needing to physically remove and replace the old pattern with a new one each time. Given the large number of patterns available, choosing the correct pattern for a particular application is problematic. Each pattern has its own advantages and disadvantages. It is possible to construct several patterns beforehand and determine the best one for use based on real-time performance but this becomes difficult when imaging dynamic scenes. Since a dynamic scene is constantly changing, the optimal pattern for the scene might change too. This depends on scene content or imaging conditions. Thus being able to optimise or choose the optimal pattern for a scene beforehand becomes an issue. A reconfigurable or adaptive mask is needed to aid the pattern selection and construction process.

Programmable aperture imaging allows for the aperture pattern to be changed electronically within the lens of the camera without having to remove and replace any parts. This allows for a faster and more convenient interchange of patterns. It is achieved with the use of a reconfigurable mask technology whose shape can be changed in software. This allows for the generation of multiple patterns in real-time with no need for prior construction. It also enables the pattern to be optimised in real-time based on scene content. While there are numerous advantages of having an electronically controlled aperture mask, there are also many technological drawbacks. This dissertation investigates the advantages and disadvantages of having an adaptive mask in the aperture plane of a camera system.

1.3 Objectives of the study

Computational photography has become a popular research area with developments and advances occurring on a daily basis. Being able to extract more information from a conventional digital photograph can prove advantageous in many areas with major applications in military, civilian and domestic fields. There are several coded imaging systems that have been developed for the purpose of depth estimation or defocus deblurring, however there are also many problems associated with them. These include:

1. They are too expensive.
2. Their software is generally not open source.
3. They are designed for a specific application.
4. They have cheap sensors and poor batteries.
5. They are hard to configure and are not reconfigurable.
6. They are not user friendly and are complicated to use.
7. They are hard to acquire.

An example of a commercial computational photography camera is the Lytro ILLUM [3].

LYTRO: A case study

Lytro ILLUM is a $1300 light field camera that was designed, developed and produced by the company Lytro Inc. [3]. It has many significant imaging features that make it stand out from other commercial cameras. The most significant feature is the micro-lens array that is able to capture millions of light rays per image. The micro-lens array consists of 200,000 hexagonal lenses placed in front of the sensor to obtain slightly different perspectives of the same scene [4]. These are then combined to form an image that can be digitally refocused later. The camera is also able to sense depth and shift perspective in the image, giving the user much more information about the scene than a conventional two dimensional camera. While the camera configuration and function provided by the device are similar to this work, the investigations covered here focus on using coded aperture masks to achieve the same objectives rather than having a costly and cumbersome microlens array.

The aim of this system is to provide a technology that combats the issues mentioned above and implement a robust and user friendly system that can be used in any environment simply and effectively. Thus the objectives of this project are to:

Figure 1.1: Lytro ILLUM imaging system [3].

1. Provide a brief introduction into the field of computational photography and thereafter investigate the state-of-the-art in this field.
2. Investigate the field of coded aperture imaging and review results of existing imaging algorithms and aperture patterns.
3. Research the use of various adaptive aperture technologies within camera systems and their advantages and disadvantages i.e. their effectiveness at being reconfigurable.
4. Design and construct a coded photography camera with an electronically programmable aperture able to produce various aperture patterns.
5. Test existing algorithms and aperture patterns with the prototype camera.
6. Investigate novel uses for the programmable aperture camera with specific attention to optimising the aperture based on scene content.
7. Capture a light field with the prototype camera.
8. Find ways of best analysing the results obtained.
9. Critically evaluate the system as a whole.
10. Determine suitable recommendations for the prototype camera and future work that could be done to improve the system.

1.4 Scope and limitations

This study covers the design and development of a programmable aperture camera that can be used to enhance current optical systems. Given the large number of choices available in the design, the following limitations have been imposed:

1. Although many adaptive mask technologies are available, due to cost and availability only two were considered.
2. Components used for the construction of the camera were purchased as needed. However, preference was given to components that were already available as this saved on cost and manufacturer lead time.
3. Only greyscale still images were used in the experiments. No video or colour images were considered.
4. All experiments were conducted in a laboratory environment.

1.5 Overview

The remainder of the dissertation is arranged as follows.

Chapter 2 provides a comprehensive review of plenoptic and computational imaging. This is required to give the reader background knowledge on the mechanics behind coded aperture imaging and light field photography. A review of related work is given to provide insight into research being conducted by peers in the same field.

Chapter 3 details the design and development of the programmable aperture camera. The selection of the various components are first presented. Insight is provided on why those specific components were chosen. Thereafter, the integration of the components to create the camera are detailed. Finally the system functionality is evaluated.

Chapter 4 details the fundamentals behind coded aperture photography, light field capture and exposure correction. The theory and mathematical principles are provided to help gain insight in capturing and processing images from the programmable aperture camera detailed in chapter 3. First the notion of a point spread function (PSF) is reviewed. Thereafter the specifics behind defocus deblurring are presented. Next, depth estimation is detailed before light field capture is examined. Finally exposure correction is discussed.

Chapter 5 presents the experimental work carried out, the results obtained, and an analysis of the results. This chapter analyses all the uses for a programmable aperture mask. The chapter is divided into 5 experiments that were performed. Experiment 1 involved capturing PSFs that were needed for later experiments. Defocus deblurring was conducted in experiment 2, which consisted of optimising the aperture online based on scene content. TExperiment 3 tested depth estimation and optimising the aperture based on noise levels. Experiment 4 was used for capturing a light field. Experiment 5 involved testing the exposure correction algorithms.

Chapter 6 provides a summary of the work as a whole. This gives a review of the entire project and highlights the successes and failures of the work. It also provides recommendations and future improvements that can be done.

Chapter 2

Literature review

The following literature review provides an introduction into topics that are relevant to implementing programmable aperture photography. The review first provides a general overview into the topic with more specific details of the actual implemented system given later on. Design paradigms will be illustrated so as to motivate the design choices in the creation of the system. Existing projects with similar intentions are examined and comparisons made to establish which ideas were borrowed from existing projects.

2.1 Plenoptic imaging

The world is made up of objects and while these objects do not communicate their properties directly to an observer, they fill the space around them with a pattern of light rays that is perceived through the observer's visual system [5]. Conventional cameras have long been used to capture two dimensional images of our three dimensional world. In recent years a new form of imaging has been developed that attempts to modify the conventional camera to capture images that are richer in information. We call this plenoptic imaging. Computations are then applied to render improved images from the rich data [6]. To better understand plenoptic imaging scientists and engineers make use of the plenoptic function, which is an idealised concept that one can use to examine the structure of information that is potentially available to an observer by visual means [7].

The plenoptic function can be defined as a seven dimensional function representing the intensity or chromacity of the light observed at every position and direction in three dimensional space [8]. The plenoptic function is essential in bridging the gap between

three dimensional objects that occur in the world and the two dimensional images that we as humans perceive. According to geometric optics, i.e. light envisioned as rays, the plenoptic function is a complete description of visual information present and can be parametrised either by spherical or Cartesian parametrisation. The complete seven dimensional Cartesian parametrisation is given by

$$P(x, y, z, \theta, \phi, \lambda, t), \tag{2.1}$$

where (x, y, z) are the coordinates of a point in three dimensional space, λ is the wavelength, t is the time observed and (θ, ϕ) indicates the angular direction of the light ray. The plenoptic function essentially measures the radiance per unit wavelength at every three dimensional spatial location. In real world applications we do not measure the plenoptic function directly, as this would require enormous resources, but rather through a plenoptic sampling basis, which are slices of the plenoptic function with different names. We call a two dimensional spatial slice (x, y) an image, a three dimensional spatio-temporal slice (x, y, t) a video, a three dimensional spatio-spectral slice (x, y, λ) a multi spectral volume, a four dimensional spatio-angular slice (x, y, θ, ϕ) a light field, and so on [9].

We can reduce this seven dimensional parameterisation to a more compact five dimensional form if we ignore the effects of time and colour (wavelength) and utilise angular coordinates for the light intensity at every x and y viewing position. This allows the plenoptic function to take the form

$$P(x, y, z, \theta, \phi), \tag{2.2}$$

The five dimensional plenoptic function is illustrated in figure 2.1. The figure shows the three spatial coordinates and the two angular dimensions that comprise the plenoptic function.

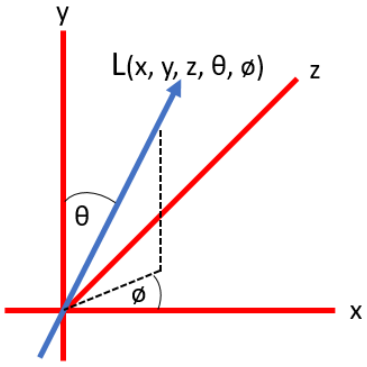


Figure 2.1: Three spatial coordinates and two angular dimensions characterising the five dimensional plenoptic function.

The manner in which the plenoptic function can be captured, processed and stored is summarised in figure 2.2. As described by Pereira et al. [5], the first step is to acquire some version of the plenoptic function using an acquisition model of choice (some sensor). The sensor may directly measure the data for the final representation and coding formats or there may be some sampling required. With so many representation formats, the data to be displayed is no longer the same as the data acquired. Thereafter an appropriate coding standard is needed to store the information in a format that can be easily accessed. Finally the image is rendered, i.e. decoded, to be displayed on a screen which produces light that can be viewed by an observer.

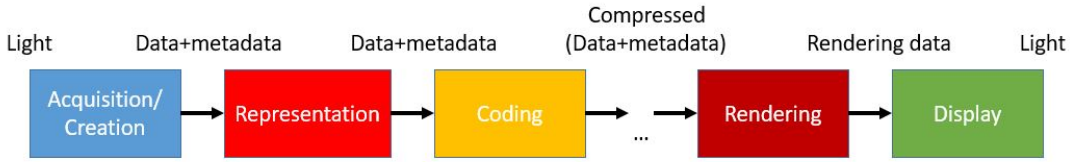


Figure 2.2: Flow diagram of the plenoptic imaging process [5].

2.2 Computational imaging

Computational imaging is an emerging field created by the convergence of computer graphics, computer vision and photography [10]. Computational imaging involves the use of a computational camera and computational decoding to produce images that are superior to and more useful than conventional images of the same scene. To differentiate computational imaging from conventional imaging one first needs to define and distinguish the conventional camera from a computational camera.

2.2.1 Conventional camera

A camera, in simple terms, is a device that captures light from scenes and is able to take a three dimensional representation of our world and store or project it using a two dimensional platform. The traditional camera is standard in that it contains a lens which focuses incoming light rays from a scene onto a detector, and is restricted in that it only captures those principal rays that pass through its centre of projection [11]. The image sampled by the detector is often the final version of the image, which can be digitally or computationally edited. This is usually a cosmetic enhancement where no more information is extracted from the captured image. A simple diagram is shown in figure 2.3 (b).

In conventional cameras, the sampling basis must consist of regularly spaced orthogonal sampling functions. This ensures that the spatial information in the plenoptic function can be interpreted directly from pixel measurements once the scale and orientation of the camera are determined [9].

2.2.2 Computational camera

A computational camera is able to utilise innovative optics and techniques to capture images which can later be modified computationally. Using a host of DIP methods, a final image can be extracted from the original captured image. The captured image is in its raw, unprocessed form and may not be meaningful up until the point the information is decoded or extracted [11]. This final image contains more information or is of a superior quality than the original captured image taken from a conventional camera.

A computational camera can have a general sampling basis. However, the choice of optics and detector determines the sampling basis. This in turn determines the way visual information is coded in the pixel measurements [9]. The mapping of spatial plenoptic information and pixel coordinates is flexible and computational cameras are able to capture different slices of the plenoptic function.

Figure 2.3 is a simple block representation of a traditional camera and a computational camera [12]. Figure 2.3 (a) is a typical setup for capturing an image of a conventional scene. Figure 2.3 (b) illustrates the traditional camera model that captures only those principal rays that pass through its centre of projection. Figure 2.3 (c) is a computational camera that alters light rays to optically code the image which is computationally decoded to produce the resultant final image.

There are various designs and configurations that computational cameras can take, each with their own advantages and disadvantages. The coding methods used in computational cameras can be broadly classified into five categories [11]:

1. **Object-side coding**
   In this configuration, external devices (optics) are attached to the camera without modifying it in any way. This puts the optical element between the camera and object. The light ray cones emerging from the various objects being imaged at different field angles will intersect with the optical element in different areas, yielding a spatially varying modulation if the surface profile is not homogeneous [13].

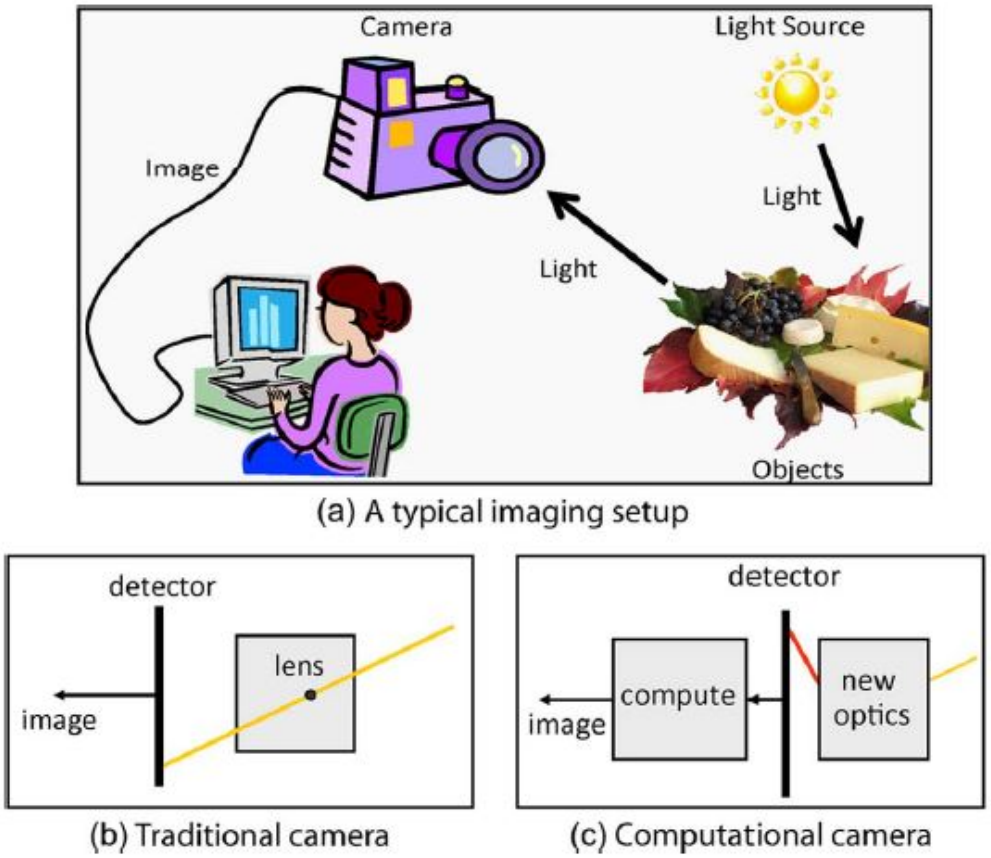


Figure 2.3: Imaging through a traditional camera and computational camera [12].

2. Pupil plane coding

Pupil plane coding involves placing the optical element in or very near to the pupil plane of a traditional camera system. All light rays from objects being imaged will pass through the pupil plane, which can thus be used to provide spatially-invariant light modulation and manipulate the system PSF [13]. The use of coded apertures is a form of pupil plane coding where the coded mask is placed in the aperture plane of a camera-lens system.

3. Focal plane coding

In this configuration, the optical element is placed on or near to the camera sensor and is sometimes also referred to as sensor-side coding [13]. The result is similar to object-side coding with an added advantage of the optical element being within the camera-lens system. Sensor-side cameras can be used to capture light fields and thus can be envisioned as a form of plenoptic camera.

4. Illumination coding

Illumination coding involves the use of a light source (e.g. a flash) where the intensity is modulated over time to light up an object in a specific illumination pattern. This allows the camera to capture images of objects under specific illumination patterns. These images now contain more scene information than those captured with a traditional flash that can be extracted beneficially [13].

5. **Camera clusters and arrays**

Camera clusters or arrays use several cameras placed next to or close to one another so that they are able to capture images of the same scene from slightly different angles. Images can be combined to form higher quality images or used in stereo vision to infer depth. This overcomes the limitations associated with the use of a single camera.

2.3 Related work

There are several contributions made to computational and plenoptic imaging over the years. This section presents related work in these fields.

1. In Levin et al. [2] a conventional camera is modified slightly to allow for the capturing of a high-resolution image and the extraction of depth information. They insert a patterned occluder into the aperture plane of a camera and lens system to alter the way that light falls on the camera sensor. The aperture mask pattern is optimised for depth extraction by maximising the zero-crossings of the filter in the Fourier domain but it can also be used to deblur an image. A novel deconvolution algorithm is proposed that uses a basis of natural image priors to find the sharp image from the blurred one.

2. Zhou and Nayar [14] develop a novel mathematical model to create a unique coded aperture mask that optimises the defocus deblurring process and produce a reconstructed sharp image of good quality. A genetic algorithm is used to choose an optimal aperture pattern. Several patterns are developed for different levels of image noise. The overall optimal pattern is then compared to other coded apertures. The method of obtaining the optimal aperture pattern is adopted and used in this research.

3. Wilson and Nicolls [15] conducted research in the fields of coded apertures, coded exposure and coded illumination. The research was developed and built on the works of Levin et al. [2] and Zhou and Nayar [14]. A coded aperture camera was constructed from a lens with its diaphragm removed and custom coded apertures, made from cardboard, were inserted into the aperture of the lens to encode images before reaching the image sensor. Existing aperture patterns were tested and compared to a novel aperture pattern that was developed to correct for defocus blur and to estimate depth. Combining the same camera and different apertures, the authors were also able to capture various light fields. Using these light fields and aggregation techniques, they were able to reconstruct a multiplexed scene. These are adapted and used in this research. Work was also done in the fields of coded exposure and coded illumination but will not be used.

4. Nayar, Braznoi and Boult [16] make use of a programmable imaging system that utilises a DMD (explained further in section 3.4.1) for image capture. They test various optical uses for the DMD, such as high dynamic range imaging and optical appearance matching. Results are then obtained and explained further. The use of a DMD as an spatial light modulator (SLM) is explored in later chapters.

5. Nagahara et al. [17] develop a novel imaging system that makes use of a liquid crystal on silicon (LCoS) as a programmable coded-aperture mask. The LCoS is placed in the aperture plane of the novel lens system and various aperture patterns are generated on this reflective type of SLM. This allowed the authors to produce any aperture pattern quickly. The system produced good images but suffered from several optical aberrations. The system was also fairly large and complex. The methods used are considered and adapted.

6. In Suh et al. [18] an imaging system is developed that utilises an LCD that acts as an SLM to produce a programmable aperture system. This system is then used to create an interactive three dimensional sensing system for human-computer interaction (HCI). The images produced from this setup are poor as the authors have chosen to use a lens-less system. Instead they have used modified uniformly redundant array (MURA) patterns that do not require lenses to produce images. The LCD also caused a large amount of light loss and diffraction artefacts.

Chapter 3

Programmable aperture camera design and development

This chapter deals with the design and construction of a programmable aperture photography camera. The section starts with a general overview of the system. Thereafter, the selection of all the components are discussed. The interconnection of the components are then detailed. Finally a testing of camera functionality is evaluated.

3.1 General system overview

Before any specific hardware was considered, a conceptual design of the entire system was done to try and get a general overview of how it would function. Figure 3.1 gives an abstracted view of the system and the hardware required. From the diagram we see the main components needed to construct a programmable aperture camera are:

1. **Lenses**
   These direct light towards the aperture and onto the camera.

2. **A camera**
   This will be used to capture the images required.

3. **An spatial light modulator (SLM)**
   This will be used as the programmable aperture to generate aperture patterns using software.

Although the system is made up of only a few components, care needs to be taken to select the best and most practical parts for the project. All the components should be easy to use, low-cost, well documented and modifiable, while at the same time able to produce high quality images.

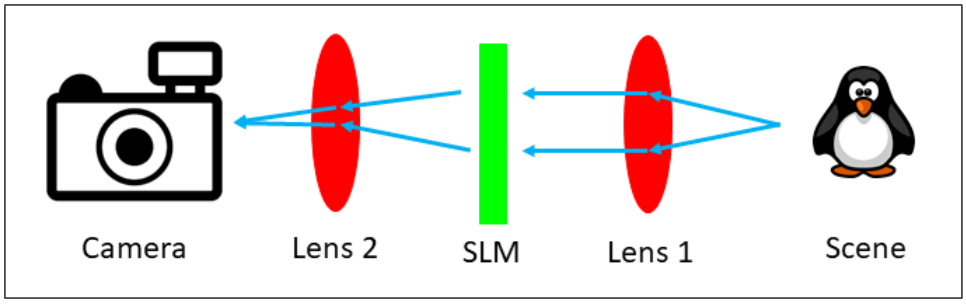


Figure 3.1: General system overview.

3.2 Camera

The camera is an extremely important piece of hardware in the system being developed. It consists of an image sensor that, along with other electronics, converts light energy to an electrical signal. The camera ultimately determines the quality of the images captured so choosing the correct camera with the right resolution is important. The factors that needed to be considered before the camera would be selected were:

1. High resolution.
2. Inexpensive.
3. Easily available.
4. Manual control.
5. Provide good quality images.

There are numerous cameras available today that satisfy the requirements listed above. Since it would not be feasible to review all of them only a few will be selected and compared. The cameras considered are listed in table 3.1. These range from single-lens reflex (SLR) to gigabit ethernet (gigE) cameras.

The Prosilica GC1380 offered the lowest resolution but its high frame rate made it attractive. The global shutter was an advantage in that there were no mechanically moving parts. It carried a high price tag with only monochrome images on offer. The Stingray

Table 3.1: Camera comparison.

Type	Name	Shutter type	Resolution	Price	FPS	Colour/Mono
SLR	Nikon D7500	Mechanical	20.9 MP	R25000	8	Colour
gigE	Prosilica GC1380	Global	1.4 MP	R30000	20.2	Mono
SLR	Canon EOS 500D	Mechanical	15.1 MP	R6000	3.4	Colour
gigE	Stingray f504	Rolling	5 MP	R15000	9	Mono

f504, while in a similar league, offered a higher resolution at a lower price but with a lower frame rate. Despite the price being lower than the Prosilica, it was still too expensive for the task at hand.

The camera chosen to be used was the Canon EOS 500D. It was the cheapest, offered a good resolution and produced images in both monochrome and colour. The Nikon did offer a higher resolution and frame rate but at a higher price. Thus the Canon EOS 500D was the most practical choice.

3.3 Lens

The main purpose of the lens would be to gather light rays from the source, direct them onto the LCD, such that the light can be modulated and thereafter converge them onto the camera sensor. The minimum requirements of the lens would be:

1. Easy to mount onto camera.
2. Inexpensive.
3. Easily available.
4. Manually focusable.
5. Easy to disassemble.

The EF lens mount is the standard used by the Canon EOS family of cameras. The lens would need to be of the EF series so that it is able to fit onto the camera. There are several lenses available that fall into this category. A few are presented in table 3.2.

From table 3.2 we can see that an increase in weight of the lens also leads to an increase in price. Ideally, the lens should be lightweight and cheap as it would be modified for

Table 3.2: Lens comparison.

Name	**Focal Length**	**Manual Focus**	**Weight**	**Price**
EF 50mm	50mm	Yes	162g	R1600
EF 75-300mm	75-300mm	No	480g	R3000
EF 24-105mm	24-105mm	Yes	795g	R15000

use. The lens chosen was the Canon EF 50mm f/1.8 II. It was cheap and could be easily modified.

3.4 Spatial light modulator (SLM)

Coded aperture masks can be created from various materials, of which each material has its own advantages and disadvantages. These materials can be divided into static or dynamic types, depending on their ability to display either a single pattern or multiple patterns. Static masks can only be encoded with a single pattern at a time and keep that shape for the entire duration of their life cycle. Dynamic or reconfigurable masks are able to switch between multiple patterns easily and efficiently.

Choosing the correct adaptive mask technology is important. The adaptive mask ultimately impacts the quality of the images that will be detected and the amount of information that can be extracted from the captured image. Before choosing the most appropriate adaptive mask technology, one needs to consider various factors such as:

1. The patterns that will be displayed on the mask.
2. The physical characteristics and properties of the adaptive mask technology.
3. Cost.
4. Local availability.
5. Ease of use.

