Reconstruction of 3D Surfaces with Complex Material Composure Using a Light Field Camera

by **Helia Farhood**

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of A/Professor Stuart Perry

University of Technology Sydney

Faculty of Engineering and Information Technology

School of Electrical and Data Engineering

September 2021

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Helia Farhood declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

SIGNATURE: Production Note:
Helia Farhood Signature removed prior to publication.

DATE: 12th Sep, 2021
PLACE: Sydney, Australia

DEDICATION

This PhD thesis is dedicated to my mother Pouran Hamidi
for giving me invaluable educational opportunities
and my husband Mohammad Najafi
for his support, constant encouragement and care.

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my primary supervisor Professor Stuart Perry, for allowing me to work with him and his kind support throughout my PhD at UTS. His valuable advice, continuous encouragement and constant support were critical throughout the development of my research work. He was a wonderful mentor and I never felt stressed before or during any meeting with him. I would like to thank Dr Eva Cheng, my co-supervisor for her guidance and her support. Her deep insights helped me at various stages of my research. I am extremely grateful to my external supervisor Dr Juno Kim for his guidance and useful advice. His important suggestions and advice are priceless. He spent endless time reviewing and proofreading my papers.

I would also like to acknowledge Daniel Heckenberg for his valuable inputs to this work. This research was supported by a scholarship from Animal Logic Pty. Ltd. Dr Juno Kim was supported by an Australian Research Council (ARC) Future Fellowship (FT140100535) and the Sensory Processes Innovation Network (SPINet).

I express my deepest gratitude to my mother for all her support and a heartfelt thanks to my husband Mohammad Najafi and my little son Mahan Najafi for their patience and encouragement.

I also thank all the departmental staff for helping me during these years of my study both academically and administratively.

LIST OF PAPERS (PUBLICATIONS)

Helia Farhood, Stuart Perry, Eva Cheng, Juno Kim, “Enhanced 3D Point Cloud from a Light Field Image," Remote Sensing, vol. 12, p. 1125, 2020. Published in April 2020. Journal Ranking: Q1. (Related to Chapter 6)

Helia Farhood, Stuart Perry, Eva Cheng, Juno Kim, “3D point cloud reconstruction from a single 4D light field image," in Optics, Photonics and Digital Technologies for Imaging Applications VI, 2020, p. 1135313. Published in April 2020. (Related to Chapter 5)

Helia Farhood, Stuart Perry, Eva Cheng, Juno Kim, “Depth estimation: combination of sub-aperture matching and defocusing for a 4D light field”, Elsevier, Pattern Recognition, February 2020 (under review). Journal Ranking: Q1. (Related to Chapter 4)

Helia Farhood, Stuart Perry, Eva Cheng, Juno Kim, “Reflection Recovery in Light Field images with Complex Material Appearance”, has been submitted to Elsevier Journal of Signal Processing Image Communication, September 2020. (Related to Chapter 7)