The adaptive mask technology should be able to display a wide range of aperture patterns but also have good physical characteristics such as high frame rate, good resolution and sufficient light transmissivity (i.e. allow a lot of light to pass through to the detector) among other characteristics. The adaptive mask will serve as an spatial light modulator (SLM). An SLM is a device that is able to modulate or change the amplitude, phase or

polarisation of light waves in space and time [20]. There are several different technologies that could be used as SLMs but due to the factors stated above only two devices will be considered. These are the liquid crystal display (LCD) and the digital micro-mirror device (DMD).

3.4.1 Digital micro-mirror device (DMD)

A DMD is a device consisting of thousands of tiny mirrors that are arranged in a rectangular pattern and individually controlled. The DMD is a reflective SLM. The mirrors can be controlled to tilt at ± 10-12°, i.e. either to an on or an off position.

Each mirror is independently controlled by loading data onto the complementary metal oxide semiconductor (CMOS) memory cell to steer the reflected light [21]. Each mirror is typically 13.7 μm square and represents a single pixel, with an array of pixels usually arranged to have a resolution of 1024 × 768. The mirrors are fairly reliable and the hinges do not fatigue or get biased over time [22]. They have a high fill factor of 90% or greater with infinite contrast ratio. Apart from the mirror chip itself, control hardware is needed to set the state of the memory array and the control bus [22].

Figure 3.2 shows the structure of a DMD device. Each mirror is attached to the top of a CMOS substrate cell which can electrically control the angle of the mirror by means of polarity, and the hinge allows the mirror to freely tilt at ± 10-12°.

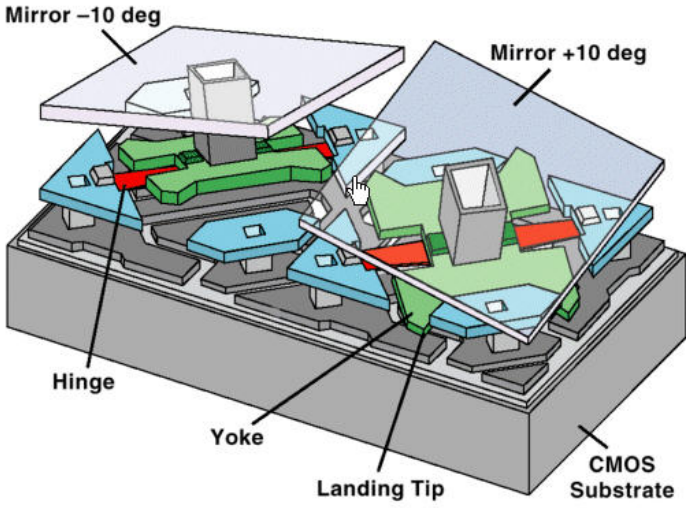


Figure 3.2: Structure of a DMD [23].

DMDs are most commonly used in video projectors due to their physical properties. They

are preferred over LCD or LCoS technology as they have higher frame rates and allow more light throughput.

3.4.2 Liquid crystal display (LCD)

The term liquid crystal is used to describe a substance in a state between liquid and solid but which exhibits the properties of both [26]. Liquid crystals contain molecules that are able to arrange themselves and all point in a particular common direction. LCDs are the most common display type today, both for projection and direct viewing, because they are compact, inexpensive and have good optical properties [22]. LCDs are affected by an electric current, and by varying the electrical current passed through them we are able to exploit their chemical and physical nature which in turn affects the way light passes through them.

There are two configurations for the molecules in a liquid crystal display, namely nematic phase and twisted-nematic phase. In nematic phase, the molecules in the display are not spatially fixed but align themselves with each other. In twisted-nematic phase, the display crystals are configured between two linear polarisers which are perpendicular to each other [22].

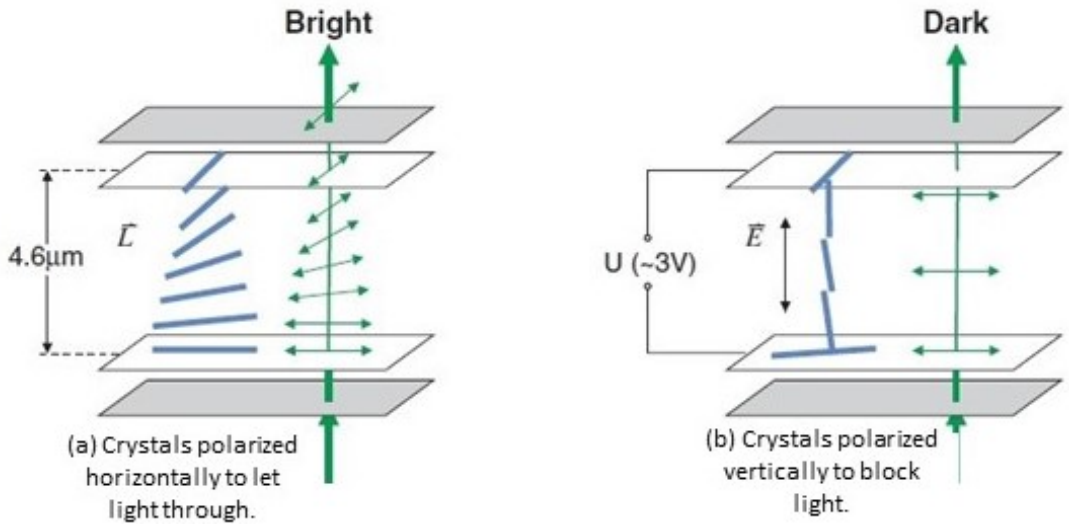


Figure 3.3: The polarisation of the liquid crystals in an LCD screen [26].

Because of the robustness of LCD technology they are used in many different applications such as in cell-phones, televisions and computer screens as well as various other electronic devices.

3.4.3 Comparison between LCD and DMD technologies

The DMD and LCD technologies are very different in the way that they operate. Each SLM has its own advantages and disadvantages. Since only one mask is used in the camera, they need to be compared to determine the most appropriate for the task. Table 3.3 compares the properties of both.

Table 3.3: Comparison between LCD and DMD technologies.

	DMD	**LCD**
Light throughput efficiency	High (>90%)	Low (<40%)
Contrast ratio	High (1200:1)	Medium (4500:1)
Frame rate	High (5-10 KHz)	Medium (0.5-1 KHz)
Resolution	Up to 1080p (1920×1080)	VGA (640×480)
Optical design	Complex optics needed	Simple optics needed
Optical aberrations	Severe diffraction [25]	Little to no diffraction
Cost	R50 000 - R150 000	R300 - R1000
Support	Little support available	Good support available
Ease of use	Complicated to use	Very easy to use
Availability	Not locally available	Locally available
Size	Small ($\pm$10mm^2)	Small (sizes vary)

The DMD, on paper, looks like the more attractive option due to its high reflectance, frame rate and resolution, but the main problem is the diffraction that is caused by the micro mirrors. The diffraction severely degrades the quality of the image, and although it can be corrected, the correction process is also responsible for degrading the image. Thus the added advantages of coded apertures are nullified by the diffraction correction process. The DMD, in its current state, is not suitable as an adaptive coded aperture mask technology.

The LCD, while not perfect, does not suffer from severe diffraction artefacts. It does have very good properties and being cheap and easily available does make it the more practical choice. Despite the many shortcomings, these can be addressed and are explained further in table 3.4.

Some of the solutions listed in table 3.4, such as increasing camera exposure and gain, will increase the presence of noise in the image but this can be easily addressed using techniques described in later chapters. Therefore, after much consideration, the LCD was selected as the adaptive coded aperture mask technology.

Table 3.4: Solutions to LCD shortcomings.

Problem	Solution
Low light throughput	Increase camera exposure or gain
Fragile	Enclose in protective case
Light leak	Image subtraction to compensate for extra light
Optical aberrations	Post capture correction

3.4.4 Conclusion

The LCD replaces the traditional hexagonal aperture in the lens to allow for the generation of coded apertures in real time. Since only binary (open and closed) apertures are examined and tested, the LCD only needs to display monochrome images. There are several other requirements that must be considered before the correct LCD can be chosen. These include:

1. Physical dimensions should allow for the LCD to mount easily into the lens.
2. Highly transparent to allow for minimal light loss.
3. High contrast between opaque and transparent areas.
4. High resolution to allow for the generation of many aperture shapes.
5. Easy to program.
6. Cheap and easily available.

After considering all the above factors, the EA DOGM132W-5 1-bit LCD was chosen. The main contributing factor to choosing this LCD was the development board that can be easily purchased with the device. This allows for rapid prototyping without the need for the purchase of a separate micro controller. The technical specifications of the LCD are listed in table 3.5.

The development board that was also purchased was the EA 9780-2USB test board. It supports the entire EA DOG range and is USB compatible. It allows for easy connection and rapid prototyping. This saved on time and development costs which proved ideal for the project.

Table 3.5: LCD specifications.

Manufacturer	Electronic Assembly
Series	DOGM132W-5
Physical size	55 × 31 mm
Frame rate	20KHz
Pixel size	0.4 × 0.35 mm^2
Resolution	132 × 32
Operating voltage	3.3V
Interface	SPI
Display colour	Black on White
Price	R300

3.5 Computing platform

The computing platform needs to be able to perform several tasks. These range from aperture pattern generation, camera triggering, image processing and storage of image data. The computer should allow the user to have seamless control of the programmable aperture camera system while also restricting the user to certain actions so that the system is not damaged.

In summary, the computing platform needs to meet the following minimum requirements:

1. Easy to use graphical user interface (GUI).
2. Intel core i7 processing power.
3. Sufficient storage space. (>20GB).
4. Sufficient RAM (>4GB).
5. Ethernet and universal serial bus (USB) compatibility.

Based on the above criteria, the laptop owned by the author is sufficient as it meets all the above requirements and saves on costs and time to purchase. Therefore the author's laptop was used for this project.

3.6 Interconnection and assembly

In order to reach the aperture plane of the lens, one needs to disassemble the lens. Fortunately a detailed description of how to disassemble the lens is given by Wilson and

Nicolls [15]. This was used to modify the lens to allow for the placement of the adaptive mask. The only components that were needed were the front and back optical lenses, the front ring and the housing for the lens as shown in figure 3.4.

(a) Front focal lens. (b) Back focal lens and housing. (c) Front focal lens ring.

Figure 3.4: Different parts of the Canon EF 50mm f/1.8 II lens.

The objective was to place the LCD between the front and back focal lenses. Figure 3.5 shows the placement of the LCD in front of the back focal lens within the housing. Part of the housing on the side had to be removed to allow for the LCD lead pins to be reached and connected.

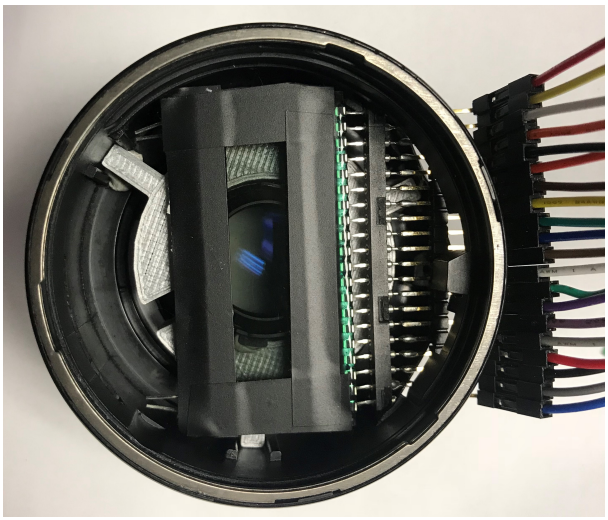


Figure 3.5: LCD placed in front of back focal lens within housing.

Figure 3.6 shows the complete coded aperture camera from two different views. We can see the LCD connected directly to the development board via 20 colour coded jumper cables. The development board is connected via USB to the computing platform. This allows for the pattern on the LCD to be quickly and easily changed without having to tamper with any component within the camera as it is all done in software.

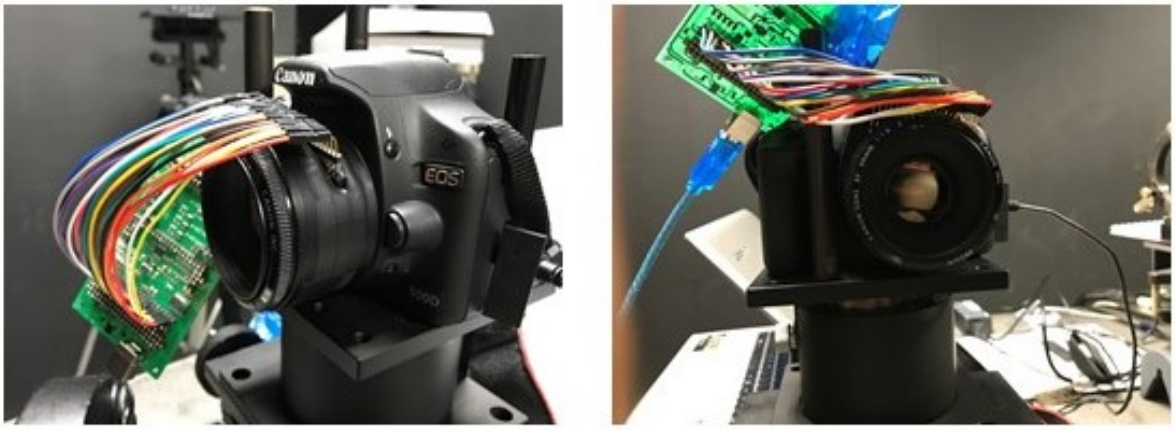


Figure 3.6: Constructed coded aperture camera with all components.

3.7 Calibration and validation

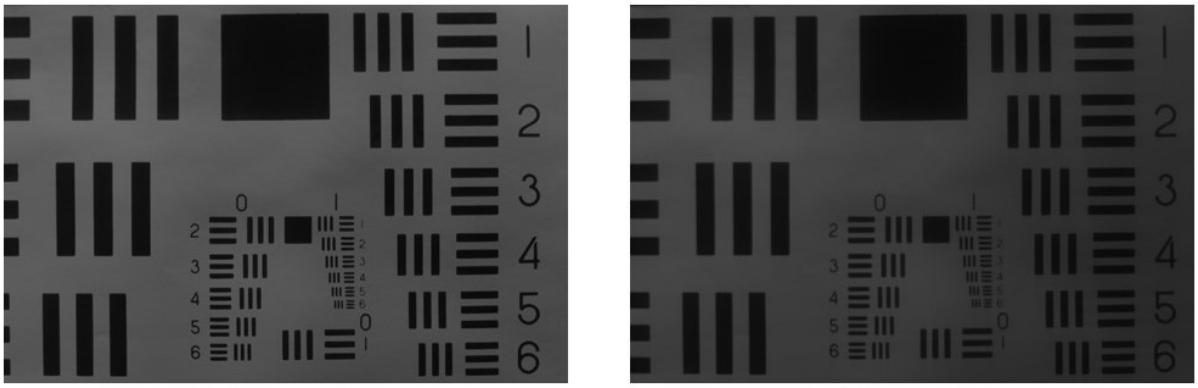
Before the camera performs coded aperture photography, certain parts of the system are tested. A few camera parameters are also calculated. These are described below.

3.7.1 Light throughput efficiency

The LCD, by nature, is a transmissive medium and not all light that reaches the LCD will pass through it. Due to the actual material by which the LCD is made, as well as the electronic components embedded within the device, a good portion of the light is absorbed or reflected. In order to determine how much of light is actually absorbed by the LCD a simple test was conducted which compared the light throughput of the camera with and without the LCD.

The test involved a camera with a standard circular aperture within the lens. Two pictures were taken of the same scene using the same aperture size, exposure and ISO setting but one included the LCD in the aperture plane and the other did not. Since only the LCD was being varied, the size of the aperture as well as the ISO and exposure settings were irrelevant. Figure 3.7 shows the images with and without the LCD.

To determine the light loss, the pixel intensities of each image were added together and compared. The pixel intensities are unsigned integer values in the range 0 (black) to 255 (white). For simplicity, the values are normalised to lie between 0 and 1. The resolution of each image is 2304×3456. This gives a total of 7 962 624 pixels. Table 3.6 shows the



(a) Without LCD. (b) With LCD.

Figure 3.7: Images of a standard USAF 1951 resolution chart.

comparison of the sum of the pixel intensities with and without the LCD.

Table 3.6: Image pixel intensities with and without LCD.

	LCD	**No LCD**
Sum of pixel intensities	7.6×10^5	2.1×10^6

If we compare the ratios of pixel intensities, we see that the LCD is only 0.3595 or 36% efficient. This means that 64% of the light is absorbed or reflected. Although this is a problem, a simple solution would be to increase to the exposure time of the camera such that sufficient light is able to reach the camera sensor.

3.7.2 Light leak compensation

As explained in section 3.7.1 above, the LCD is at best 35-40% transparent. Just as the LCD cannot let through all the light that reaches its surface, it also cannot block all the light from passing through. Therefore the LCD is never truly opaque.

A simple solution to this problem is to capture an extra image with all the LCD pixels 'off' and subtract this image from the desired image. This is illustrated in figure 3.9. The light transmitted in both states is subtracted and the final image is the result of just the aperture shape and size.

Figure 3.10 (a) shows an image captured with a circular aperture. The camera settings remained the same and another image was captured but this time with all the LCD pixels 'off'. This is shown in figure 3.10 (b). Figure 3.10 (c) shows the resulting image once (b)

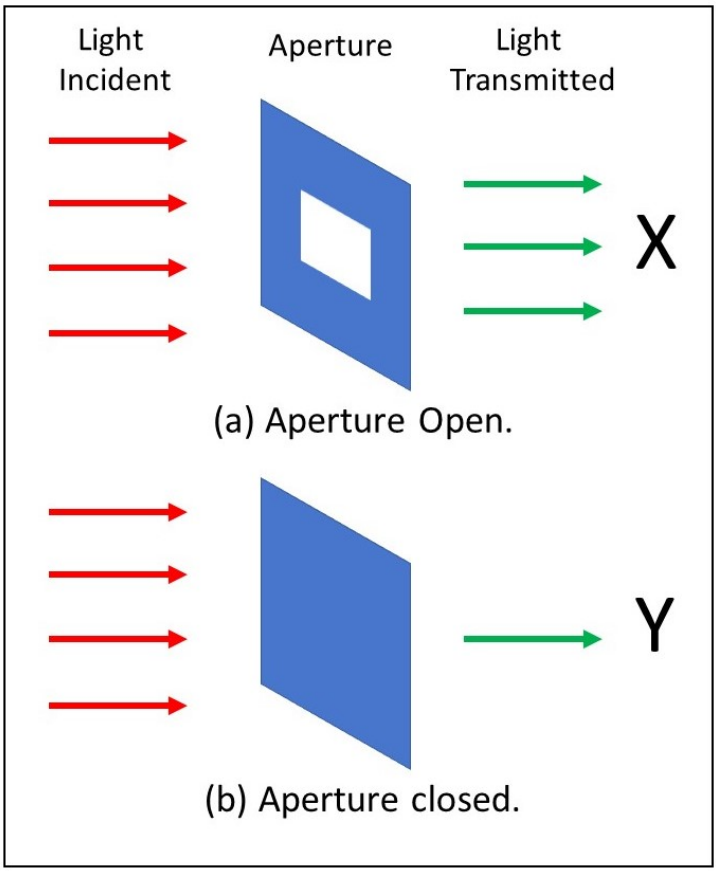


Figure 3.8: Light leak illustration.

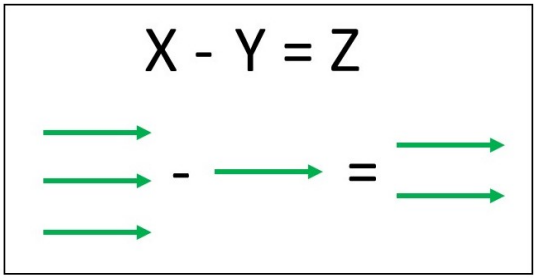


Figure 3.9: Subtraction of the light transmitted in both states (shown in figure 3.8) gives the resultant light.

was subtracted from (a). Figure 3.10 shows the image enhanced by scaling each pixel by a factor of 10 to brighten it. This is the required resultant image that can be used.

If we look carefully at the subtracted enhanced image, we can see that the image appears quite grainy. This is due to the noise that was present in the image capture stage and is explained further in the section below.

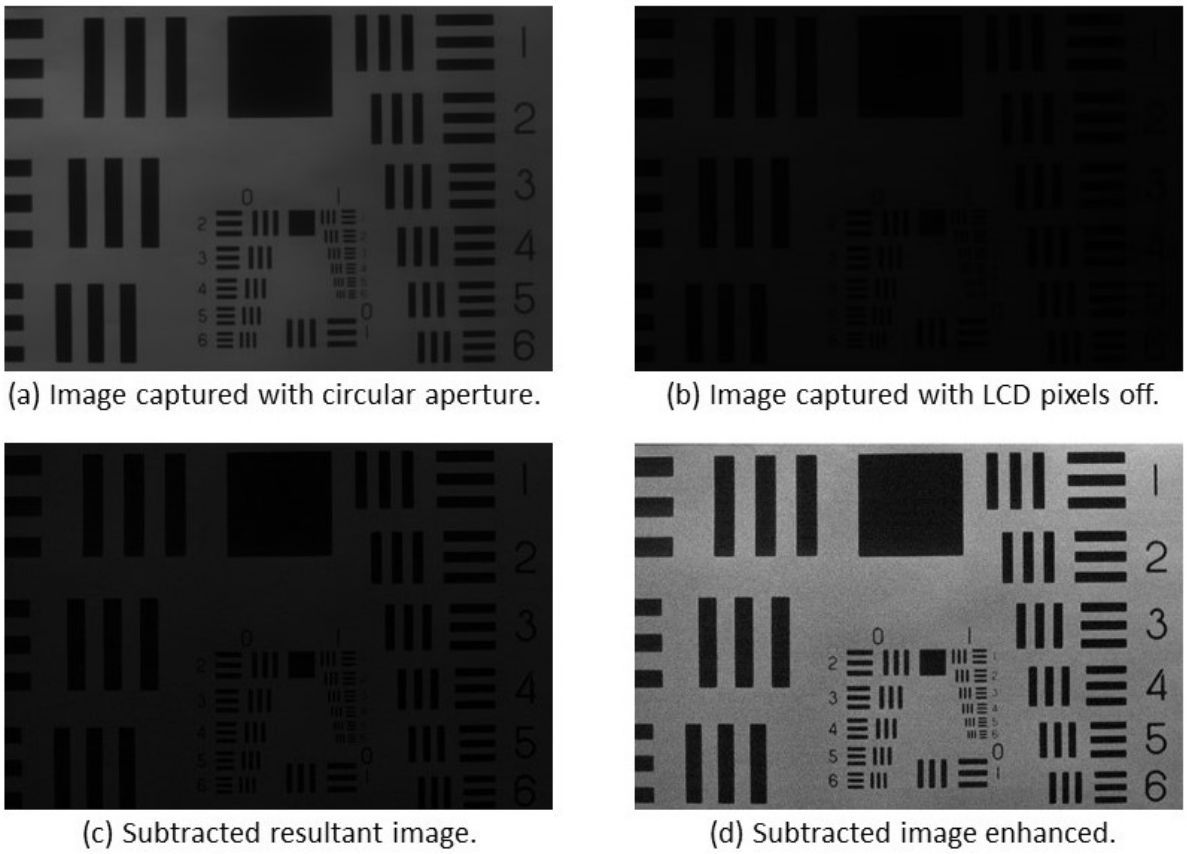
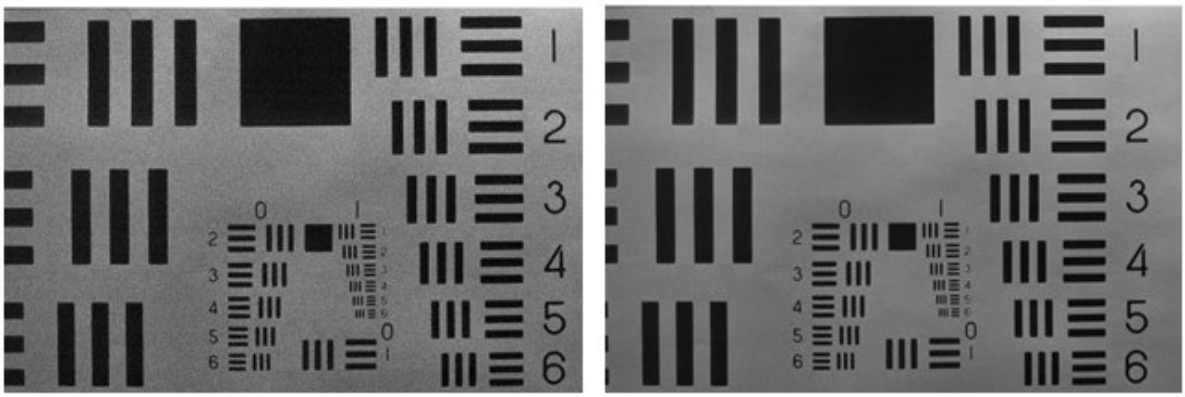


Figure 3.10: Light leak correction process.

3.7.3 Image noise

Image and camera noise are important factors to consider as they will ultimately impact the quality of the images that are produced. Noise is by no means a simple process to model as there are several ways that it can be introduced into the system under observation.

A simple, yet effective method of noise reduction is to take multiple images of the same scene in quick succession and average them together. By keeping all the camera settings the same and making the target completely uniform, any intensity deviations between pixels can be assumed to be purely additive noise. In figure 3.11 (a) the image appears very grainy. This is due to the noise present in the image. Figure 3.11 (b) shows the same image but this time averaged over 10 iterations. The image appears smoother and less grainy than before. To improve the results even further, we can average the image over many more iterations (100+). This will further help to reduce the noise present in the images.



(a) Image before averaging. (b) Image after averaging.

Figure 3.11: Image noise reduction.

3.7.4 Effective F-number

The F number or F# of a camera system is the ratio of the focal length f to the diameter of the aperture D. Mathematically

$$N = \frac{f}{D}, \tag{3.1}$$

where N is the F number. Since the aperture of the lens has been replaced with an LCD, the F# of the primary lens as defined by the manufacturer is no longer effective. The new F# can be calculated using the size of the LCD and the new focal distance. Since the LCD is rectangular, the diameter is calculated using $2\sqrt{uv/\pi}$, where (u, v) is the dimension of the LCD, and was defined in [17] for their rectangular LCoS. Thus equation 3.1 can be modified to

$$N = \frac{2}{f}\sqrt{\frac{uv}{\pi}}. \tag{3.2}$$

Substituting $f = 20mm$ and $u = v = 8mm$ we get an effective F# of F/2.22 which is smaller than the F/1.8 as defined by the manufacturer of the primary lens.

Chapter 4

Programmable aperture fundamentals

There are many uses for a programmable aperture in a camera system. This chapter focuses on coded apertures and how they can be implemented on a programmable mask. The basic theory and mathematical procedures used as building blocks in the rest of this research are detailed here. The applications include defocus deblurring, depth estimation and light field capture. Using a programmable aperture for exposure correction is also discussed. The first section provides a brief overview of the PSF, which is required as it relates to the other sections that follow. Thereafter, in section 2, optimising an aperture to deblur an out-of-focus image online is presented. Section 3 discusses depth estimation using coded apertures on a programmable mask. Thereafter, in section 4, light field capture is explained with applications in stereo vision and refocussing detailed. Finally, exposure correction is presented in section five. All of these applications can be implemented using the programmable aperture photography camera described in chapter 3.

4.1 The point spread function (PSF)

A point spread function (PSF) can be defined as the image taken by a camera system of an almost ideal point light source. The quality of the imaging system can be determined by the quality and spread of the PSF. The following explanation was adopted from [13].

For a typical Lambertian scene without occlusion, the image formation can be formulated

as an integral of the corresponding PSF of every scene point:

$$F(x,y) = \int_{p\varepsilon\Omega} K(x,y|p)I(p)dp, \tag{4.1}$$

where p is any visible three dimensional point on the scene geometry Ω, $I(p)$ is the light intensity at p and $K(x,y|p)$ is the PSF for the point p. The image formation is well known to be a process of dimension reduction, in which a large amount of information is lost. We can see that the PSF $K(x,y|p)$ is the kernel of the mapping from a three dimensional scene to a two dimensional image in this process. As the kernel of image formation, the PSF determines how the scene texture $I(p)$ is distorted and how the scene geometry Ω is encoded in the two dimensional image. The kernel $K(x,y|p)$ is solely determined by the camera design. If we consider an ideal pinhole camera model, then $K(x,y|p)$ is a Dirac delta function. An example of a pin-hole camera can be seen in figure 4.1.

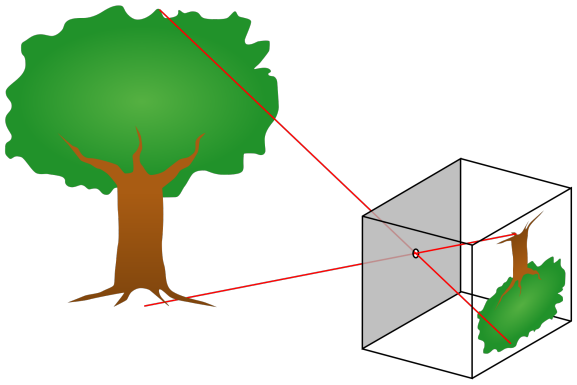


Figure 4.1: An example of a pin-hole camera model imaging a tree [33].

In a typical thin lens model, $K(x,y|p)$ is a disk function whose scale is determined by the relative position p to the focal plane as shown in figure 4.2. Objects at a greater distance from the focal plane will appear increasingly blurry. The scale of the PSF indicates the amount of blur in the image. The smaller the scale of the PSF, the more in focus the image. This is due to the light ray from each point on the object converging to a single point on the camera image sensor. When an object is out of focus we have a single ray from each point on the object converging to multiple points on the image sensor and thus the image appears blurry.

In practice a PSF is usually the combination of multiple optical effects, including but not limited to diffraction, aberration, defocus and veiling glare [13].

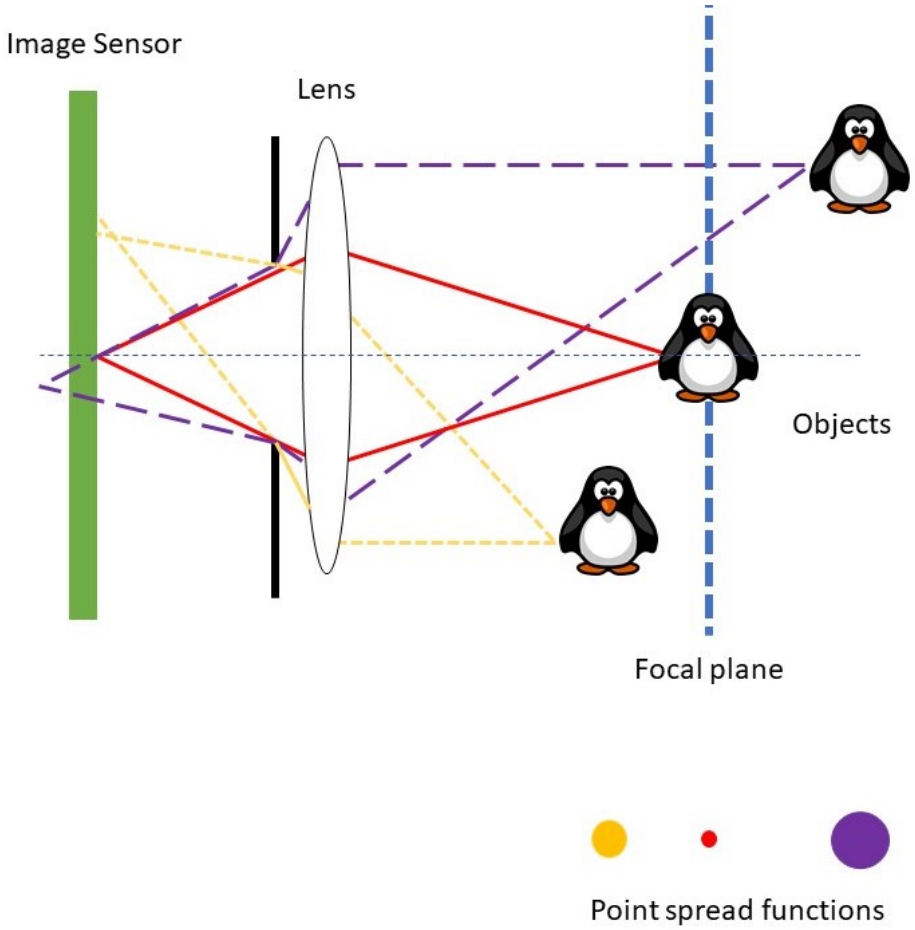


Figure 4.2: Model of a thin lens camera with a finite sized aperture. Objects not at the focal plane will appear blurry in nature. Image adopted from [13].

4.2 Defocus deblurring

Objects appear sharp when they are placed at the focus distance of a lens. Any object outside of this will appear blurry. Trying to bring blurred objects back into focus post-capture can be a challenge. Traditional cameras with circular apertures perform very poorly in trying to deblur an out-of-focus image.This is due to the presence of zero crossings in the frequency domain of the circular aperture which causes the loss of information when convolved with an image. This lost frequency information can often not be recovered or estimated. Coded apertures perform much better at this task. They can be designed to have minimal zero crossings in the frequency domain which leads to a preservation of information. Using a programmable mask to implement coded apertures supports the deblurring process further by making it easier to implement and design such apertures. This section details the formation of defocus blur as well as methods to improve the deblurring process.

4.2.1 Formulating defocus blur

An out-of-focus image can be expressed as simply the convolution between an in-focus sharp image with the PSF of the aperture pattern plus additional noise. In a mathematical form

$$f = f_0 \otimes k + \phi, \tag{4.2}$$

where f is the out-of-focus blur image, f_0 is the in-focus sharp image, k is the PSF of the aperture and ϕ is white Gaussian noise usually modelled by $N(0, \sigma^2)$ (which describes a normal distribution with zero mean and σ^2 variance). We can take the Fourier transform of the above equation to get a frequency domain equivalent of the spatial domain formula. In the frequency domain

$$F = F_0 \cdot K + \Phi, \tag{4.3}$$

where F,F_0, K and Φ are the Fourier transforms of f, f_0, k and ϕ respectively. We can see that the convolution in equation 4.2 becomes multiplication in equation 4.3 as this is a standard property of the Fourier transform. One can also consider the modulation transfer function (MTF), which is simply the power spectrum of K, and is commonly used to determine the quality of imaging systems in optical design [13].

To be able to recover the sharp, in-focus image f_0 from the blurred image f one would need to know the PSF k that caused the blur. A deconvolution, which is the inverse of the convolution process, can be used to solve for the sharp image. Mathematically

$$f_0 = f \otimes^{-1} k, \tag{4.4}$$

and if we consider the Fourier transform deconvolution becomes a division

$$F_0 = \frac{F}{K}. \tag{4.5}$$

The deconvolution process, however, is not simple and there are many ways of solving for the sharp image from the blurred one. In Zhou et al. [13] an evaluation metric is presented and a summarised derivation is adopted here, but their paper should be consulted for a more in-depth explanation.

Given a blur pattern K and a defocused image F the focused image F_0 can be estimated by solving a maximum a posteriori (MAP) problem

$$\hat{F}_0 = \arg\max P(F_0|F, K) = \arg\max P(F|\hat{F}_0, K) \cdot P(\hat{F}_0). \tag{4.6}$$

If we assume a Gaussian model and then taking its logarithmic energy function, the above

MAP problem can be solved as the minimisation of

$$E(\hat{F}_0|F,K) = ||\hat{F}_0 \cdot K - F||^2 + H(\hat{F}_0). \tag{4.7}$$

We can constrain the regularisation term $H(\hat{F}_0)$ to be $||C \cdot \hat{F}_0||^2$, where C is a matrix. We can formulate the popular Wiener deconvolution algorithm by minimising $E(\hat{F}_0|F,K)$. Mathematically,

$$\hat{F}_0 = \frac{F \cdot \hat{K}}{|K|^2 + |C|^2}, \tag{4.8}$$

where $\hat{K}$ is the complex conjugate of K and $|C|^2$ is the signal-to-noise ratio (SNR) $|\frac{\sigma}{F_0}|^2$.

4.2.2 Evaluation of deblurring quality

In order to differentiate between good and bad aperture patterns, an aperture evaluation metric would need to be considered and implemented. A simple yet effective metric would be to evaluate the quality of recovery using the L_2 distance between the original image F_0 and the recovered image $\hat{F}_0$:

$$R = ||F_0 - \hat{F}_0||^2. \tag{4.9}$$

For a given noise level σ and PSF K we can compute the expectation of R as [13]

$$R(K,\sigma) = E||F_0 - \hat{F}_0||^2. \tag{4.10}$$

Substituting in equation 4.8 and using $|C|^2 = |\frac{\sigma}{F_0^2}|^2$

$$R(K,\sigma) = \sum \frac{\sigma^2}{|K|^2 + |\frac{\sigma}{F_0}|^2}. \tag{4.11}$$

The above metric indicates the level to which noise will be amplified in the resultant image. Therefore, the PSF that minimises equation 4.11 will be optimal as the least amount of noise will be amplified. Thus the optimal pattern has the smallest $R(K,\sigma)$ value for a given noise level.

4.2.3 Online aperture pattern optimisation

Being able to choose the correct aperture pattern based on specific scene content is by no means a trivial task. If we consider an $N \times N$ aperture, the possible combination

of apertures that can be derived is $2^{N\times N}$. For an aperture size of 13×13, this gives a possible solution space of size $2^{13\times 13} = 7.5 \times 10^{50}$. Clearly an exhaustive search is not possible to achieve.

Overview

One of the most difficult tasks in coded aperture imaging is being able to select an optimal aperture pattern such that its performance for a certain task surpasses that of a conventional circular aperture. There have been numerous attempts in the literature to optimise the aperture shape based on specific criteria using novel techniques and search algorithms. In Levin et al. [2] a novel aperture shape was developed that was extremely useful for depth estimation. In Zhou et al. [14] and Wilson et al. [15] aperture patterns were developed that could allow for scene content in extremely blurred images to be recovered far better than if they had been blurred by conventional circular apertures.

The problem with the techniques used was that the patterns were developed to be generic, i.e. good for a variety of different scenes and not specific to a single scene. This was required at the time as the aperture patterns were made from static materials and thus could not be changed once their final shapes had been defined. This means that one could not optimise the aperture based on the scene content as one would need a different aperture mask for each scene, which is not practical and feasible.

Since the problem of generating any aperture pattern online has been solved by placing a programmable aperture in the aperture plane, the task of optimising the aperture pattern based on specific scene content can now be achieved. Optimising the aperture pattern based on the specific scene captured will be further discussed below.

Matrix representation

If we consider an $N \times N$ binary aperture, with each element being either a 1 (transparent) or 0 (opaque), then we can represent the aperture as a square matrix of the form

$$k = \begin{bmatrix} k_{11} & k_{12} & k_{13} & \dots & k_{1N} \\ k_{21} & k_{22} & k_{23} & \dots & k_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{N1} & k_{N2} & k_{N3} & \dots & k_{NN} \end{bmatrix}.$$

This is a simple grid that will allow us to visualise which portions of the aperture will allow light to pass through and which will not.

Performance metric

We have already devised a metric in section 4.2.2 to determine the quality of deblurring given a particular aperture pattern. However, we need to rearrange equation 4.11 such that the metric is computationally realisable. By multiplying both numerator and denominator by $|F_0|^2$ and rearranging, we get

$$R(K,\sigma) = \sum \frac{\sigma^2|F_0|^2}{|K|^2|F_0|^2 + |\sigma^2|} = \sum \frac{\sigma^2|F_0|^2}{|K \cdot F_0|^2 + |\sigma|^2}. \tag{4.12}$$

Thus by simply capturing a sharp in-focus image of our scene, we can determine to what extent the aperture will aid in deblurring the image should it move out of focus.

Genetic algorithm search

Now that we have a performance metric that allows us to differentiate between good and bad apertures, we still need a way to test several apertures to determine the most suitable one. Since an exhaustive search is not possible, as the solution space is too large, a more realistic approach that can be performed online is needed.

The method used by Zhou et al. [12] was a genetic algorithm based optimisation solution. The genetic algorithm proved to work well in both cases and an optimal solution was quickly found without having to search the entire solution space. The algorithm is explained below and was adopted from Zhou et al. [12]:

1. Initialise the following variables:
   a. The number of generations (G).
   b. The number of samples in each generation (S).
   c. The size of the sample (L).
   d. A single sample (b).
   e. The number of best performing samples in a generation (M).
   f. User defined probabilities for crossover and mutation of samples($c1$ and $c2$).
2. Initialise: $g = 0$. Randomly generate S binary sequences of length L.
3. Image f_0 of scene is captured in-focus.

4. For $g = 1$ to G

   a. Selection: for each sequence b, the corresponding blur function K is computed as $\sum P_{ij} b_{iN+j}$, where P_{ij} is the matrix defined as $P_{ij}(x, y) = 1$, for $[x, y] = [i, j]$ or $P_{ij}(x, y) = 0$, otherwise. It is then evaluated by using equation 4.12. Only the best M out of S sequences are selected.

   b. Repeat until the population (the number of sequences) increases from M to S. Crossover: duplicate two randomly chosen sequences from the M sequences of step 4a, align them, and exchange each pair of corresponding bits with probability of $c1$, to obtain two new sequences.
   Mutation: for each newly selected generated sequence, flip each bit with probability $c2$.

5. Evaluate all the remaining sequences using equation 4.12 and output the best one.

The above process will be used to determine the optimal aperture pattern for a specific scene. Since the process will ideally be done online, the number of generations will be minimised so that the algorithm reaches an optimal solution in a short time. This will not necessary yield the best aperture but should yield an aperture that is good enough for our application and can be used.

4.2.4 Deblurring

Deblurring is simply the process of recovering the in-focus image from the out-of-focus image that has been captured. Since a convolution of the sharp image with a PSF is what caused the image to be blurry, a simple deconvolution with the same PSF can be used to undo the effect.

There exist several deconvolution algorithms that can be used to recover the sharp image from the blurry one. Each algorithm has its own advantages and disadvantages. Each algorithm also accepts different input values to run. Although many algorithms exist, the Weiner deconvolution algorithm was used to deblur the images. The reason for using the Weiner deconvolution algorithm is due to its speed and ease of implementation. Most software packages already have an optimised version of the algorithm built in and thus coding the algorithm from first principles will likely yield a slower version than those already available. Unlike the Richardson-Lucy algorithm, the Weiner algorithm is not iterative and thus can be used to deconvolve an image quickly. The downfall of the algorithm is the need to know the amount of noise in the image prior to commencing the deconvolution. This value can be estimated and was chosen empirically. The Richardson-

Lucy algorithm does not require this parameter and so can yield a better result. But it is computationally slower and may take several minutes to converge to an optimal solution. Thus a trade-off between performance vs. runtime had to be made. For this reason alone, the Weiner algorithm was chosen.

Equation 4.8 is the mathematical form of the Weiner deconvolution algorithm and requires three parameters to estimate the sharp, in-focus image f_0. The parameters are:

1. The blurred image f.
2. The PSF that caused the blur k.
3. The SNR as a numerical value.

There are standard implementations of the Weiner deconvolution algorithm in Matlab and OpenCV which take in the above parameters and output a resultant image. Thus, for time saving reasons, they are used in this project.

4.3 Depth estimation

Estimating depth of objects in an image allows one to envision three dimensions of an ordinary two dimensional image. Standard digital cameras with circular apertures are unable to perform depth estimation. Coded apertures, however, make the depth estimation process simpler. Having a programmable mask makes it easier to implement such apertures, which further supports the depth estimation process. This section details the theory and mathematical principles behind depth estimation.

4.3.1 Background to estimating depth

Depth estimation, or ranging, is the task of determining the depths of various objects in the scene being imaged. As shown in previous sections, light rays emerging from objects at the focal point of a camera system will converge onto single points on the image sensor. On the other hand, objects not at focus will emit light rays that will spread over multiple points on the image sensor, thus forming what is known as a circle of confusion [15].

The shape of the PSF is determined directly by the shape of the aperture, and the scale or spread of the PSF is determined by the distance of the object from the focal length of

the lens. For an object at the focal length the PSF will resemble a tiny dot, but as the object is moved further away the PSF will start to resemble the shape of the aperture. Thus the object distance from the focal plane is directly proportional to the scale or size of the PSF. Figure 4.3 shows the scaling of the PSF for a circular aperture as the object of interest moves further away from the focal plane. Thus we can say that the more blurred the object appears, the further away it is. The problem of determining depth can then be solved by figuring out the PSF that caused the blur and determining the scale to which it has grown [2].

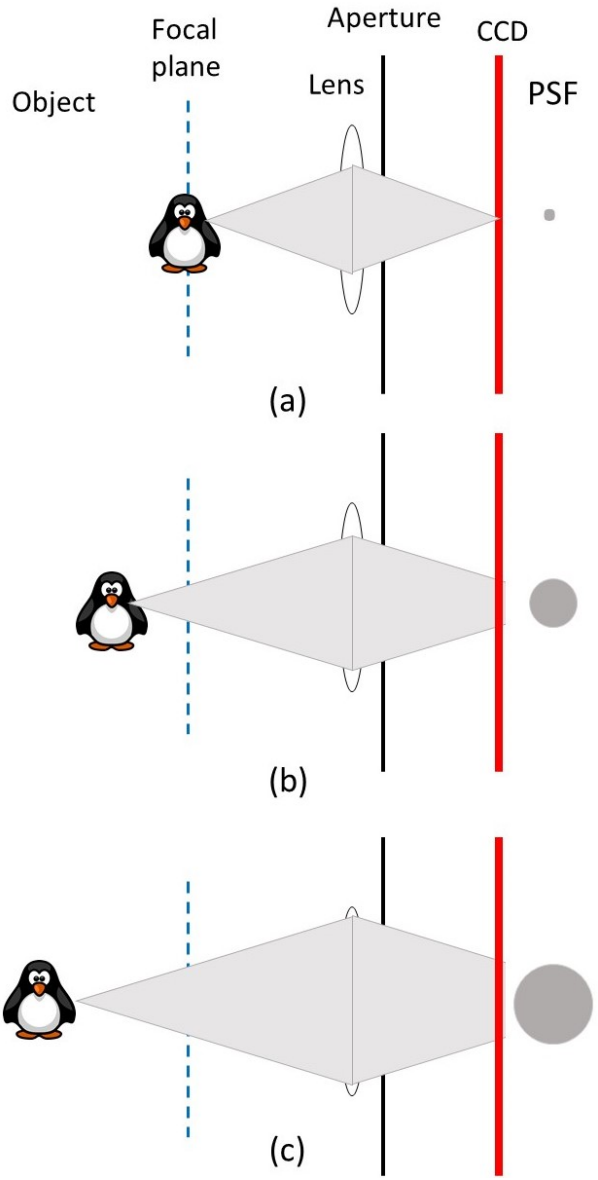


Figure 4.3: PSF scaling for circular aperture as object moves further away.

The most effective method for determining the PSF scale is by analysing the frequency spectrum of the observed blurry image [15]. The zero locations in the frequency spectrum of the PSF, when multiplied by the sharp image, appear in the blurred image and thus do not change positions. A shift in the PSF scale significantly alters the zero locations in

its spectrum and, coupled with the relative ease with which zeros in the observed image can be located, make it a robust method for estimating PSF scale [2].

The problem with using a conventional circular aperture to determine depth is that the zero locations in the frequency spectrum are regularly spaced and thus increase the chances of scale misidentification [2]. This means that several scales have coincidental zero locations so choosing the right scale is problematic. Since coded apertures are more irregular they do not have regular zero spacings in the frequency domain, and thus identifying the correct blur scale becomes easier.

4.3.2 Formulating depth

In order to evaluate the performance of a particular aperture for being able to estimate depth, a performance metric would need to be designed and developed. The following metric was taken from Wilson et al. [15] as it works well and is easily implementable.