Helia Farhood, Xiangjian He, Wenjing Jia, Michael Blumenstein, Hanhui Li, “Counting People Based on Linear, Weighted Local random Forest”, The International Conference on Digital Image Computing, DICTA 2017. Published in December 2017.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Representing real world objects on a digital screen is a significant and challenging topic in the area of computer vision and augmented reality. This work addressed the challenge of reconstruction of 3D surfaces with complicated material appearance by using a light field camera. Most recent research uses single images to address this problem, but without using a light field camera, encounter difficulties and limitations to overcome this problem. However, we show that by using a light field camera without user interaction or any requirement for object planarity or symmetry, reconstruction of a 3D model with high accuracy is possible. A light field camera, also known as a Plenoptic camera can capture rich information about the spatial and angular distribution, as well as intensity and colour of light in a single shot. Light field cameras can be used to improve the performance of traditional computer vision problems, such as depth estimation, post-capture refocusing, illumination estimation, and material estimation which are not easy for traditional methods with a standard image. For reconstruction of 3D models, creating a 3D point cloud is essential and is often obtained based on a depth map. As a result, first we developed a robust method to estimate an accurate depth map based on the combination of sub-aperture image matching and defocusing cues for a 4D light field format. The depth map is refined using a fast-weighted median filter providing robustness to noise. Therefore, the proposed approach compared with other state of-the-art algorithms can estimate depth of real-world images and challenging images more accurately. In the second part, we proposed a novel strategy for the creation of a 3D point cloud from the depth map of a single 4D light field image. The proposed method is based on the transformation of point-plane correspondences. Considering the estimated depth map from the previous part, we applied histogram equalization and histogram stretching to enhance the separation between depth planes. The suggested method avoids feature extraction, segmentation and the extraction of occlusion masks required by other methods, and due to this, our method can reliably mitigate noise. In the third step, we improved our suggested method to obtain a dense and more accurate three-dimensional (3D) point cloud. We applied intelligent edge detection by using feature matching and fuzzy logic from the central sub-aperture light field image and the depth map. The results showed that our new method can reliably mitigate noise and had the highest level of detail compared to other existing methods. Finally, having obtained the 3D point cloud, we handled the problem of reflectance in complex material appearance. We developed a new strategy to recover reflectance information based on colour analysis as well as brightness analysis of a light field image. Our aim is to separate specular pixels by using two different strategies. On the one hand, we estimated light source colour from different angles of the RGB light field image to find specular pixels. On the other hand, we binarized light field sub-aperture images to obtain brightness areas in different viewpoints and extract specular highlights. Experimental results demonstrate the effectiveness of our method in both synthetic and real-world images compared to other state of the art methods. Overall, 3D reconstruction can cover many applications and solve many problems of computer graphics and computer vision that is still a challenging topic.

Chapter 1: Introduction

The reconstruction of an accurate 3D model from a real-world scene is an important challenge in the field of computer vision, augmented reality, 3D scanning, rendering and distance measurements. The aim of this work is to overcome the problem of reconstructing 3D surfaces with complicated material appearance using Light Field (LF) cameras. Generally, there are two approaches to 3D reconstruction, one of them is based on multiple-images and another relies on a single-image. The first approach requires the capture of a set of images. An important aspect of this process is the relationship between multiple views that convey the information about the structure of corresponding sets of points and this structure is related to the pose and calibration of the camera. The first approach has many issues such as errors in the position of the camera in each capture and the distance between the centres of projection and radial distortion. The second approach is based on a single image capture. The purpose of this approach is to create a 3D model using a single image. Creating 3D point clouds from a single capture has attracted attention from the research community and most recent research has focused on using single-images. However, 3D reconstruction from a single image still has many issues and is a challenging topic [1]. Many attempts have been made to handle the problem of 3D reconstruction from a single image, however, when those methods are applied to real-world images, they generally suffer from a sparsity of information. Since in the one capture there will be only a single view of the scene, as a result some parts of the scene will not be sufficiently visible for accurate 3D reconstruction. This problem causes the generation of 3D point clouds to be problematic for single capture techniques. Moreover, work exists attempting to train an auto-encoder to learn a mapping from 2D input images to 3D point clouds. But this type of estimate is not very accurate and requires extensive training. There are some other limitations such as reliance on user interaction and costly global optimization processes for creating 3D point clouds based on a single projection. However, we demonstrate that it is possible to reconstruct 3D models from single images with high accuracy by using a light field camera without user interaction or any requirement for object planarity or symmetry.

In recent years' research has focused on algorithms for the reconstruction of a 3D model from a single light field image also known as a Plenoptic image. A light field camera with one image capture through an array of micro-lenses, can collect a wide variety of information about the colour, intensity and direction of light in a scene. This means that, instead of utilising two or more single cameras to increase the number of viewpoints, light-field cameras provide many viewpoints of a scene with a single snapshot and each such view is termed a "sub-aperture image". In contrast, in a traditional digital camera, the lack of data about the directional intensity of light makes solving many problems of remote sensing and computer vision difficult.

The concept of light fields has been extensively adopted for several areas. One significant application of light field is for the creation of 3D models. A light field image describes the distribution of light rays in free space which allows for the capture of richer information from our world. To reconstruct a 3D model from light field data, it is essential to create a 3D point cloud based on a depth map. As a result, the reconstruction of the 3D object requires an accurate estimation of the scene's depth. Therefore, in this work, we first focused on developing an accurate depth map estimation, followed by a novel strategy for the creation of 3D point clouds based on the obtained depth map and finally we show how we can handle the problem of complex reflectance in light field images.