We define a sparse derivative distribution prior and assume the distribution is Gaussian with zero mean:

$$P(f_0) \propto \prod_{i,j} \exp(-0.5\alpha((f_0(i,j)-f_0(i+1,j))^2+(f_0(i,j)-f_0(i,j+1))^2) \approx N(0,\Psi) \quad (4.13)$$

and

$$\Psi^{-1} = \alpha((C_{gx})^T C_{gx} + (C_{gy})^T C_{gy}). \quad (4.14)$$

Here (i,j) are pixel indices, α is a scalar set so that the variance of the prior distribution matches that of natural images, N is the normal distribution, g_x and g_y are the horizontal and vertical derivative filters, and C_{gx} and C_{gy} are the convolution matrices for g_x and g_y respectively. The same prior can also be expressed in the frequency domain as

$$P(F_0) \propto e^{-0.5\alpha F_0^T \bar{\Psi}^{-1} F_0}, \quad (4.15)$$

where $\bar{\Psi}^{-1}$ is the Fourier transform of Ψ^{-1} in the form

$$\bar{\Psi}^{-1} = \alpha \cdot \mathrm{diag}(|G_x(v,w)|^2 + |G_y(v,w)|^2), \quad (4.16)$$

where (v,w) are the coordinates in the frequency domain, and G_x and G_y are the Fourier transforms of g_x and g_y respectively. Since the observed image f is produced by convolving the sharp image f_0 and the PSF k, the probability distribution $P_k(f)$ of observed images blurred with PSF k can be calculated as a linear transform of the image prior for sharp

images $P(f_0)$. Therefore $P_k(f)$ will also be a Gaussian distribution, but with its variance transformed by k:

$$P_k(f) \sim N(0, \Sigma_k), \tag{4.17}$$

where the covariance matrix Σ_k is a linearly transformed version of the prior covariance with added noise

$$\Sigma_k = C_k \Psi (C_k)^T + \eta^2 I. \tag{4.18}$$

However, since the Fourier transform of a convolution matrix is always a diagonal matrix, $P_k(f)$ can be simplified to

$$P_k(f) \propto \exp(\frac{-0.5\Sigma_{v,w}|F(v,w)^2|}{\sigma(v,w)}), \tag{4.19}$$

where $\sigma(v,w)$ are the diagonal entries of $\bar{\Sigma_k}$:

$$\sigma(v,w) = \frac{|K(v,w)|^2}{\alpha|G_x(v,w)|^2 + \alpha|G_y(v,w)|^2} + \eta^2. \tag{4.20}$$

The function $P_k(f)$ can now be used to calculate the probability that an observed image f_0 was blurred with a particular scale k. Now that a model determining the probability that an observed image f was blurred with PSF k is defined, a performance metric can be developed. Since two scales k_1 and k_2 of a particular PSF k that are easily distinguishable will produce very different P_k distributions, we can use the distance between the P_k distributions as a performance measure. The method used for measuring the distance between two probability distributions is the Kulback-Leiber (KL) divergence and is defined as

$$D_{KL}(P_{k_1}(f), P_{k_2}(f)) = \int_f P_{k_1}(f)(\log P_{k_1}(f) - \log P_{k_2}(f))df, \tag{4.21}$$

where P_{k_1} and P_{k_2} are the probability distributions for k at scales k_1 and k_2 respectively. By substituting equations 4.19 and 4.20 into equation 4.21, we get the simplified version

$$D_{KL}(P_{k_1}(f), P_{k_2}(f)) = \sum_{v,w} \left(\frac{\sigma_{k_1}(v,w)}{\sigma_{k_2}(v,w)} - \log \frac{\sigma_{k_1}(v,w)}{\sigma_{k_2}(v,w)} \right). \tag{4.22}$$

4.3.3 Aperture pattern optimisation

In section 4.2.3, a genetic algorithm is used to generate an aperture optimised for deblurring a particular out-of-focus scene. We can make use of the same genetic algorithm to solve for a pattern optimised for estimating depth. The fitness function used to find the

optimal deblurring pattern can be replaced with equation 4.22. The new fitness function is independent of scene content but still affected by the noise level present. Since noise varies from one imaging condition to another, the aperture can be optimised for a particular scene based on noise.

4.3.4 Estimating depth

Equation 4.22 is maximised for a PSF that has frequencies (v, w) where the ratio between $K_1(v, w)$ and $K_2(v, w)$ is large. In order to construct a principled performance criterion, the KL divergence in equation 4.22 is calculated between various scales of a particular aperture shape and the minimum divergence between any two scales is recorded [2]. Thus all the possible PSF scales should be captured and tested to determine the correct scale which maximises the function.

Once all the scales have been tested, the pattern scale that produces the highest probability is said to be the correct scale for that depth. Since we know the calibration of the scales we can infer the correct depth for the object in the scene.

4.4 Light field capture

As described in section two, four dimensional light fields contain a lot of useful visual information. This can be used to generate all sorts of novel and improved images of a scene computationally. Capturing a light field involves the use of several different shaped apertures. Each aperture allows the camera to capture an image of the same scene from a slightly different perspective. These images can then be aggregated later on. The problem with using static apertures is that the process of removing and swapping each aperture is time consuming. The process can become tedious if a large number of images are captured. This further becomes problematic if the scene changes during the interchange of apertures as this will render the previously captured images useless. Having a programmable aperture allows for the quick and easy interchange of patterns. The process of capturing light fields can be done automatically in software. This section details the theory and mathematics behind capturing a light field using a programmable aperture mask.

4.4.1 Background to light field capture

Capturing the four dimensional light fields can simply be envisioned, in terms of geometric optics, as the intersection of a light ray with two planes, (u, v) and (s, t). The plane (u, v) is found at the aperture stop of the imaging system and the (s, t) plane is found at the image plane of the imaging system. Figure 4.4 shows the mapping of light rays from a region on the uv plane to every point on the st plane. This means that by opening up small portions on the uv plane, a slightly different image of the same scene can be captured at full resolution on the st plane. If multiple photographs are taken, each with a different aperture region open at a time, then the full light field can be reconstructed from the multiple images [34].

The uv plane can be divided into any number of regions but in this project we will restrict the division into an $N \times N$ binary aperture. While having a large N value increases the angular resolution to which images are captured, it also decreases the amount of light reaching the sensor. This will cause an increase in the exposure time and thus an increase in the noise that enters the system. Thus a trade-off will have to be made between angular resolution and image quality.

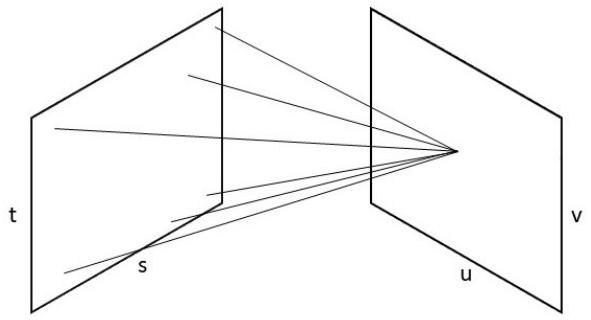


Figure 4.4: Mapping of light rays from a region to on the uv plane to every point on the st plane [35].

4.4.2 Creating virtual photographs

As stated in the previous section, it is possible to capture multiple images of the same scene from slightly different perspectives. This is due to the mapping of individual regions on the uv plane to st plane. It is also possible to combine multiple regions of the uv plane, post capture, to generate a whole new image or range of images. This allows for the adjustment of the aperture shape or the position of the focal plane without having to take any additional photographs. Any aperture shape can be synthesised by integrating the light field over the appropriate (u, v) subset [34].

Figure 4.5 shows how the light field is captured. Each portion of the aperture is opened, one at a time, and an image captured. The full resolution of the camera sensor is utilised each time. This enables multiple photographs to be captured of the same scene but each offering a slightly different view each time.

Figure 4.6 shows the aggregation of several light fields captured to improve the image quality. It is possible to aggregate several different combinations of captured light fields to create virtual images. To mathematically define the aggregation of light fields we can consider a light ray emitting from an object surface point and denote its radiance by the light field [34]

$$l_0([x\ y]^T), \tag{4.23}$$

where x and y are the intersections of the light ray with the two coordinate planes. The light ray travels a distance Z between the object and the lens and a distance F between the lens and the image sensor. During the transition, the light ray will be sheared a number of times [34]. This shearing can be reparametrised into a single linear transformation

$$l([x\ y]^T) = l_0(M[x\ y]^T) = l_0\left(\begin{bmatrix} -\frac{Z}{F} & Z\Delta \\ \frac{1}{F} & \frac{1}{f} - \frac{1}{F} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}\right), \tag{4.24}$$

where f is the focal length of the lens and $\Delta = 1/Z + 1/F - 1/f$. If we modify the shape of the aperture so that only the light rays arriving in a small specified region y are allowed to pass through, then we can denote the resulting image a subset of the lightfield as

$$I_y(x) = l([x\ y]^T), \tag{4.25}$$

where I_y is the light field image. A light field with angular resolution N requires N exposures, one for each y. At each exposure the captured image M_y is a linear combination of N light field images,

$$M_y(x) = \sum_{k=0}^{N-1} w_{yk} I_k(x), \tag{4.26}$$

where the weights $w_{yk} \in [0, 1]$ can be represented by the vector $w_y = [w_{y0}\ \ w_{y0}\ \ ...\ \ w_{y(N-1)}]$.

4.4.3 Stereo vision

The process of obtaining depth using stereo vision techniques was first discovered in 1833 by Charles Wheatstone [36]. In his discovery, he observed that the human eyes are separated by some horizontal distance and this allowed each eye to view the same scene from a slightly different vantage point. This makes the location of objects in view differ

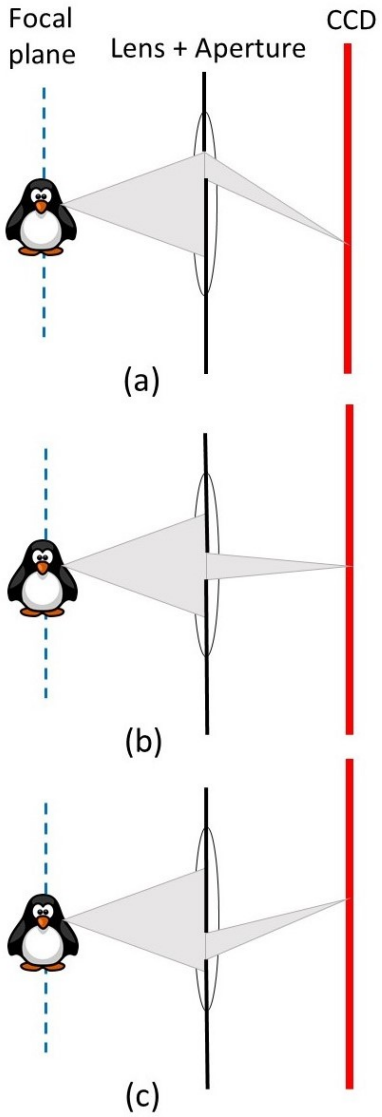


Figure 4.5: Capturing the light field. Each portion of the aperture is opened, one at a time, to capture a single sample.

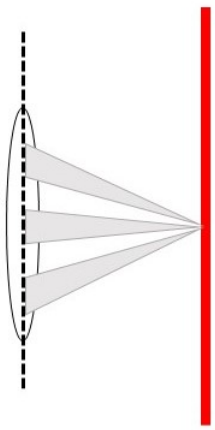


Figure 4.6: Aggregating the samples captured of the light field in figure 4.5 to create a new virtual image.

slightly from eye to eye. The brain is then able to infer depth from the information it receives from both eyes.

The same principles can be taken and applied in image processing to infer depth from images of a scene. The technique uses two stationary cameras placed on the same horizontal plane but at a slight distance apart. Figure 4.7 shows the optical centres of the two cameras, C_l and C_r, where l and r represent the left and right cameras respectively. The epipolar geometry relates the point p and the projections onto the left and right image planes, p_l and p_r respectively, which is then imaged by the respective cameras[37]. These cameras are able to image the same scene from slightly different viewpoints. The images are then input into any one of many stereo vision algorithms from which a disparity map can be generated. A disparity map is simply an image in which the intensity of each pixel is inversely proportional to the object's distance in the scene.

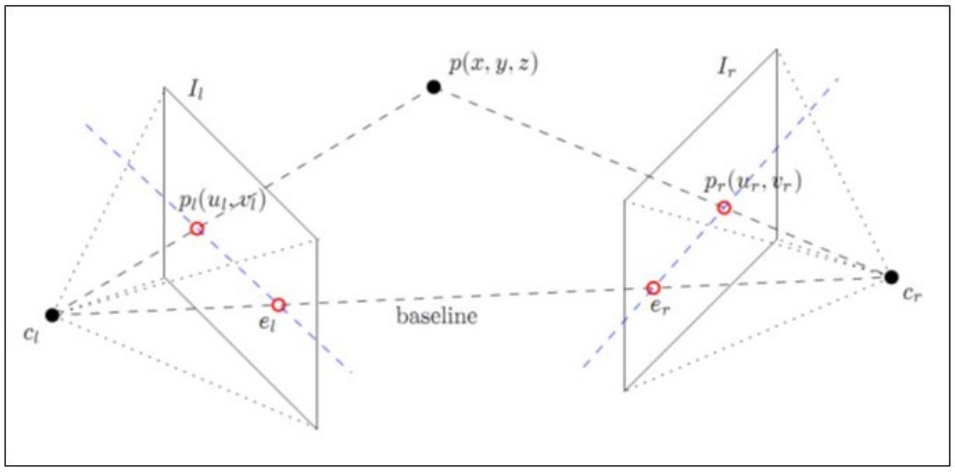


Figure 4.7: Epipolar geometry in a stereo setup [36].

Coded apertures can be used to extract depth from a scene by combining specific apertures with stereo vision techniques. A slightly different view of the same scene is captured by a single camera which utilises two different aperture patterns with horizontally-varying openings. Figure 4.8 illustrates this concept by showing the same scene captured with two different aperture shapes. We see a slight perspective shift and we can use this to estimate the depth of the objects in the image.

4.5 Exposure correction

Image quality can be affected by various conditions, but one of the factors that can lead to unpleasant images is the improper exposure to light. It is possible to adjust the

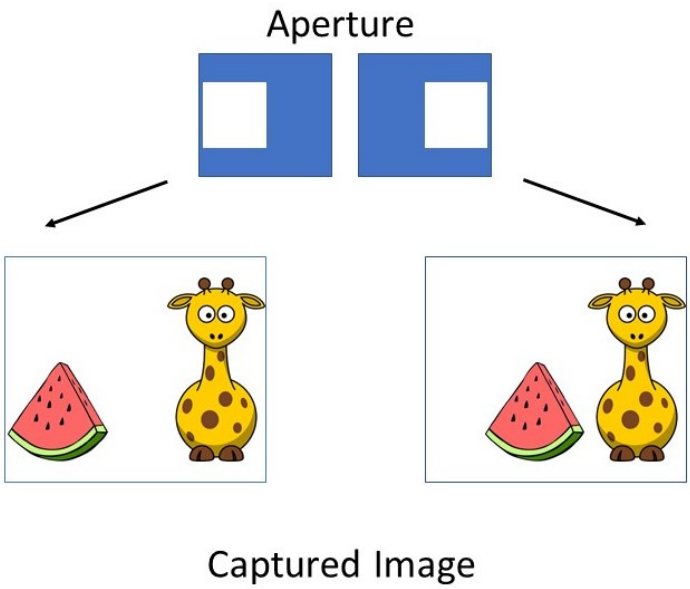


Figure 4.8: Two different apertures capturing the same scene from slightly different perspectives.

exposure duration (T) of the camera, but having a longer exposure usually increases the amount of noise that enters the system. Another option is to change the aperture size (F) to adjust the light. Having a programmable aperture allows for the easy adjustment of the aperture size. Since the aperture is modified in software it can be automatically updated in real-time to allow for the correct exposure. By sacrificing DOF, it is possible to correctly expose an image without the added noise.

Coded aperture masks allow for exposure variation as they alter the way that light reaches the camera sensor. The problem with using coded apertures to adjust exposure is that they optically encode the image. This means that a further computational step is needed to decode the image. This is time wasteful and often degrades the image quality. Therefore, this section focuses only on using a circular aperture to adjust the exposure as no further computational steps are needed after image capture.

4.5.1 Background to exposure correction

Many sophisticated cameras today make use of advanced algorithms to try and determine the optimum exposure value (EV) for particular scenes. The EV specifies the relationship between the aperture size (F), and the exposure duration (T). Both of these factors ultimately determine the amount of light that will reach the camera sensor. There is no exact rule as to what the correct exposure should be. It is possible to abstract a generalisation and to define the best exposure that enables one to reproduce the most important regions

with grey or brightness, more or less in the middle of the possible range [38]. Figure 4.9 shows three images of a scene with different EVs. The underexposed image appears too dark while the overexposed image is too bright and has become saturated. The correctly exposed image is in the middle with most to all objects in the scene being visible.

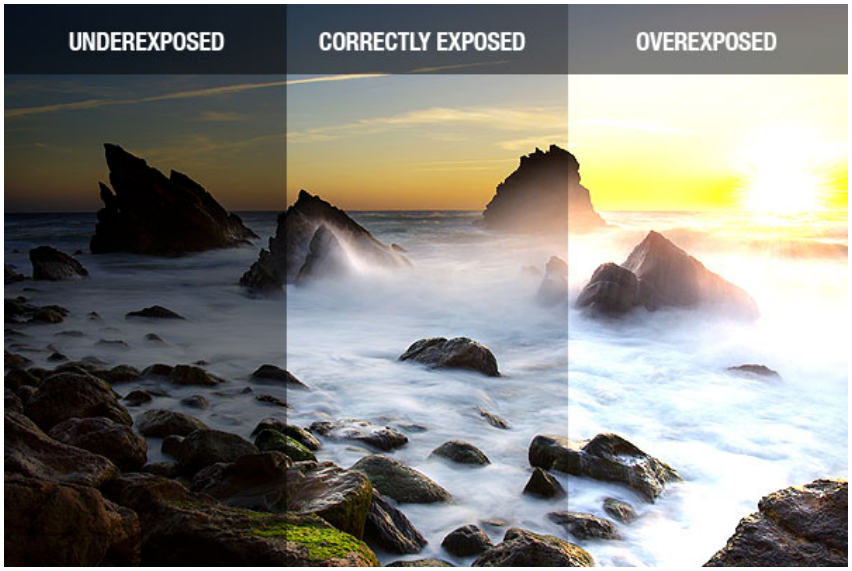


Figure 4.9: An example of an underexposed scene, correctly exposed scene and overexposed scene [39].

Most cameras that contain electronically-controlled shutters and apertures would have preprogrammed settings that a user can select from which will determine the correct aperture size and shutter speed for that particular scene. The mechanical diaphragm that determines the aperture size is often not a reliable tool as it consists of moving mechanical parts that can break easily. This means that the diaphragm itself has a limited lifespan and would often need to be replaced. Since we have removed the physical diaphragm from the system and replaced it with an LCD, we don't have to worry about the lifespan of the LCD as it has no mechanically moving parts. We can now utilise the LCD and the shutter to correctly and efficiently determine the appropriate EV for each and every scene.

4.5.2 Exposure correction algorithm

To start, we can mathematically define the EV as

$$EV = log_2(\frac{F^2}{T}) = 2log_2(F) - log_2(T). \tag{4.27}$$

From equation 4.27, the EV is directly proportional to the F number and inversely proportional to the shutter speed. There are many ways in which the optimal EV can be chosen but most algorithms make use of the following [38]:

1. Take a picture with a predetermined exposure value (EV_{pre}).
2. Convert the RGB values to brightness B.
3. Derive a single value B_{pre} (centre-weighted mean, median etc.) from the brightness picture.
4. Based on linearity assumption and equation 4.27, the optimum exposure value EV_{opt} should be the one that permits a correct exposure. The picture taken at this EV_{opt} should give a number close to a predefined ideal value B_{opt}. Thus:

$$EV_{opt} = EV_{pre} + log_2(B_{pre}) - log_2(B_{opt}). \tag{4.28}$$

The ideal value B_{opt} for each algorithm is typically selected empirically.

In real world images, very seldom does the entire image contain important information that one needs to take into consideration when determining the correct exposure settings. It is possible to use a subset of the image to help determine the correct exposure to reproduce the most important regions with the best level of brightness. There are several methods to determine the most important regions in an image but the one used here was adopted from Battiato et al. [38], which extracts features based on contrast and focus. Focus characterises the sharpness of the image and is useful for determining regions where high frequency components are present. Contrast, on the other hand, refers to the range of tones present in the image.

To determine the amount of contrast and focus in an image two different algorithms are used, one for contrast and one for focus. A summary of the algorithms are presented below. For the contrast measure:

1. An image is captured with an arbitrary aperture F number F and shutter speed T.
2. The image is then divided into N blocks of equal dimensions.
3. For each block, a histogram is built in the range 0 to 255. In order to remove irrelevant peaks, the histogram is slightly smoothed by replacing each entry with its mean. Thus the original histogram entry is replaced with the grey level

$$\tilde{I}[i] = \frac{I[i-2] + I[i-1] + I[i] + I[i+1] + I[i+2]}{5}. \tag{4.29}$$

4. The mean value M is then calculated for each block

$$M = \frac{\sum_{i=0}^{255} i \cdot \tilde{I}[i]}{\sum_{i=0}^{255} \tilde{I}[i]}. \tag{4.30}$$

5. The deviation D is computed as

$$D = \frac{\sum_{i=0}^{255} |i - M| \cdot \tilde{I}[i]}{\sum_{i=0}^{255} \tilde{I}[i]}. \tag{4.31}$$

6. Blocks that have a D value that is above a certain threshold are considered to have a high contrast and thus contain significant features. These are selected.

For the focus measure:

1. An image is captured with an arbitrary aperture F number F and shutter speed T.
2. The image is divided into N blocks of equal dimensions.
3. Each block is convolved with a simple 3×3 Laplacian filter to determine edges in the image.
4. To disregard irrelevant high frequency pixels (noise), the outputs of the convolution at each pixel are thresholded. The mean focus of each block is computed as

$$F = \frac{\sum_{i=1}^{P} thresh[lapl(i), \sigma]}{P}, \tag{4.32}$$

where P is the number of pixels, σ is the noise level and the $thresh()$ operator discards values lower than a fixed threshold.

Once the values F and D are computed for all blocks, relevant regions will be classified using a linear combination of both values. Figure 4.10 illustrates a simplified version of the above-stated process for feature extraction. After the relevant regions are identified they can be used to determine the optimum EV by utilising equation 4.27. Figure 4.11 shows a simplified summary of the exposure correction algorithm. The algorithm performs an iterative loop to find the correct aperture shape such that the exposure in the image is correctly adjusted and selected.

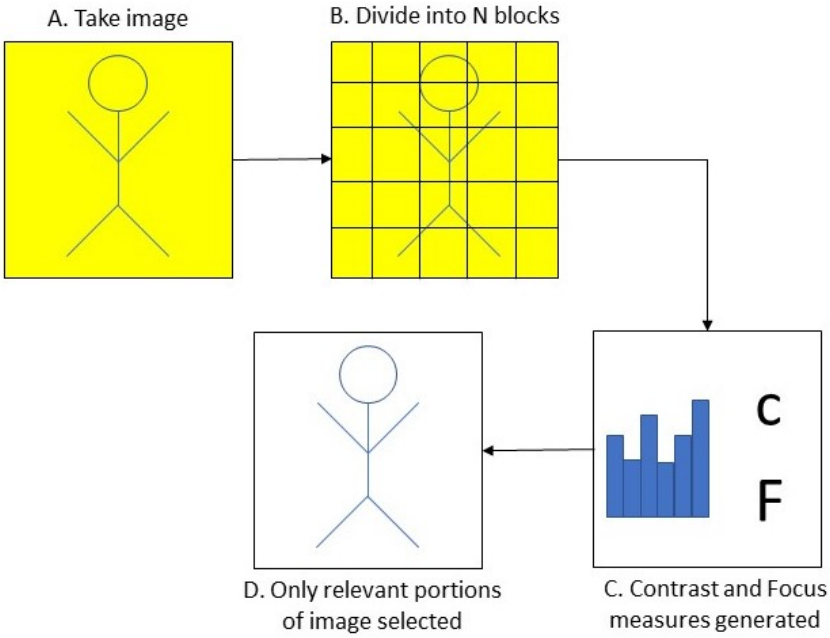


Figure 4.10: Feature extraction using contrast and focus measure.

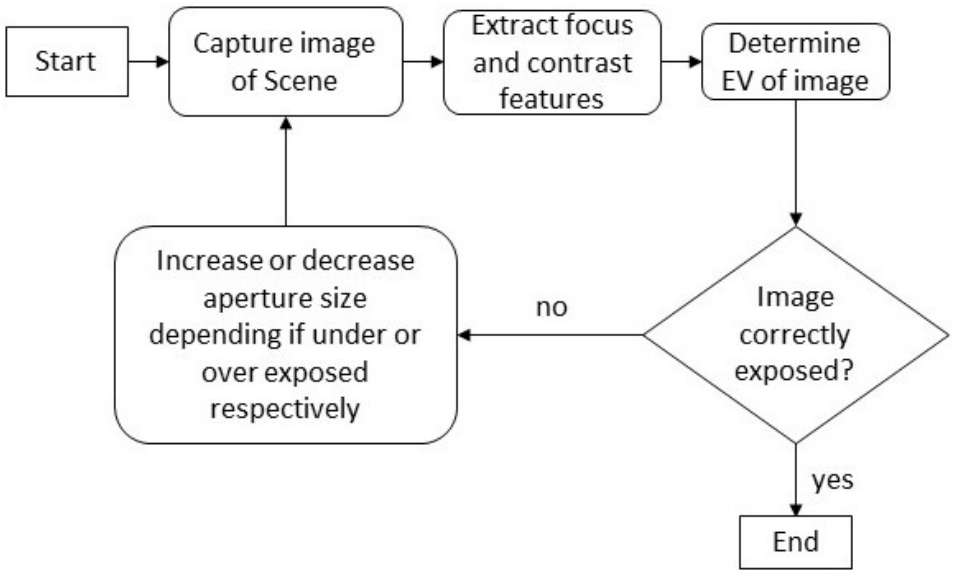


Figure 4.11: Flow chart of algorithm for exposure correction.

Chapter 5

Methodology and results

This chapter details the uses for a programmable aperture camera. The experimental procedures carried out and the results obtained are presented. This work was performed using the theory and mathematical principles discussed previously in chapter 4. The programmable aperture camera, mentioned in chapter 3, was used in all of the experiments that follow.

Experiment 1 describes the capture of different PSFs using various aperture masks. These are needed to perform deblurring and depth estimation in the experiments that follow. Experiment 2 details the optimisation of the aperture pattern online for deblurring an out-of-focus image. The aperture was updated based on scene content which can only be achieved using the reconfigurable mask. Experiment 3 estimates depth in an image using an optimised pattern. It is then compared to existing apertures. Experiment 4 describes the capture of a light field using several different aperture shapes. This was used to perform virtual refocussing and stereo vision. Finally, experiment 5 details the ability of the programmable aperture to adjust to various levels of light and provide the correct exposure for a given scene.

5.1 Experiment 1 : PSF capture

This section details the experimental procedures carried out for capturing PSFs which were needed in later experimental work.

5.1.1 Aim

The aim of this experiment is to determine how well the system is able to capture a PSF at different distances away from the plane of focus. Since a PSF is required to perform tasks such as deblurring and depth estimation, it is important to establish the quality of the PSF captured. This directly affects the deblurring and depth estimation processes. The shape of the PSF will differ depending on the shape of the aperture. To ensure the PSF resembles the aperture shape, several different patterns are tested.

5.1.2 Apparatus

For this experiment, the following equipment was needed:

1. A point light source in the form of a light emitting diode (LED).
2. A movable tray table.
3. A vertical stand.
4. A tape measure.
5. Black insulation tape.
6. Camera tripod.
7. Computing platform (described in section 3.5).
8. Three different aperture pattern images (stored on the computing platform).

The patterns selected are shown in figure 5.1. Figure 5.1 (a) is optimised for deblurring a blurred image [12]. Figure 5.1 (b) is optimised for extracting depth from an image [2]. Figure 5.1 (c) is an ordinary circular aperture. Since each LCD pixel is represented by a square block, the apertures had to be constructed on a block by block basis. Because of the abstract shapes of the different aperture patterns, constructing all the shapes using the same number of blocks was not realisable. They were, however, kept as close as possible to ensure that the same amount of light was let through.

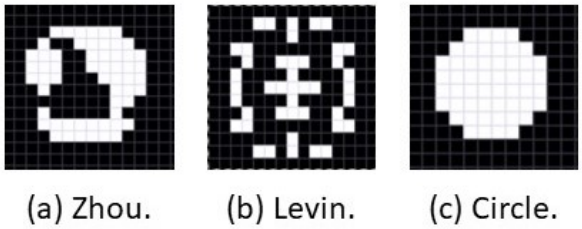


Figure 5.1: Three aperture patterns selected.

5.1.3 Method

The point light source was mounted on a vertical stand and placed on a movable tray table. The camera was mounted on a tripod and placed in one corner of the room. It was connected to the computing platform which allowed for remote capture of images and generation of different aperture patterns on the LCD.

Next, the floor was measured using the tape measure and marked in increments of 0.1m starting from the focus distance of the camera (0.5m away) up until 1.5m away in a straight line. The point light source was moved to the various distances marked on the floor. At each distance, images of the point light source were captured using the different apertures. These images were then stored. Figure 5.2 illustrates the procedure.

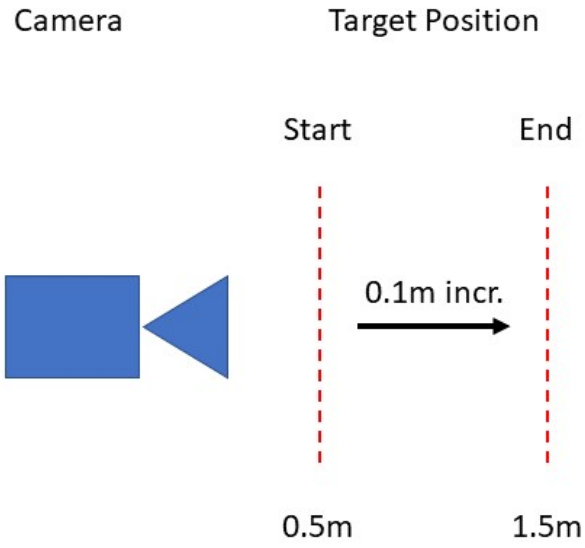


Figure 5.2: Experimental setup with camera in fixed position and target moved from 0.5m to 1.5m away in increments of 0.1m.

5.1.4 Results

Figure 5.3 shows the PSFs captured at various distances with different aperture patterns.

5.1.5 Analysis

From the results obtained, it is clear that the shape of the PSF is directly related to the shape of the aperture. The further away the point light source is placed, the larger the scale of the PSF and the more the shape starts to resemble the aperture.

5.1.6 Conclusions

The system was effective in both generating aperture patterns and capturing the required PSFs. These were used in the experiments that follow.

5.2 Experiment 2 : Defocus deblurring

This section details the experimental procedures carried out for deblurring an out-of-focus image.

5.2.1 Aim

The aim of this experiment was to determine how well the system could deblur an out-of-focus image. It is known that coded apertures are far superior to conventional circular apertures for defocus deblurring. The ability of the programmable aperture camera to implement such patterns was investigated. This experiment first optimised the aperture based on the scene. Thereafter, the optimised pattern was compared to existing aperture patterns that are known to be good for defocus deblurring. Being able to optimise the aperture pattern online is a major benefit of the programmable aperture.

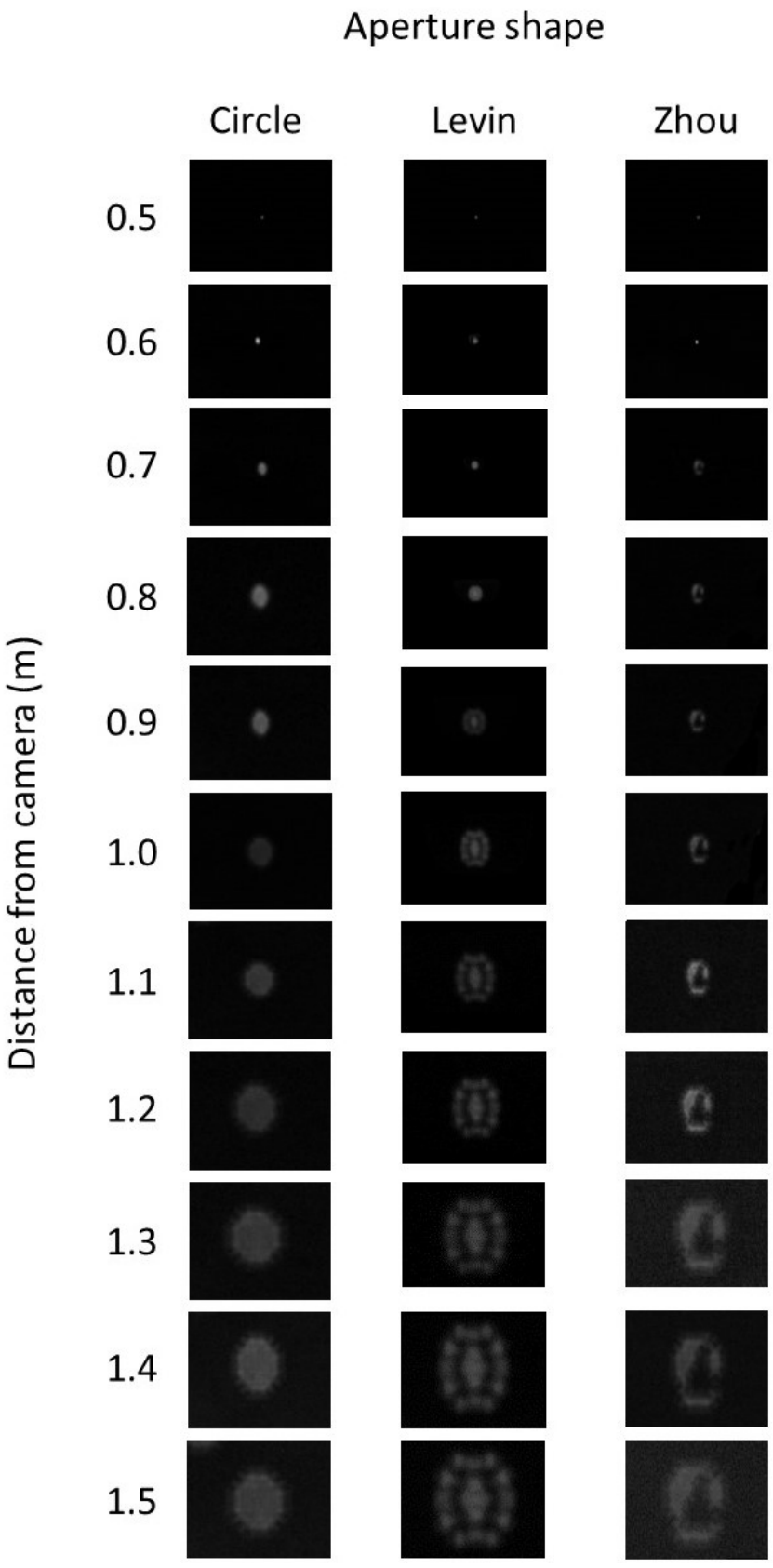


Figure 5.3: PSFs captured at different distances for various aperture shapes.

5.2.2 Apparatus

For this experiment, the following equipment was needed:

1. A target scene.
2. A movable tray table.
3. A vertical stand.
4. Camera tripod.
5. Computing platform (described in section 3.5).
6. PSFs captured in section 5.1.
7. Different aperture pattern images (stored on the computing platform).
8. Deblurring algorithms discussed in section 4.2.4.

The target scene chosen was the standard USAF 1951 resolution test chart. This is shown in figure 5.4. The chart consists of various numbers, small and large, as well as vertical and horizontal lines of varying size and spacing. An explanation into reading the chart is provided in Appendix A.3.

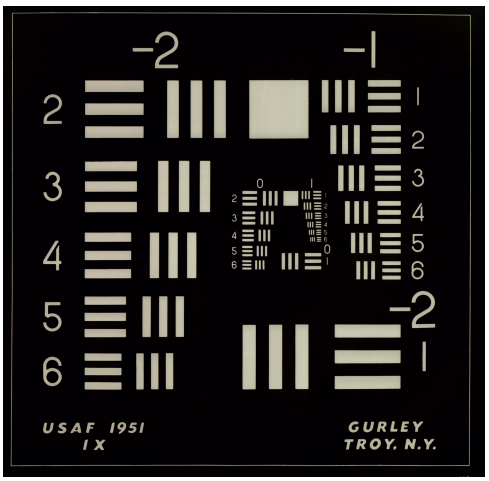


Figure 5.4: Standard USAF 1951 resolution chart.

5.2.3 Method

The experimental description for preforming deblurring is described below. First the aperture was optimised online. It was then compared to other apertures to determine its deblurring qualities.

Online aperture optimisation

The task of determining the optimal aperture pattern for deblurring involves first capturing a sharp in-focus image of the scene. The target was placed at focus (0.5m away) and an image captured. Thereafter, the image was input into the optimisation algorithm described in section 4.2.3. Considering the optimisation algorithm is a genetic algorithm, the longer the algorithm is run the better the chances of determining an optimal pattern. Finding the balance between run-time and performance was done empirically. Table 5.1 shows the parameters used in the algorithm.

Table 5.1: Genetic algorithm parameters.

G	60
S	60
M	12
$c1$	0.2
$c2$	0.05
$sigma$	0.03

Once the algorithm produced a suitable aperture pattern, it was then tested for its deblurring properties. This is described in the next section.

Comparison of deblurring apertures

After the aperture was optimised for the scene, as described in the previous section, it was then used to perform deblurring. The test chart was mounted on a vertical stand and placed on the movable tray. The camera was mounted on a tripod and placed in one corner of the room. It was connected to the computing platform which allowed for remote capture of images and generation of different aperture patterns on the LCD.

Since the floor was already marked during the previous experiment, the tray table with the target scene was moved to 0.5m, 1.0m and 1.5m distances away from the camera. Figure 5.5 illustrates the target positions. Images were captured using the optimized aperture and those mentioned previously in figure 5.1. These images were given to the deblurring algorithms (mentioned in section 4.2.4) to perform the image sharpening operations.

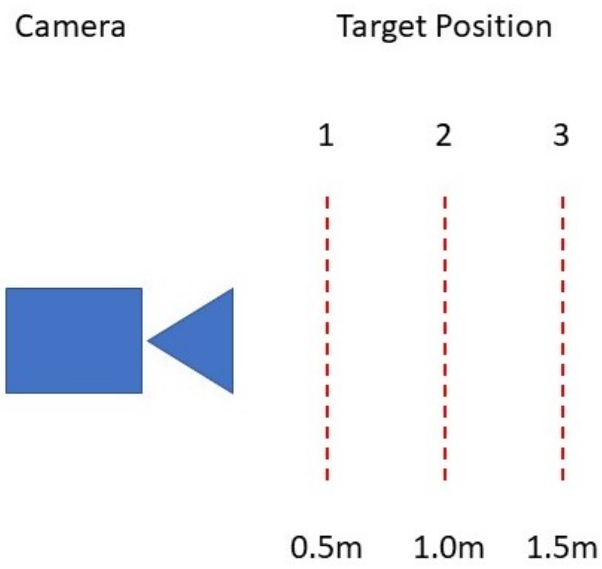


Figure 5.5: Experimental setup with camera in fixed position and target moved to three positions away from lens: 0.5m (focus), 1.0m and 1.5m.

5.2.4 Results

The results of the aperture optimisation and deblurring experiments are presented below.

Online aperture optimisation

The algorithm was run five times and the average time of execution was recorded as shown in table 5.2. After five executions, the pattern shown in figure 5.6 was the best performing with the lowest $R(k, \sigma)$ value.

Table 5.2: Genetic algorithm execution times (seconds).

(a)	279
(b)	288
(c)	274
(d)	286
(e)	285
Average	282,4

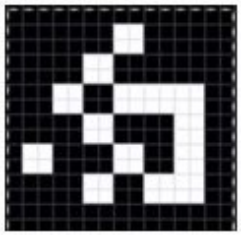


Figure 5.6: Optimised aperture pattern.

Comparison of deblurring apertures

Figure 5.7 presents the deblurring results at 0.5m away from the camera (focus). Figure 5.8 shows deblurring performed at 1.0m away. Deblurring at 1.5m away is shown in figure 5.9. A close up of the deblurring performed at 1.5m is shown in figure 5.10.

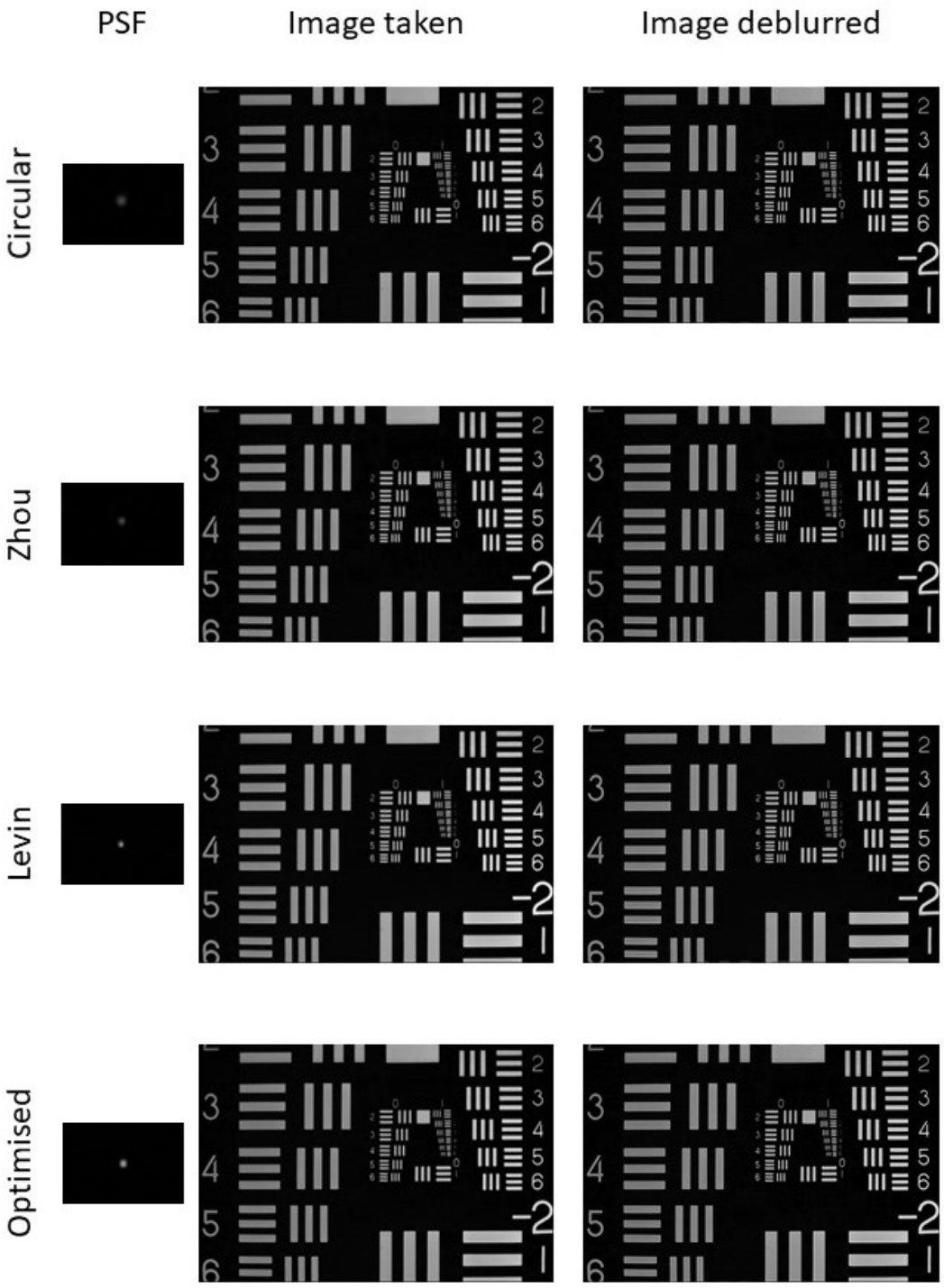


Figure 5.7: Defocus deblurring at 0.5m (focus).

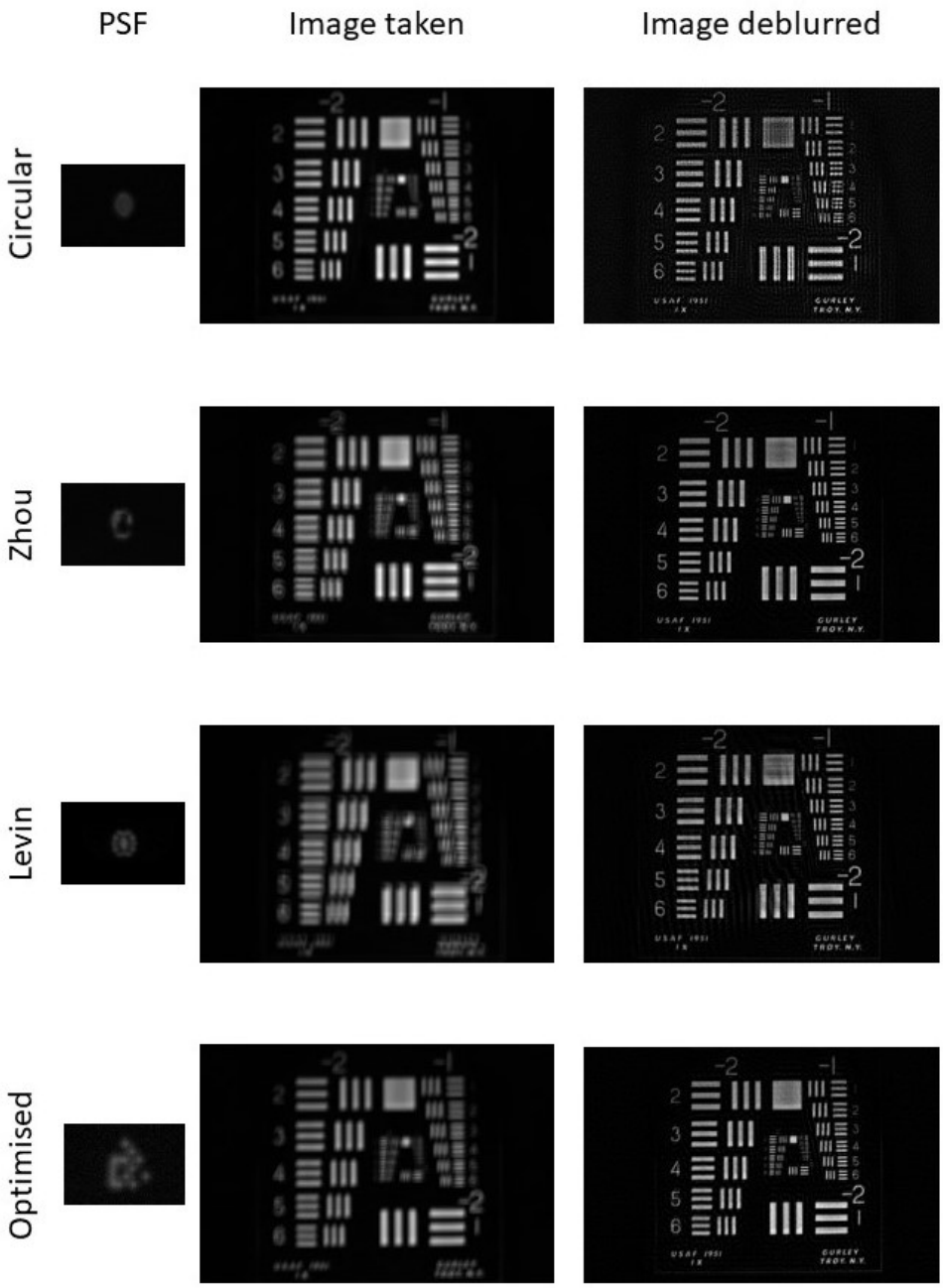


Figure 5.8: Defocus deblurring at 1.0m.

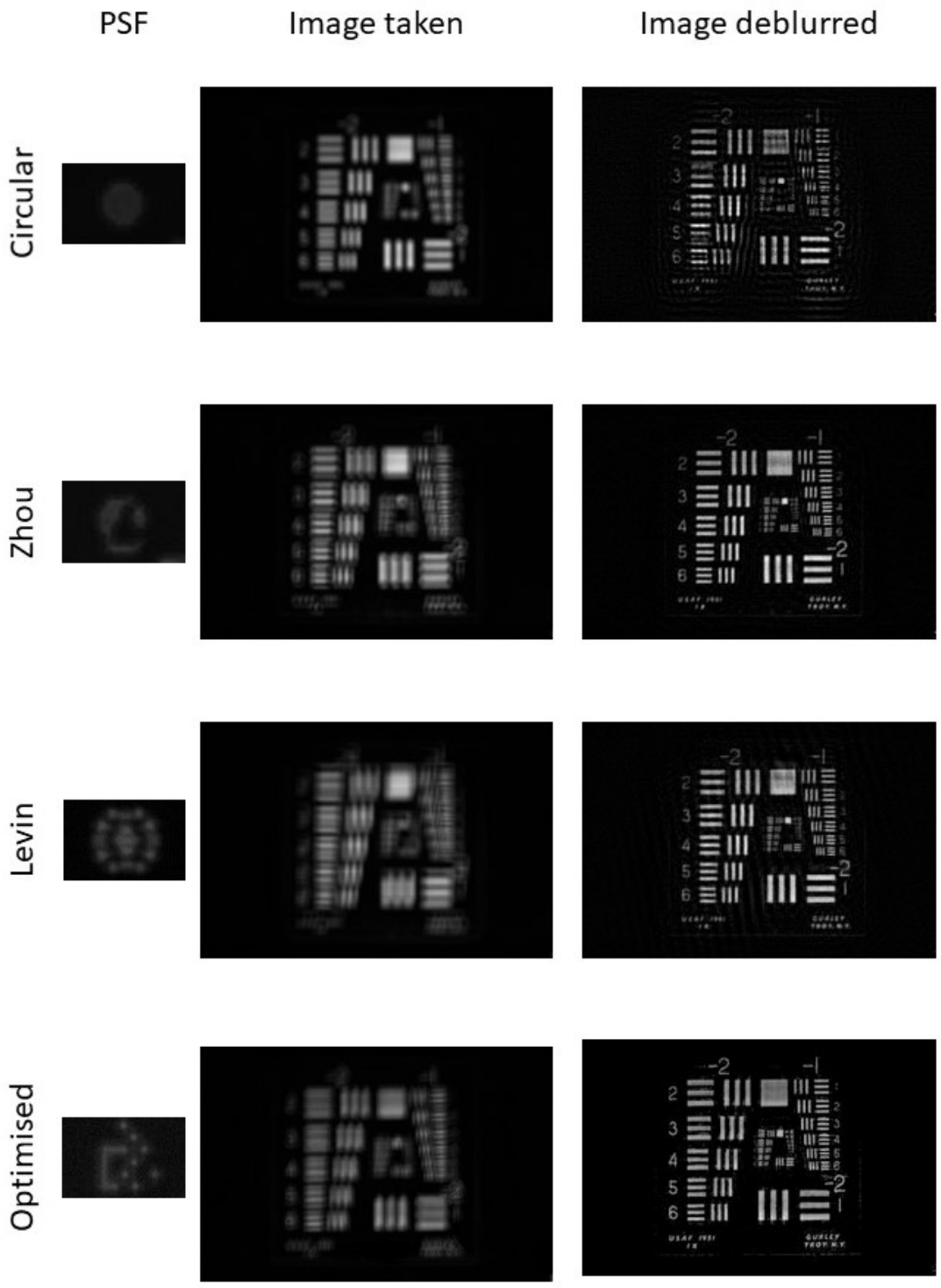


Figure 5.9: Defocus deblurring at 1.5m.

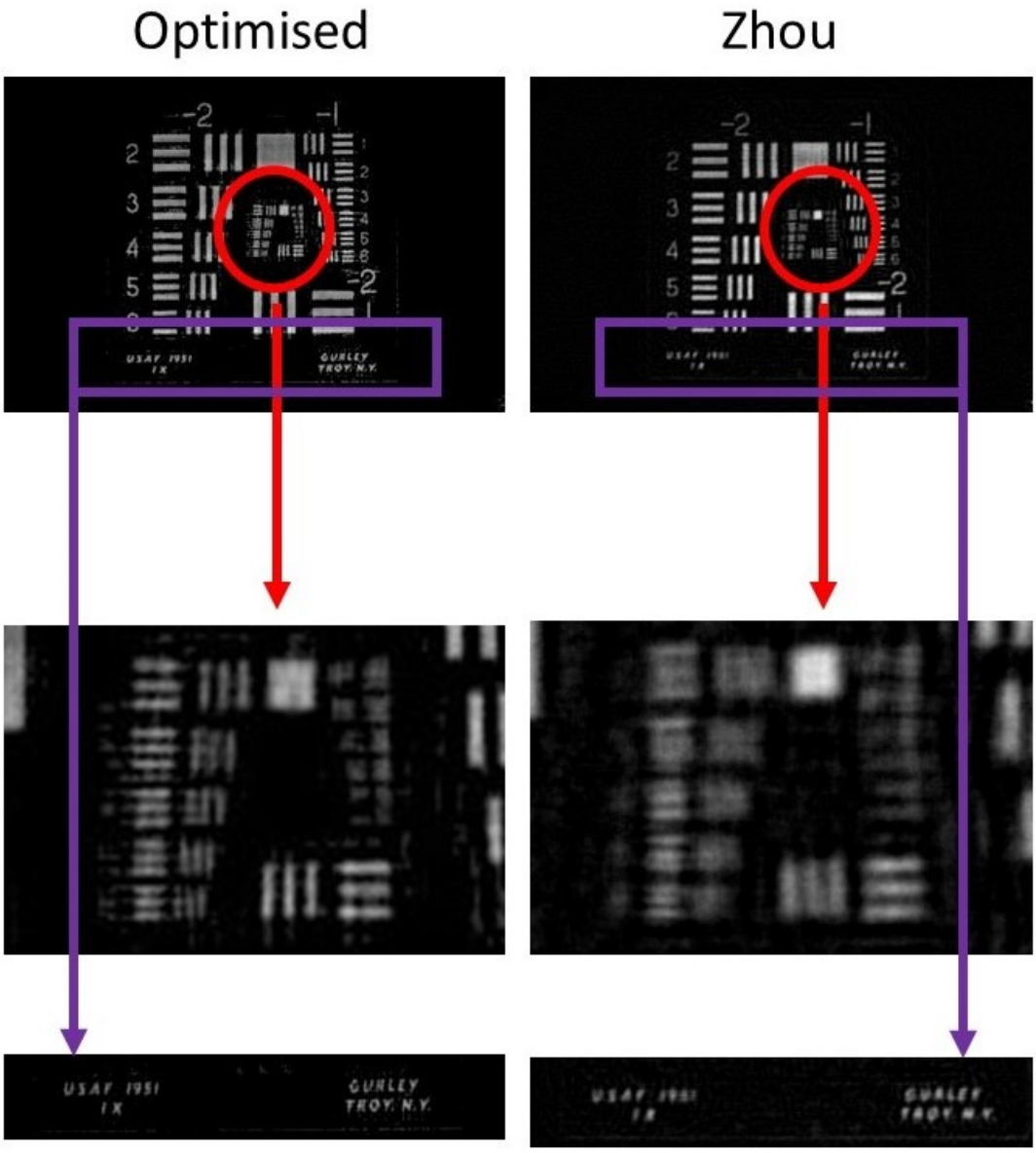


Figure 5.10: Defocus deblurring at 1.5m with close up of region.

5.2.5 Analysis

Online aperture optimisation

From table 5.2, the average execution time was around five minutes. Since the algorithm was run for a short time, it may not have had enough time to converge to the global minimum. However, we do not require a pattern that is the best but rather one that provides realistically suitable deblurring characteristics as compared to other patterns. Therefore, after five executions the pattern shown in figure 5.6 was the best performing pattern with the lowest $R(k, \sigma)$ value.

Comparison of deblurring apertures at 0.5m

Figure 5.7 shows the deblurring results at the focal distance of the camera. Since the target scene is already in focus, all the images taken appear very sharp with lots of detail present. The PSFs that were captured are all similar. They resemble tiny dots. The differences between the images are almost indistinguishable. This is due to the PSF being similar to that of an impulse response of the system and thus the image captured is the final image after deblurring. All of the groups in the resolution chart can be seen. The smallest element visible is group 1 element 6. This gives a maximum resolution of 3.56 line pairs per millimetre.

Comparison of deblurring apertures at 1.0m

Figure 5.8 shows the deblurring results at 1m from the camera. As we can see, the images taken with all three apertures show a significant amount of blur present. The PSFs that were captured now start to represent the shapes of the apertures that were used.

For the circular aperture, the image taken is significantly blurred but not as severely as the images taken with the other apertures. Group -2 is still visible with the elements distinguishable from each other. When the image is deblurred, detail is recovered. However, lots of noise is present in the image, which makes it appear grainy. The smallest element visible is group 0 element 2. This gives a maximum resolution of 1.12 line pairs per millimetre. The deblurring process has also introduced ringing artefacts which have degraded the image.

For the Zhou aperture, the image taken is significantly more blurred than for the circular aperture. The deblurred image, however, is much cleaner than the one deblurred from the circular aperture. Little noise and ringing is present in the image. The smallest element visible is group 1 element 2. This gives a maximum resolution of 2.24 line pairs per millimetre. The writing at the bottom of the image is also clearly visible and readable.

For the Levin aperture, the image taken is also significantly more blurred than for the circular aperture image. Lots of detail is not present in the image. The deblurring process does introduce ringing and amplification of noise into the image but is significantly less than that in the circular deblurred image. The smallest element visible is group 0 element 5. This gives a maximum resolution of 1.59 line pairs per millimetre. The writing at the bottom of the image is also visible and readable.

For the optimised aperture, the deblurring process has amplified very little noise with almost no ringing. The smallest element visible is group 1 element 3. This gives a maximum resolution of 2.52 line pairs per millimetre. Thus it is the best performing aperture in terms of resolving power. The writing at the bottom of the image is also clearly visible and readable.

Comparison of deblurring apertures at 1.5m

Figure 5.9 shows the deblurring results at 1.5m from the camera. The blur present in all the images is too significant to recognise any of the lines and numbers. Since the target scene is further away, the resolution chart appears smaller in the image. This reduces the level of detail present. The PSFs are all larger in size and their shapes are quite distinct in that they are identical to the aperture patterns.

For the circular aperture, the deblurred image has severe ringing. There is lots of noise present in the image. The smallest element visible is group -1 element 3. This gives a maximum resolution of 0.630 line pairs per millimetre. The writing at the bottom of the image is barely visible due to the ringing.

For the Zhou aperture, the deblurring process has recovered significant detail in the image. There is little to no ringing present and very minimal noise amplification. The text at the bottom of the image is present but not clear. The smallest element visible is group 0 element 2. This gives a maximum resolution of 1.00 line pairs per millimetre.

For the Levin aperture, severe ringing and noise is present in the image. However, details

have been recovered. The smallest element visible is group -1 element 6. This gives a maximum resolution of 0.891 line pairs per millimetre. Therefore it performed better than the circular aperture but not as well as the Zhou aperture.

The improvement of the optimised aperture over the other two apertures can be seen clearly at a deblurring distance of 1.5m away from the camera. The smallest element visible is group 1 element 1. This gives a maximum resolution of 2.00 line pairs per millimetre. This is double the resolving power of the Zhou aperture. Figure 5.10 shows a close up of group 0 and 1 for the optimised and Zhou apertures. Many of the smaller lines are still distinguishable from each other. The text at the bottom of the image is clearer and more readable with the optimised pattern than with the Zhou aperture.

5.2.6 Conclusions

The optimised aperture outperformed all the other apertures for defocus deblurring. It offered superior resolving ability, with little noise amplification and ringing introduced. The camera itself also offered good images and supported the deblurring process. Thus being able to optimise an aperture pattern online is possible using a programmable aperture mask.

5.3 Experiment 3 : Depth estimation

This section details the experimental procedures carried out for estimating depth in an image.

5.3.1 Aim

The aim of this experiment was to determine how well the system could estimate depth in an image using an optimised aperture. The optimised aperture is compared to existing apertures that are known to be good for either estimating depth or defocus deblurring. Conventional cameras are not able to calculate depth accurately as their circular apertures do not easily support the depth recovery process. Coded apertures are better at determining depth of objects in an image. The programmable aperture allows for the

depth recovery mask to be implemented quickly. This means that depth can be estimated promptly when needed.

5.3.2 Apparatus

For this experiment, the following equipment was needed:

1. A target scene.
2. A movable tray table.
3. A vertical stand.
4. Camera tripod.
5. Computing platform (described in section 3.5).
6. PSFs captured in section 5.1.
7. Different aperture pattern images (stored on the computing platform).
8. Depth estimation algorithms discussed in section 4.3.

5.3.3 Method

First the aperture was optimised for depth estimation. It was then compared to other apertures.

Aperture optimisation

The same parameters as stated in section 5.2.3 were used to optimise the aperture for depth estimation.

Comparison of apertures for depth estimation

This test used the target scene shown in figure 5.4 and the experimental setup illustrated in figure 5.5. The test chart was mounted on a vertical stand and placed on the movable

tray. The tray table with the target scene was moved to 0.5m, 1.0m and 1.5m distances away from the camera. The camera was mounted on a tripod and placed in one corner of the room. It was connected to the computing platform which allowed for remote capture of images and generation of different aperture patterns on the LCD.

Images were then captured using the optimised aperture and apertures mentioned previously in figure 5.1. Thereafter these images were given to the depth estimation algorithms (mentioned in section 4.3) to determine how far away the objects are from the camera.

5.3.4 Results

Aperture optimisation

The algorithm was run five times and the pattern shown in figure 5.11 was the best performing.

Figure 5.11: Depth optimised aperture pattern.

Comparison of apertures for depth estimation

The results of depth estimation for different apertures are presented below.

5.3.5 Analysis

In figure 5.12 'Circular', we have the results of the depth estimate using a circular aperture. The target placed at 0.5m away from the camera and the graph is maximised at 0.6m. This is off by 0.1m. Although wrong, this is not a bad estimate. For the target placed at 1.0m, the graph shows a position of 1.4m, which is incorrect by 0.4m. For the target placed at 1.5m the graph is maximised at 1.2m. This is incorrect by 0.3m which means that it failed on 3 occasions to determine the correct depth.

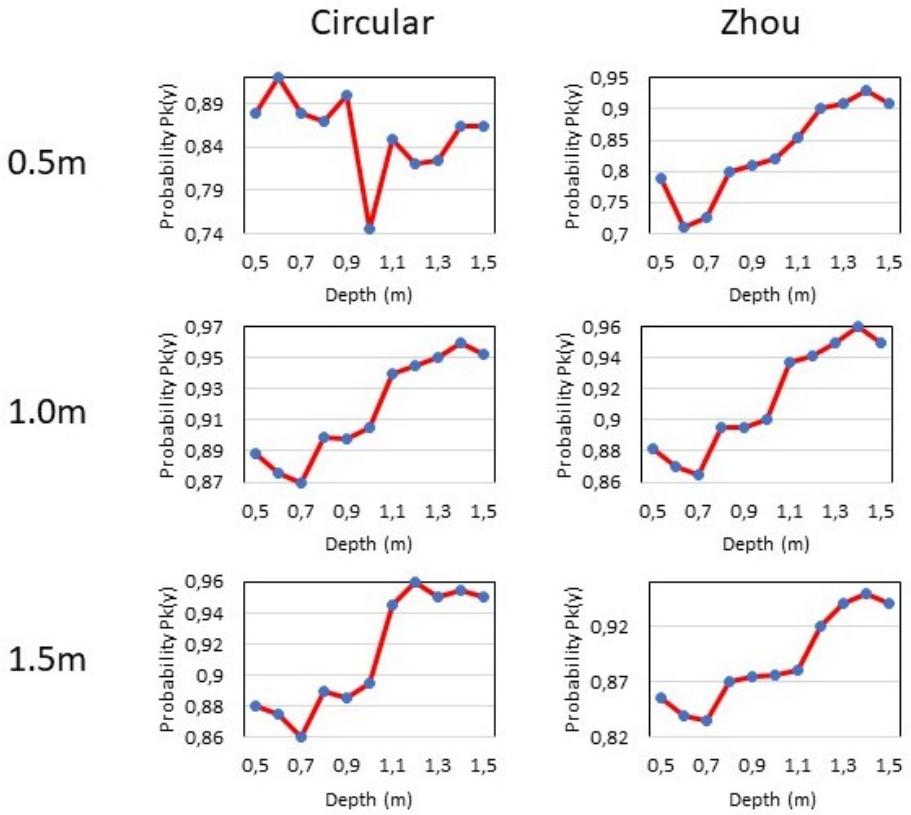


Figure 5.12: Estimating depth using circular and Zhou apertures.

The results of estimating depth using a non-depth optimised pattern is shown in figure 5.12 'Zhou'. Like the circular aperture, the algorithm fails to correctly determine the correct depths at all three levels: the algorithm always finds a maximum at 1.4m for all three depths. This could be due to the nature of the pattern that was optimised for defocus and thus does not have sufficient zero crossings in the frequency domain. Thus the algorithm struggles to determine the correct depth.

In figure 5.13 'Levin' we have the results of depth estimation using the pattern developed by Levin et al. If we look at the depth estimates for targets placed at 0.5m, 1m and 1.5m, we see that the graphs are correctly maximised at 0.5m, 1.0m and 1.5m respectively. This means that the algorithm was able to correctly identify the right blur scales at the various depths for the pattern. The sharpness of the peaks are also significant compared to the circular aperture. They make the correct depth estimate stand out from the rest. The algorithm distinctly separated the correct depth scale from the others.

Figure 5.13 'Optimised' shows the results of the pattern that was optimised for depth estimation based on image noise. At 0.5m, the algorithm correctly identifies the depth.

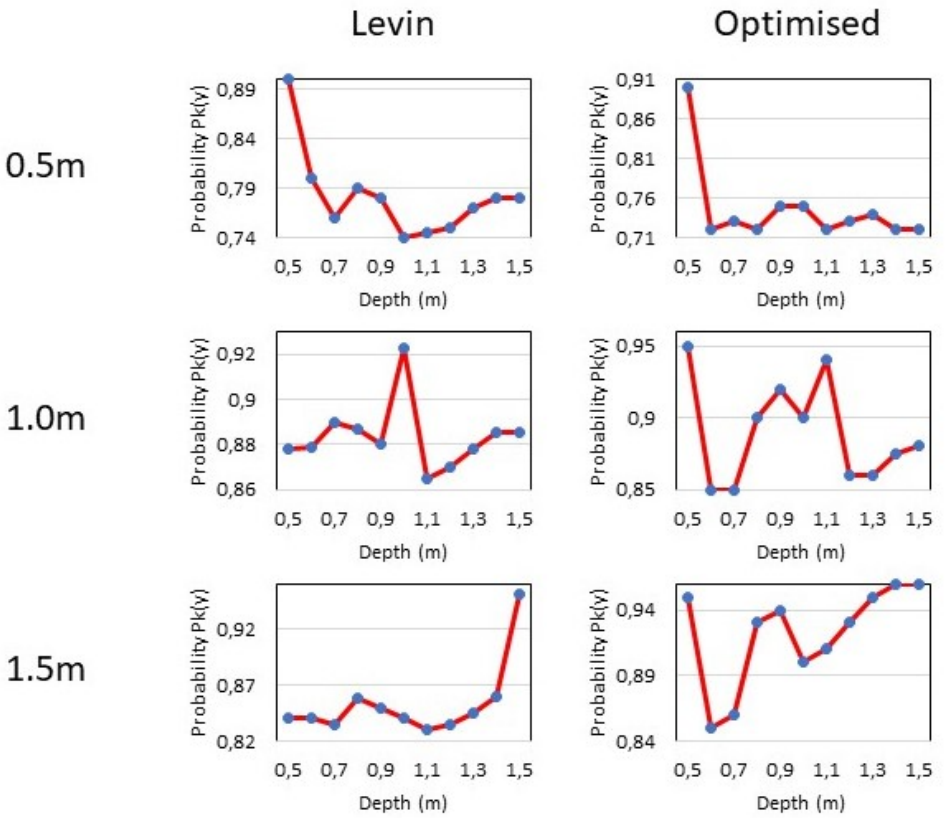


Figure 5.13: Estimating depth using Levin and optimised apertures.

The graph shows a clear maximum at 0.5m. For the target placed at 1.0m, the graph shows a maximum again at 0.5m but with a peak also present at 1.1m. If we only consider the peak at 1.1m then the estimation is off by 0.1m. Finally for the target at 1.5m, the graph is correctly maximised at 1.5m. Thus the depth optimised aperture managed to correctly identify two out of three depth scales.

5.3.6 Conclusions

The depth optimised pattern performed better than both the circular aperture and the aperture optimised for defocus deblurring. However, the pattern developed by Levin et al. [2] outperformed all of the other patterns as it correctly identified the depth scales on all three occasions. Therefore, a generic pattern proved to be superior to an optimised one.

5.4 Experiment 4 : Light field capture

This section details the experimental procedures carried out for capturing, processing and storing a four dimensional light field.

5.4.1 Aim

The aim of this experiment was to determine how well our system was able to capture, process and store a four dimensional light field. Capturing a light field with a conventional camera is not possible. Capturing a light field with static aperture masks is a tedious process. Having a reconfigurable aperture mask allows one to capture a light field in the fraction of the time with no prior aperture construction needed. Constructing a light field involves the capturing of images with different aperture shapes. Put simply, a captured light field is a two dimensional array of two dimensional images [15]. The ability of the programmable aperture camera to capture these images was investigated.

5.4.2 Apparatus

For this experiment, the following equipment was needed:

1. A few random objects of various sizes and shapes to form a test scene.
2. A tray table.
3. Camera tripod.
4. Computing platform (described in section 3.5).
5. Algorithms discussed in section 4.4.
6. Nine different aperture masks (generated in software).

5.4.3 Method

Capturing a light field

Figure 5.14 shows the general overview of the how the experiment was set up. Objects were placed at various distances from the camera and imaged using multiple apertures over time. Both the camera and the scene remain stationary with only the apertures being programmed electronically.

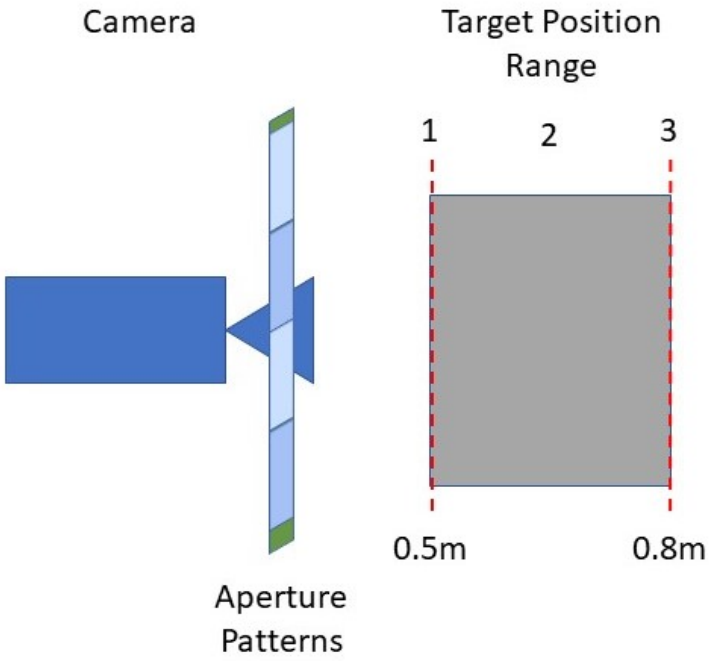


Figure 5.14: Light field capture experimental setup.

Figure 5.15 shows the setup of objects used in the light field capture experiment. The objects and their respective distances and positions in front of the camera are shown. The objects that were used were:

1. A globe of the world.
2. A Nescafe glass coffee container.
3. A coffee mug with a picture of a dog.
4. A Freshpak tea box.

There were also various stands and cardboard boxes used to adjust the objects to the correct heights, but they are not considered in the results as they were only used for object placement. Figure 5.15 (a) shows the side view of the placed objects. The front

view of the objects is shown in figure 5.15 (b). In figure 5.15 (b) the Freshpak box is closest to the camera, with the coffee mug behind it, followed by the Nescafe container and finally the globe of the world. This is supported in figure 5.15 (a).

(a) Side view. (b) Front view.

Figure 5.15: Experimental setup of various objects at different distances from the camera.

Figure 5.16 shows the different aperture patterns used to capture the required light field. The entire aperture is divided into nine blocks (3×3). Each block is opened, one at a time, and an image captured. Care is taken not to move the camera as the perspective shifts should only be caused by the different apertures.

Generating virtual photographs

The virtual photographs are generated using a Matlab implementation of the algorithm developed by Wilson et al. [15]. The required input to the algorithm is a given lightfield, a specified aperture shape, and a distance α between the virtual image plane and the aperture plane. The resulting output is a virtual photograph. Since the value of α is normalised, any value above one represents a virtual focal plane closer than the original and any value less than one indicates a virtual focal plane further from the original focal plane.

Figure 5.17 shows the virtual aperture shapes that were used to generate virtual photographs. Figure 5.17 (a) shows a pinhole aperture, 5.17 (b) a fully open aperture, 5.17 (c) a diagonal aperture and 5.17 (d) a random pattern aperture. A similar testing approach was adopted from Wilson et al. [15], where the value of α was varied over the range 0.98 to 1.02 in increments of 0.001 and the results recorded.

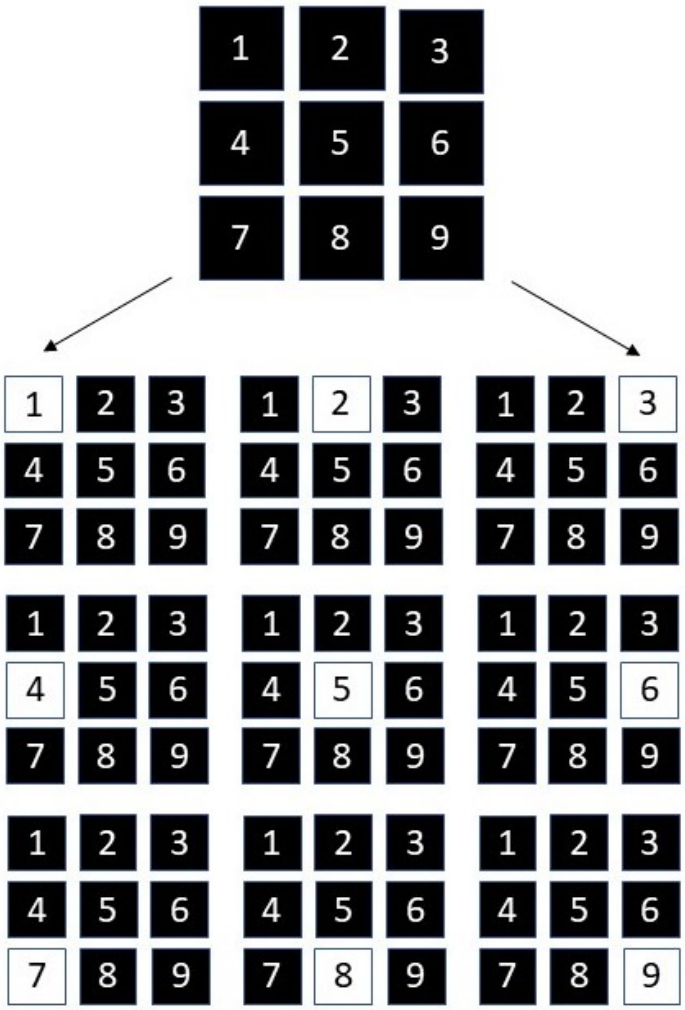


Figure 5.16: Aperture shapes with a 3×3 angular resolution used to capture the light field.

Refocusing

The refocusing was carried out in a similar way to generating virtual photographs mentioned above. In this implementation the entire aperture shape was used (fully open) to refocus on specific objects in the scene while blurring out others. This was done using the techniques described in section 4.4.2.

Stereo vision

Generating a stereo disparity map, as explained in section 4.4.3, involves using two images taken of the same scene from slightly different perspectives (viewing angles). Since a light field was already captured in the previous section, one is able to make use of a suitable

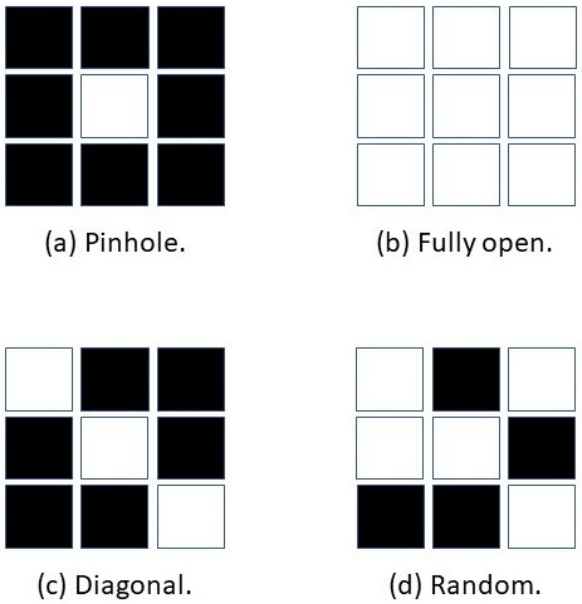


Figure 5.17: Virtual aperture shapes used to generate virtual photographs.

image pair from the light field to generate a stereo disparity map.

The image pair that was chosen corresponds to $(u, v) = (1, 2)$ (right image) and $(u, v) = (3, 2)$ (left image). This is shown in figure 5.18. These images are aligned in the horizontal direction and therefore can be directly input to any suitable stereo matching algorithm. They also offer a good horizontal distance between openings and thus provide good disparity between images. The algorithm used was developed by Abbeloos [40] in Matlab. It produces a depth map using just two images, namely a left and right image pair.

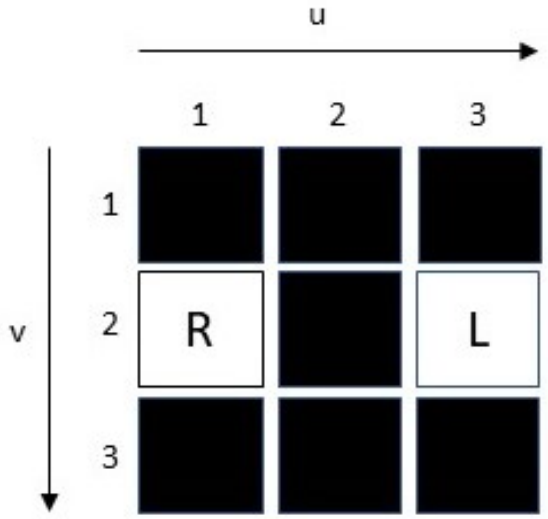


Figure 5.18: Left and right image pair from light field. Right image corresponds to $(u, v) = (1, 2)$. Left image corresponds to $(u, v) = (3, 2)$.

5.4.4 Results

The results from light field capture are presented below.

Capturing a light field

Figure 5.19 shows the captured light field using the different aperture openings on the programmable mask.

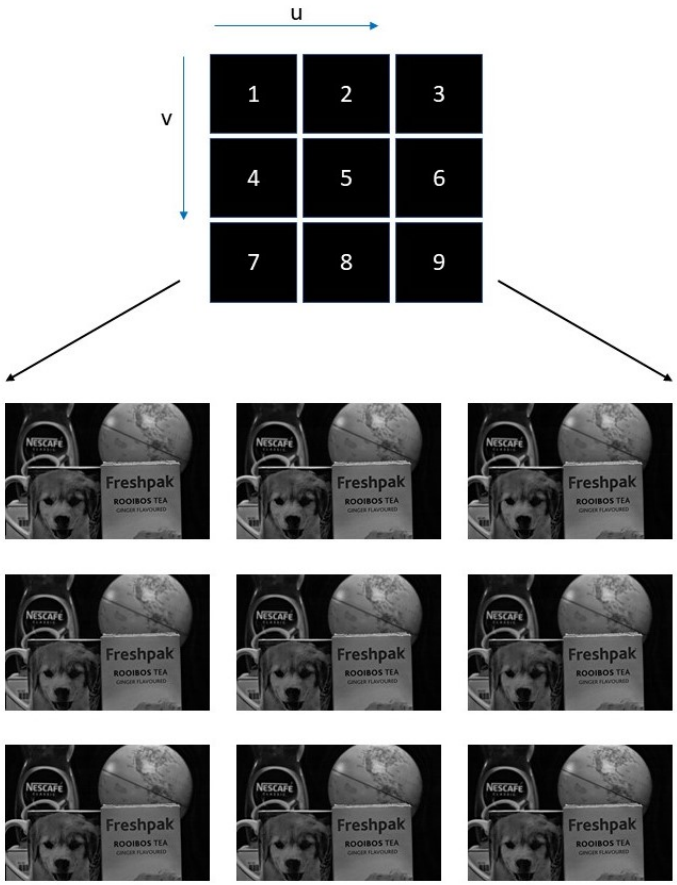


Figure 5.19: Captured light field with uv indexing.

Figure 5.20 shows the actual image dimensions being expressed in an xy indexing.

Refocusing

Figure 5.21 shows the results of the refocusing experiments carried out using $\alpha = 0.99$. By increasing α to 1.00, we see the resultant image formed in figure 5.22. Finally, by

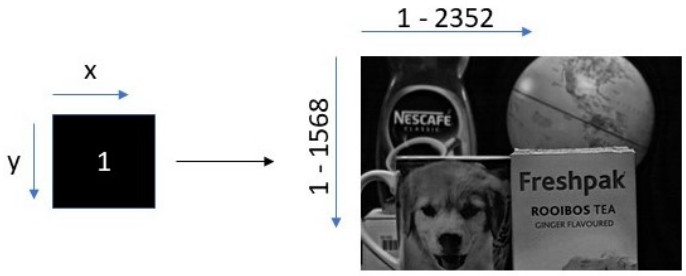


Figure 5.20: Subset of light field captured with xy indexing.

making $\alpha = 1.01$ a new image is formed and shown in figure 5.23.

Figure 5.21: Refocusing at $\alpha = 0.99$.

Figure 5.22: Refocusing at $\alpha = 1.00$.

Figure 5.23: Refocusing at $\alpha = 1.01$.

Generating virtual photographs

Figure 5.24 shows the virtual photos generated using synthetic apertures.

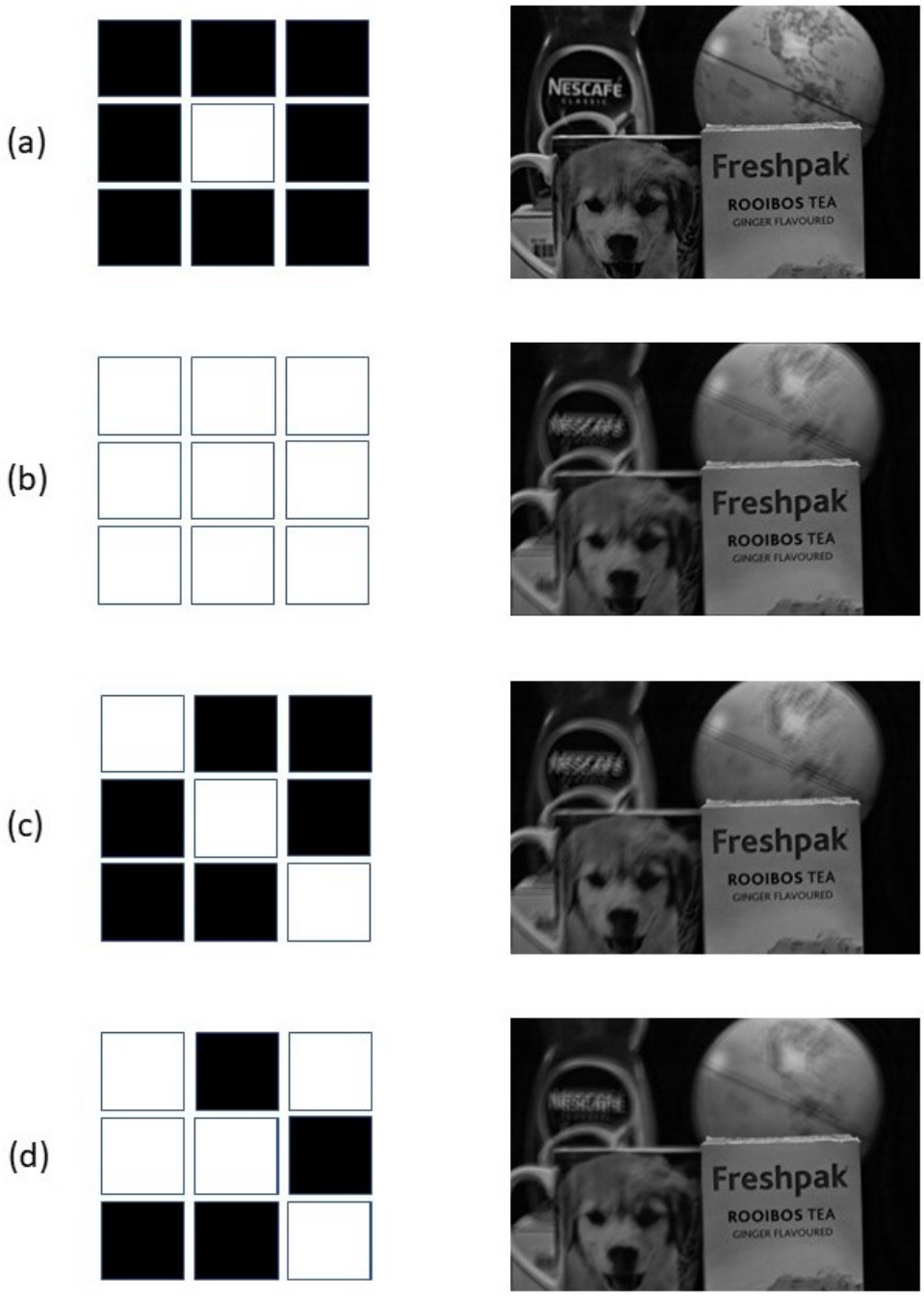


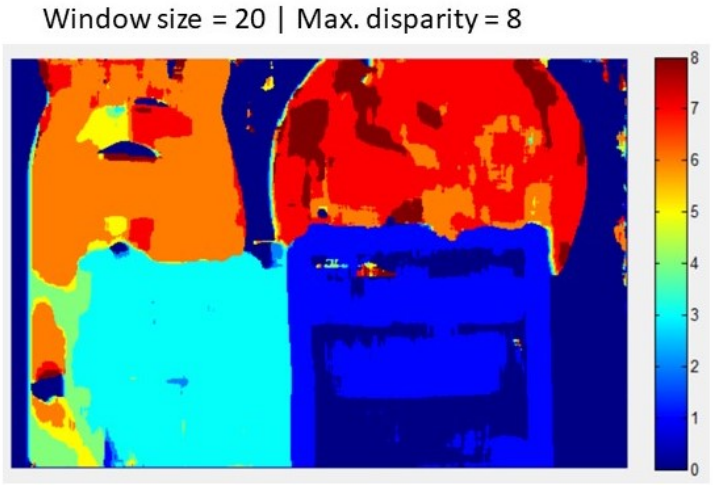
Figure 5.24: Virtual photographs generated with synthetic apertures.

Stereo vision

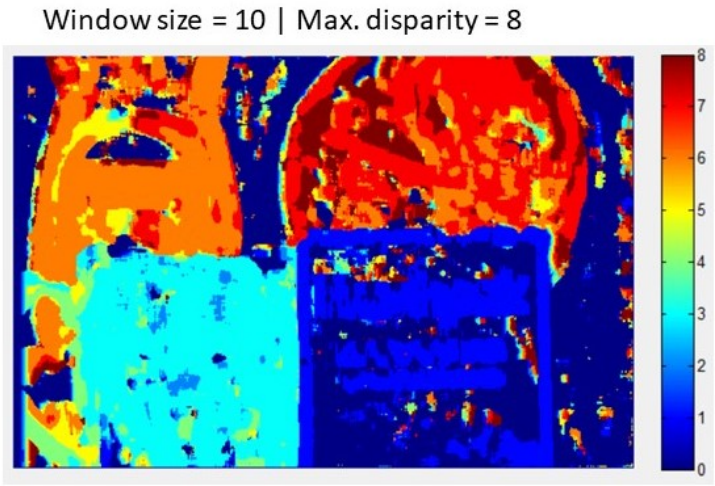
Figure 5.25 shows the images used to generate the stereo depth map. Figure 5.26 shows the depth maps generated from the stereo image pair.

(a) Left image. (b) Right image.

Figure 5.25: Stereo image pair.



(a)



(b)

Figure 5.26: Stereo disparity maps generated from the image pair.

5.4.5 Analysis

Capturing a light field

Figure 5.19 shows the captured light field under uv indexing. Here we have an array size of 3×3, which translates to nine images that were taken when one corresponding portion of the aperture was opened at a time.

The xy indexing involves the actual dimensions of the images in the light field. This means we have an array size of 2352×1568 as shown in figure 5.20. By combining the uv and xy indexing methods we can locate a pixel in any image within the light field.

Refocusing

The captured light fields are aggregated and refocused onto specific parts of the scene. The entire light field was used to carry out the refocusing as this maximised the angular resolution.

In figure 5.21 (a) we have a refocusing using $\alpha = 0.99$ utilising the entire light field. Here we can see that specific portions of the scene appear more in focus than others. For instance, in figure 5.21 (b) we see a crop of the Nescafe bottle with the words clearly readable and focused. In figure 5.21 (c) the coffee mug with the picture of the dog appears slightly blurred. The Freshpak box in figure 5.21 (d) appears very blurred. This is due to the relative positions of the objects in the scene.

In figure 5.22 (a) we have a refocusing using $\alpha = 1.00$. In this instance the coffee mug with the picture of the dog appears in focus in figure 5.22 (c), while the Nescafe bottle in figure 5.22 (b) has now been slightly blurred. The Freshpak box in figure 5.22 (d) has been brought closer to focus, but still appears blurry.

In figure 5.23 (a) we have a refocusing using $\alpha = 1.01$. In this instance the Freshpak box is in focus in figure 5.23 (d), while the Nescafe bottle is severely blurred in figure 5.23 (b). The coffee mug with the dog on it has been taken out of focus and looks slightly blurred as compared to figure 5.22 (c).

Generating virtual photographs

As stated in section 5.4.3, four virtual aperture shapes were used. The most notable results can be seen in figures 5.24 (a) and (b).

In figure 5.24 (a) we have a pinhole aperture shape. We can see the resulting image has almost all the objects in the scene in focus, which is expected as a pinhole aperture has a large DOF. All of the objects appear sharp and the text on most of the objects is readable.

In figure 5.24 (b) we have an aperture fully open, with most of the objects blurred, apart from the Freshpak box which is at the focal length of the lens. This is expected as the larger the aperture size, the shallower the DOF.

For the diagonal pattern in figure 5.24 (c) and the random pattern in figure 5.24 (d), we have random blur appearing throughout the image with some parts appearing clear and some not. By opening up more aperture blocks we improve the image quality by letting in more light, but sacrifice the DOF.

Stereo vision

As stated in section 5.4.3, the two images required to perform depth from disparity were taken from the captured light field in the previous section and correspond to $(u, v) = (1, 2)$ (right image) and $(u, v) = (3, 2)$ (left image). The image pair that was chosen can be seen in figure 5.25. Before they could be fed into the stereo matching algorithm, the maximum pixel disparity as well as window size have to be specified. Maximum pixel disparity corresponds to objects closest to the camera and was estimated to be eight pixels. Figure 5.26 shows two depth maps generated from the algorithm. In (a) we see the depth map generated with a window size of 20 and in (b) we have a window size of 10. The smaller the window size the higher the resolution of the map generated, but the higher resolution increased the number of false matches. Thus a compromise had to be made between resolution and accuracy when choosing window size.

5.4.6 Conclusions

From the results obtained, it was evident that the camera captured, processed and stored a light field successfully. Using the captured light field we were able to perform refocussing, generate virtual photographs and extract depth using stereo vision. The refocussing allowed us to focus on certain portions of the image while blurring out the rest. This allowed us to highlight certain portions of the image when needed. Generating virtual photographs using synthetic apertures allowed us to create images in software that ordinarily would have only been formed with physical apertures in a lens. Stereo vision successfully extracted the correct depth scales of objects in the scene and created a useful depth map. Normally one would require two cameras to perform stereo vision but this was done with one by simply varying the aperture. Thus the programmable aperture camera performed well as a light field camera.

5.5 Experiment 5 : Exposure correction

The experimental description of exposure correction is described below.

5.5.1 Aim

The aim of this experiment was to determine how well the programmable aperture camera adjusted to various levels of light exposure. The aperture determines the amount of light that enters the system. Having an image that is over or underexposed is undesirable. The camera should be able to detect the light level and adjust the aperture size accordingly. This allows the correct amount of light into the system.

5.5.2 Apparatus

For this experiment, the following equipment was needed:

1. A target scene.
2. Camera tripod.
3. Variable light source.

4. Computing platform (described in section 3.5).

5. Exposure correction algorithms discussed in section 4.5.

5.5.3 Method

A simple method of testing the exposure correction algorithm is to illuminate a target scene with a variable light source and monitor how the algorithm adjusts to the illuminated objects. The target scene should be both over and under illuminated and the results recorded. Figure 5.27 below illustrates the experiment.

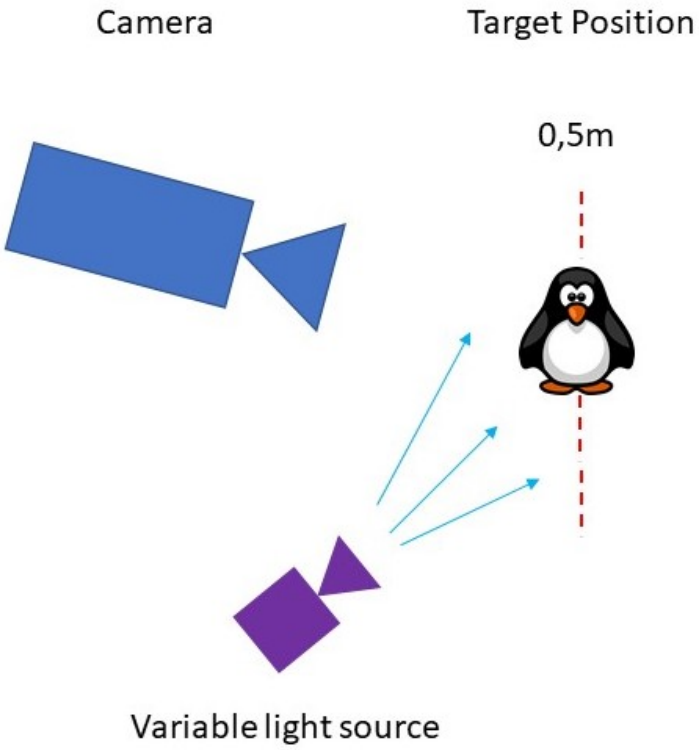


Figure 5.27: Experimental setup for testing exposure correction.

The variable light source is responsible for providing the illumination and can be adjusted to three different intensity levels. A simple lamp was used as the light source since it was readily available and easy to mount and set up. Many other more accurate light sources could have been used that provide better uniform illumination, but due to time and resource constraints the lamp was used. Images were then taken according to the different light levels and the response of the algorithm to these light levels was recorded. For simplicity, the same target scenes used in the previous sections were used here.

5.5.4 Results

To determine the effectiveness of the algorithm two scenes were tested at different exposure levels and the images at each stage of the algorithm were recorded.

Figure 5.28 shows the results of the exposure correction for an underexposed resolution chart scene.

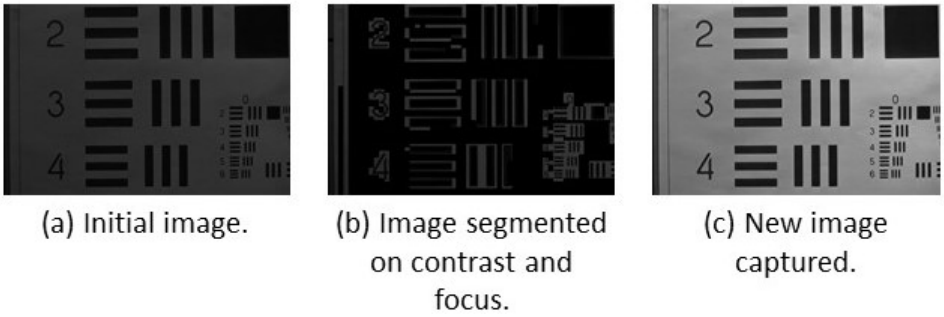


Figure 5.28: Exposure correction algorithm on resolution target scene.

Figure 5.29 shows the physical size adjustment of the aperture opening.

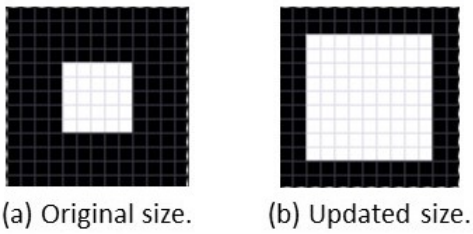


Figure 5.29: Aperture size adjustment.

Figure 5.30 shows the results of exposure correction for an underexposed complex scene.

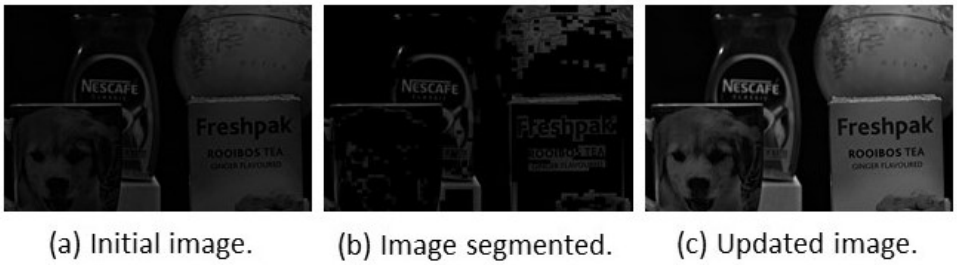


Figure 5.30: Exposure correction algorithm on underexposed complex scene.

Figure 5.31 shows the results of exposure correction for an overexposed complex scene.

Figure 5.32 shows the results of exposure correction for an overexposed resolution chart scene.

Figure 5.31: Exposure correction algorithm on overexposed complex scene.

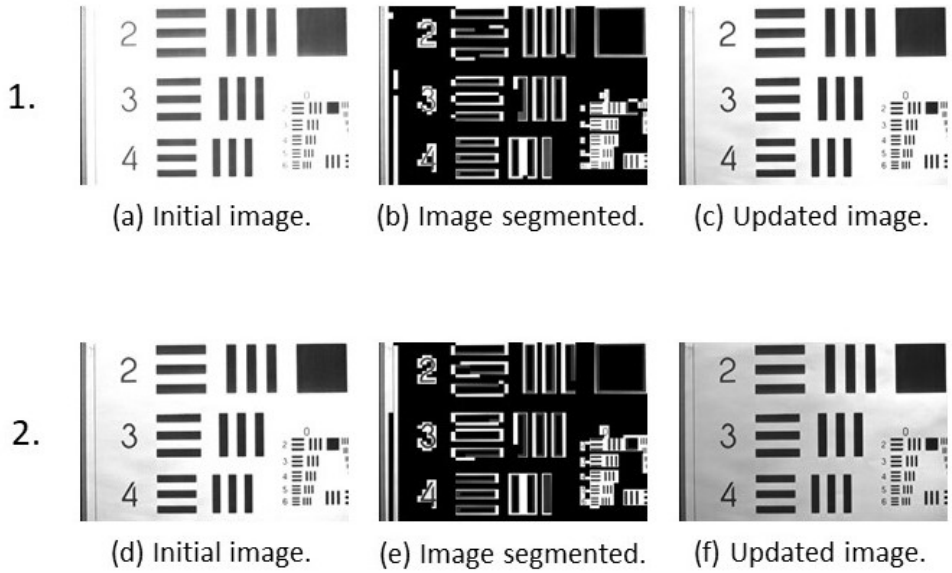


Figure 5.32: Two iterations of the iris adjustment algorithm.

5.5.5 Analysis

Figure 5.28 (a) shows a resolution chart image captured using an arbitrary aperture size. The image is underexposed and dark. It is then segmented based on contrast and focus features, shown in figure 5.28 (b). Once this is complete, the algorithm determines the brightness level and thus the EV value of the image. The optimum EV is calculated next, and from this value a more appropriate aperture size is determined. The aperture size is then updated and an image captured. Figure 5.28 (c) shows the image captured using the updated aperture shape. The image is brighter and more details are visible. Although the brightness in the image is not completely uniform, it is still brighter than the original. The non-uniform brightness is most likely caused by the lamp used to illuminate the target. To find the optimal aperture size took only one iteration of the algorithm. Figure 5.29 shows the aperture size before and after the algorithm was applied.

Another test to determine the efficiency of the algorithm is to use a more complex scene rather than a standard resolution chart. This scene consists of the same objects used in

section 5.4 and was used again to test the algorithm efficiency. Figure 5.30 (a) shows the initial image of the scene captured at a very low exposure. In figure 5.30 (b) we see the image segmented for good contrast features and focus. The resulting image after updating of the aperture size is shown in figure 5.30 (c). The image is now brighter with more features present.

The same scene is captured again but this time at a greater illumination level, as shown in figure 5.31 (a). The algorithm is repeated and the resultant image can be seen in figure 5.31 (c) after the aperture size has been updated. The image brightness is now at the correct level.

Figure 5.32 (a) shows the same resolution chart scene used earlier but this time with the scene overexposed and saturated. The image is then segmented to extract features based on contrast and focus as shown in figure 5.32(b). Figure 5.32 (c) shows the updated image captured with the new aperture size. The image, however, is still overexposed, and thus the algorithm repeats again.

Figure 5.32 (d) shows the same image as in figure 5.32 (c) but this time the algorithm uses this image as the initial image. Figure 5.32 (e) shows the image segmented and features extracted. Figure 5.32 (f) shows the resultant image. This image, although not uniformly exposed, has a much better exposure than figure 5.32 (a), with more details visible. Thus the algorithm took two iterations to correctly determine the aperture size for the right exposure level.

5.5.6 Conclusions

The reconfigurable aperture camera successfully updated the aperture opening size based on the level of light. If the image was underexposed the aperture would increase the opening to allow more light in. If the image was overexposed the aperture size would decrease. This means that the camera was successful in correctly adjusting the exposure levels to capture the correctly illuminated scene.

Chapter 6

Conclusions and recommendations

This chapter provides a summary of the work as a whole and suitable improvements are discussed that could be implemented in the future. First the camera design and construction is summarised. Thereafter a discussion into coded aperture, light field and exposure correction applications is presented.

Programmable aperture camera

The programmable aperture camera was created with various off-the-shelf components. These components were slightly modified and interconnected. This kept the cost of producing the camera down. The components chosen were fairly cheap and easily available. The aperture shapes were either stored as images in software or generated when needed. These were displayed on the programmable mask. Unlike static masks that needed to be physically inserted, these masks could be implemented at the click of a button. This saved a lot of time in not having to pre-construct the shapes. The camera's position was not disturbed when changing masks as it did not require the camera to be physically touched. The generation of the masks and the capturing of images were tested and they functioned correctly.

There are many improvements that could be made with the programmable aperture camera. Firstly, a better adaptive mask technology could be chosen. At this point there are only a few SLMs that are viable choices. In the future there could be many more that offer improved resolution, light throughput and functionality. If one is able to optically solve the diffraction caused by the DMD, then it could be used as a programmable mask. The actual camera that was bought, before modification, was chosen due to budget and

time costs. Future cameras should have improved resolution and features at a lower cost. This would improve the image quality. The camera casing could also be redesigned to be more rugged. This would allow the camera to be used in environments outside of the laboratory. The computing platform used was a fairly large laptop computer. In future, a more compact computing system could be used such as a Raspberry Pi or Intel Edison system on chip. The entire system could also be ported to a FPGA.

Coded apertures

Coded aperture theory was initially discussed with the main focus on depth estimation and defocus deblurring. The mathematical principles were presented, which helped to devise performance metrics and optimise the aperture online for defocus deblurring. Existing and optimised apertures were used to photograph a target scene. These images were then input into depth estimation and deblurring algorithms. The outputs of these algorithms were analysed and discussed.

For defocus deblurring, the aperture optimised online outperformed existing generic apertures for the same application. It was able to resolve the target scene better while also limiting ringing and noise levels. All the coded masks performed better than the circular aperture and thus confirmed that coded apertures are much better than circular apertures for deblurring. The aperture optimisation algorithm was successful in optimising the aperture online.

For depth estimation, an optimised pattern was compared to existing coded aperture masks. The mask that was optimised for depth outperformed the circular aperture and the aperture optimised for deblurring. However, it did not perform as well as the generic mask developed for depth. The main reason for the optimised aperture's poor performance is the process used to optimise the aperture. Currently the aperture is optimised based on image noise. It could be that the incorrect noise level was chosen which lead to the incorrect pattern being developed and used.

The biggest improvement that could be done is to generate an aperture instantaneously to perform tasks such as deblurring and depth estimation. Currently the depth optimisation process uses only image noise to update the aperture pattern. Future work would allow for the pattern to be optimised based on scene content. The algorithm does take time to run, so the result is not instantaneous. If it were, this would allow for all sorts of benefits in dynamic scenes. Thus a faster implementation of the algorithm is needed. Coded apertures are also used for astronomy and compressed superresolution applications.

Having an adaptive mask in those applications could be investigated.

Light field capture

An introduction into light field theory was first presented. Thereafter, the mathematical principles and algorithms used were discussed. A 3×3 resolution light field was captured using the programmable aperture camera. This resulted in nine images that could be used for synthesising virtual photographs, refocussing or performing stereo vision.

Using the captured light field, four virtual photographs were created. This was done using various combinations of images in the light field. The generated photographs appeared as though they had been captured with physical apertures. Instead they were generated in software.

The captured light field also allowed for digital refocussing. Various objects in the scene could be brought into focus or blurred out. This is useful in photography to highlight certain portions of the image while blurring out unimportant parts.

Finally, stereo vision was performed using the captured light field. Using a stereo image pair, a depth map was generated that showed the distances of objects in the scene relative to the camera. Normally one would require two cameras but this was done with only one.

The resolution of the light field captured was 3×3. This was done to keep the aperture openings large enough to let in sufficient light. This reduced the exposure time and limited the noise. The main reason for light loss was due to the low light transmissivity of the LCD. If a better SLM was used, this would allow for a higher resolution light field to be captured. This would have improved benefits in virtual refocusing and photograph generation.

Exposure correction

The reasoning behind exposure correction was presented first. Next the mathematical principles required to implement exposure correction were discussed. Thereafter it was tested using a target object and a variable light source. The algorithm was successful in adjusting the size of the aperture opening depending on the lighting levels in the room. It adjusted to correctly expose the images.

Exposure correction performed in this work used only the aperture size to control the amount of light entering the system. Future work could allow for both the aperture size and the exposure time to determine the correct EV. This would adjust the exposure levels without sacrificing DOF.

Bibliography

[1] R. Gonzalez and R. Woods, Digital image processing. New Delhi: Dorling Kindersley, 2014.

[2] A. Levin, R. Fergus, F. Durand and W.Freeman, "Image and Depth from a Conventional Camera with a Coded Aperture", ACM Transactions on Graphics, vol. 26, no. 3, pp. 70-1 to 70-10, 2007.

[3] "Lytro - Illum", Illum.lytro.com, 2017. [Online]. Available: https://illum.lytro.com/illum. [Accessed: 31- Mar- 2017].

[4] R. NG, M. Levoy, M. Bredif, G. Duval, M. Horowitz and P. Hanrahan, "Light field photography with a hand-held plenoptic camera", Computer Science Technical Report CSTR, vol. 2, no. 11, pp. 1-11, 2005.

[5] F. Pereira and E. da Silva, "Efficient plenoptic imaging representation: Why do we need it?", 2016 IEEE International Conference on Multimedia and Expo (ICME), 2016.

[6] D. Dansereau, "Plenoptic Signal Processing for Robust Vision in Field Robotics", Ph.D, The University of Sydney, 2014.

[7] E. Adelson and J. Bergen, "The Plenoptic Function and the Elements of Early Vision", Computational Models of Visual Processing, vol.1, no.1, pp. 3-20, 1991.

[8] P. Dragotti and M. Brookes, "Unsupervised Representation and Understanding of Multi-View Images", in 3rd SEAS DTC Technical Conference, Edinburgh, 2008.

[9] O. Cossairt, "Tradeoffs and Limits in Computational Imaging", Ph.D, Columbia University, 2016.

[10] D. Gutierrez, B. Masia and A. Jarabo, "Computational Photography", in CEIG - Spanish Computer Graphics Conference, 2012.

[11] S. K. Nayar, Computational camera: Approaches, Benefits and Limits. New York: Columbia Univ., 2011.

[12] C. Zhou and S. Nayar, "Computational Cameras: Convergence of Optics and Processing", IEEE Transactions on Image Processing, vol. 20, no. 12, pp. 3322-3340, 2011.

[13] C. Zhou, "Point Spread Function Engineering for Scene Recovery", Ph.D, Columbia University, 2012.

[14] C. Zhou and S. K. Nayar. What are Good Apertures for Defocus Deblurring? In IEEE International Conference on Computational Photography, April, 2009.

[15] M. Wilson and F. Nicolls, "Coded Aperture and Coded Exposure Photography: an investigation into applications and methods", MSc.Eng, University of Cape Town, 2011.

[16] S. Nayar, V. Branzoi and T. Boult, "Programmable Imaging using a Digital Micromirror Array", IEEE 2004, Columbia, 2016.

[17] H. Nagahara, C. Zhou, T. Watanabe, H. Ishiguro and S. Nayar, "Programmable Aperture Camera Using LCoS", Kyushu University, Japan, 2016.

[18] S. Suh, C. Choi, D. Park and C. Kim, "Efficient synthetic refocusing method from multiple coded aperture images for 3D user interaction", Computational Imaging XI, vol. 865786570, pp. 1-8, 2013.

[19] K.Lewis, "Adaptive masks for discriminative coded aperture imaging", in Adaptive Coded Aperture Imaging, Non-Imaging, and Unconventional Imaging Sensor Systems II, 2010, pp. 1-9.

[20] "Holoeye Photonics AG - Spatial Light Modulators", Holoeye.com, 2016. [Online]. Available: http://holoeye.com/spatial-light-modulators/. [Accessed: 12- May- 2016].

[21] Texas Instruments, "DLP System Optics", Texas Instruments Incoporated, Texas, 2010.

[22] P. Zupancic, "Dynamic Holography and Beamshaping using Digital Micromirror Devices", MSc, Ludwig-Maximilians-Universitat Munchen, 2016.

[23] G. Bothun, "Physics 155", Physics of the internet, 2016. [Online]. Available: http://homework.uoregon.edu/pub/class/155/out155/dt2.html. [Accessed: 12- May- 2016].

[24] S. Ri, Y. Matsunaga, M. Fujigaki, T.Matui and Y. Morimoto, "Development of DMD reflection-type CCD camera for phase analysis and shape measurement", Optomechatronic Sensors and Instrumentation, vol. 6049, pp. 1 to 12, 2005.

[25] C. Palmer and E. Loewen, Diffraction grating handbook. Rochester, N.Y. Newport Corporation, 2005.

[26] Fujitsu, "Fundamentals of Liquid Crystal Displays How They Work and What They Do", Fujitsu microelectronics America Inc., California, 2006.

[27] D. Systmes, "3D ContentCentral", 3dcontentcentral.com,2017.[Online].Available: http://www.3dcontentcentral.com/Download-Model.aspx?catalogid=171&id=536065. [Accessed: 25- Sep- 2017].

[28] "Canon_EF_50_mm_f_1.8_II.jpg", Wikimedia Commons, 2017. [Online]. Available: https://commons.wikimedia.org/wiki/File:Canon_EF_50_mm_f_1.8_II.jpg. [Accessed: 25- Sep- 2017].

[29] "Canon EOS 500D", Wikipedia, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Canon_EOS_500D. [Accessed: 25- Sep- 2017].

[30] Matlab, MathWorks. [Online]. Available: https://www.mathworks.com/products/matlab.html. [Accessed: 17-Nov-2017].

[31] OpenCV library, Open Computer Vision. [Online]. Available: https://opencv.org/. [Accessed: 17-Nov-2017].

[32] L. Mertz and N.O. Young, "Fresnel transformations of images", International Conference on Optical Instruments and Techniques, pp. 305310, 1961.

[33] "Pinhole camera", Wikipedia, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Pinhole_camera. [Accessed: 26- May- 2016].

[34] C. Liang, T. Lin, B. Wong, C. Liu and H. Chen, "Programmable aperture photography: multiplexed light field acquisition", ACM Transaction on graphics., pp. 1-10, August, 2008.

[35] M. Levoy and P. Hanrahan, "Light field rendering", 23rd annual conference on ACM Computer graphics and interactive techniques, Siggraph 96, pp. 3142, New York, USA, 1996.

[36] M. Kessel, "Surface Estimation From Multiple Images", MSc.Eng, Danish Technical University, 2006.

[37] J. van der Merwe, "An evaluation of local two-frame dense stereo matching algorithms", MSc, University of Johannesburg, 2011.

[38] S. Battiato, G. Messina, A. Castorina, "Exposure correction for imaging devices: an overview", Single-sensor imaging: Methods and Applications for Digital Cameras, pages 323-349, 2008.

[39] "Exposure — Understanding Exposure - ISO, Aperture and Shutter Speed Explained", Exposureguide.com, 2017. [Online]. Available: https://www.exposureguide.com/exposure.htm. [Accessed: 25- Oct- 2017].

[40] W. Abbeloos, "Real-Time Stereo Vision", MSc, Karel de Grote-Hogeschool University College, Belgium, 2010.

[41] J. P. Rice, J. E. Neira, M. Kehoe, and R. Swanson, "DMD diffraction measurements to support design of projectors for test and evaluation of multispectral and hyperspectral imaging sensors", Emerging Digital Micromirror Device Based Systems and Applications, Dec. 2009.

[42] Texas Instruments, "Using Lasers with DLP DMD technology", Texas Instruments, 2008.

[43] M. Sheik-bahae, Fourier Optics, 1st ed. New Mexico: University of New Mexico, pp. 1-9, 2017.

[44] Marianne Breinig, "Diffraction", Modern Optics. [Online].Available: http://electron9.phys.utk.edu/optics421/modules/m5/Diffraction.htm.[Accessed: 22- Mar- 2017]

[45] A. Cable, "Resolution Test Targets", Thorlabs.com, 2018. [Online]. Available: https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=4338. [Accessed: 17- Jan- 2018].

Appendix A

Addenda

A.1 Software considerations

Like hardware, there are several software tools that can be used to create a programmable aperture camera. This section deals with the selection and use of such software tools.

A.1.1 Image capture

Image capture is a fundamental step in the coded camera design and there are several software tools that one could use to achieve the results. The two main software tools that were used were:

1. Canon EOS Windows GUI application software.
2. Canon EOS C++ software development kit (SDK).

The first choice, the Windows application, was provided by the manufacturer of the camera. This is simply installed on the Windows operating system and will automatically detect the camera connected to the computer. The software provides an easy to use, attractive interface that allows for remote capture, i.e. the camera can be triggered to capture an image remotely via the software. It also allows one to change camera parameters such as exposure, shutter speed, image size etc. Another important feature

is live view, which allows a user to view the image before it is captured and stored either in camera memory or on the computing platform.

The second choice is more robust in that it allows a user to develop his/her own application that utilises the camera. The SDK is not freely available to all users but if requested from the manufacturer, given the right motivation, can be acquired and used. The SDK allows one to embed camera functions into C++ functions such that the camera can be triggered, parameters changed or images stored using standard C++ code and not the GUI provided by the manufacturer.

A.1.2 Image processing

There are several image processing packages that are available to use to modify images. The two that were used in this project were:

1. Matlab R2014a.
2. OpenCV.

Matlab R2014a is software that was developed by MathWorks Inc [30]. It is a computational tool that allows for the rapid development of various applications and supports easy processing of data. Matlab has an image processing toolbox that allows one to quickly and easily import images for manipulation. This made testing and in the early stages easy and effective. The Matlab toolbox does also allow for the control and acquisition of cameras and images but the R2014a version of Matlab does not support the camera owned by the author. Thus the system has to be ported to a full C++ implementation using the camera SDK provided by the manufacturer.

Since the camera control and image acquisition was done using the camera SDK in C++, it gave room for the image processing to also be done in C++. This allowed for the use of OpenCV (Open Computer Vision) [31] which is an open source image processing framework. It makes for easy image manipulation techniques, such as FFTs, DFTs etc. much like the implementations given in Matlab. It is free to download and use and has a wide community of support thus making it ideal for the project.

A.1.3 Aperture programming

The LCD screen used, together with the development board, was purchased from the manufacturer. The manufacturer also provides software, in the form of a Windows application to control and manipulate the content displayed on the screen. This is fine for manually entering patterns to be displayed but the downfall is that one requires the screen patterns to be updated autonomously. Since the manufacturer does not provide an SDK, another method for updating the LCD content had to be used.

Since the LCD software kept reading the image to display from a particular folder with a specific name, the solution was to simply write a script that kept replacing the image in that folder with another image of the same name. This caused the LCD to update continuously without user interaction. This solution, while not the most efficient, proved effective and worked well for the project and thus was used.

A.2 3D printed part

The mechanical component that holds the back lens within the housing was damaged as a result of the lens deconstruction process. To compensate for this, a similar part was designed and 3D printed to accommodate the back lens within the housing. This allowed for simple yet rapid development. Figure A.1 (a) shows the 3D printed part that was designed and figure A.1 (b) shows the back lens placed in the part and all fitted into the housing.

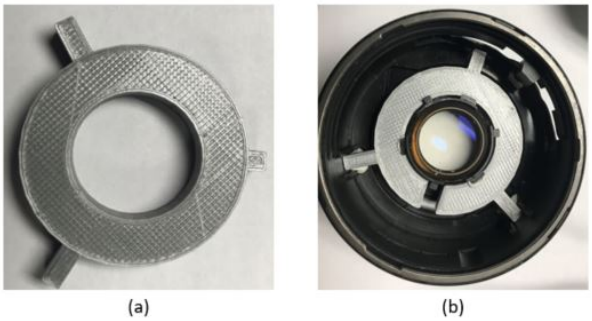


Figure A.1: (a) 3D printed back lens mount. (b) Lens and mount placed within housing.

A.3 Reading the 1951 USAF resolution chart

To be able to use the 1951 USAF resolution chart, one first needs to be able to read the chart. This would aid in determining the meanings of the lines and numbers that are visible. The following explanation was adopted from Cable [45].

The 1951 USAF resolution chart has a series of vertical and horizontal lines that are used to determine an imaging system's resolution. A set of six vertical and horizontal line pairs are in one group, and the resolution chart is made up of 10 groups. Figure A.2 shows elements 2 and 3 of group -2 on a resolution chart.

With line sets in groups of three, these targets offer the advantage of an increased ability to recognize spurious resolution. Spurious resolution is when a group of lines is sufficiently blurred such that the overlap appears to form distinct inverted lines. Spurious resolution results in the appearance of one less line in the set. Since the difference between three lines and two is more easily noticed than the difference between five lines and four, for example, it is easier to recognize the occurrence of spurious resolution in targets with sets of only three lines.

The distance between the lines in each element is the same as the thickness of the line itself. When imaging a target, the resolution of the system can be determined by viewing the clarity of the vertical and horizontal lines. The largest set of non-distinguishable vertical and horizontal lines determines the resolving power of the imaging system. Equation A.1 defines the resolution in line pairs per millimetre based on the group and element number.

$$Resolution = 2^{group+(\frac{element-1}{6})}. \tag{A.1}$$

The USAF 1951 chart used in this dissertation had a maximum resolution of 3.56 line pairs per millimetre.

Table A.1 below lists the number of line pairs per millimetre for a given element within a group based on equation A.1.

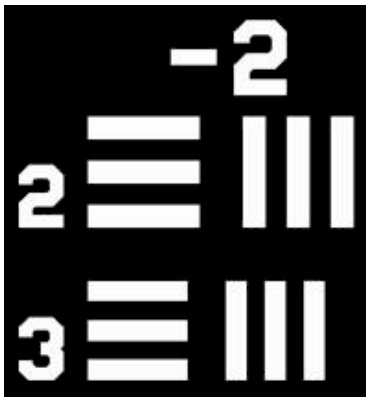


Figure A.2: Portion of the USAF 1951 resolution chart.

Table A.1: Line pairs per millimetre for each group element.

Element	**Group Number**			
	-2	**-1**	**0**	**1**
1	0.250	0.500	1.00	2.00
2	0.280	0.561	1.12	2.24
3	0.315	0.630	1.26	2.52
4	0.353	0.707	1.41	2.83
5	0.397	0.793	1.59	3.17
6	0.445	0.891	1.78	3.56