From an industry viewpoint, the global 3D reconstruction technology has seen very rapid growth. 3D reconstruction has a critical role in industrial manufacturing. Today, the number of manufacturing pipelines based on 3D-printing has been increasing such as 3D-printed shoes, 3D-printed jewelry, and even 3D-printed vehicles [1]. The number of autonomous vehicles is also increasing and producers are adopting this technology with much eagerness to enable 3D depth measurement. One of the main applications of 3D reconstruction is in the medical and healthcare fields. Medical technologies mostly are costly when they come into the market, however, numerous new 3D-printed solutions can be obtained at a reasonable price point. For instance, specialists have developed 3D-printed skin for burn victims or in cancer treatment, and 3D printing is making huge leaps forward. Several 3D-printed medical solutions are still at a prototype or pilot-study level, but initial tests are looking promising in a wide range of areas. There are many other benefits in the field of 3D bioprinting, and many of them have been a part of successful surgeries and treatments.

Central to the reconstruction of a 3D scene from light field data, is the creation of an accurate 3D point cloud based on the intermediate step of the creation of a depth map. As a result, the reconstruction of the 3D scene requires an accurate estimation of the scene's depth. Therefore, in this work, after a comprehensive description of light field technology and a detailed literature review (chapters 2 and 3), we first focus on developing an accurate depth map estimation algorithm in chapter 4. In chapter 4 we described our proposed method for estimating the depth map including handling the problem of occluded areas. The second important step to have a complete 3D model is converting the depth map to a 3D point cloud, which we discuss in detail in chapters 5 and 6. In these two chapters (5 and 6), we introduced a novel strategy for the creation of 3D point clouds based on the obtained depth map from previous chapter. As one of the most challenging aspects of reconstruction of 3D model is the handling of the problem of reflection within the scene and we consider this issue as a final step. Then by considering the result of chapters 4, 5 and 6, we move to focus on the problem of reflectance in following chapter. Therefore, in chapter 7 we show how we can handle the problem of complex reflectance in light field images and how this leads to accurate 3D scene estimation from light field images.

1.1 Depth Map Estimation

Recovering information about the 3D shape of surfaces from images is a major challenge for vision, especially in natural scenes where there are additional surface attributes that we can infer from the same image structure (e.g., gloss, colour, opacity). Computer vision is an essential enabler for several applications relied on in various domains ranging from the industrial to the domestic. Recent years have seen increasing applications for 3D computer vision systems including, virtual and augmented reality for gaming and education as well as 3D manufacturing, and 3D capture for commercial, industrial and cultural heritage applications. 3D reconstruction is central to these applications, and accurate depth map estimation is an essential part of 3D reconstruction [2]. Accurate depth information can also be used in critical real-world applications such as medical photography, robot-assisted surgery, agriculture, intelligent driver assistance systems, better rendering of 3D scenes, creating 3D maps, automatic 2D-to-3D conversion in film, and robotics [4,5]. Moreover, when depth information is combined with traditional applications involving 2D imagery, opportunities arise in various areas, such as Human Computer Interaction (HCI), human body posture analysis and gesture recognition. One approach to estimating 3D depth maps is to use a light field camera, a type of Plenoptic camera that captures rich information about the intensity, colour and direction of light. The specific structure of the light field expands the disparity space to a continuous space making depth estimation more robust and precise [3]. In this work we observed that using the 4D data representation of a light field is more reliable and can help achieve a superior estimate of the depth map.

To date, considerable research has been undertaken to solve some problems of depth map estimation based on LF data and various methods have been developed relying on different representations of LF data [4].

Approaches have used multiple cues such as defocus and correspondence [5] for depth map estimation. However, because their methods depend on defocusing areas by shearing the light field, these methods have poor depth estimation for the parts of an image where objects are far from the main lens of the focal plane [5]. Furthermore, the problem is compounded by the fact that they do not consider occlusion pixels before computing the final depth map which causes inaccuracies in the depth map for images with occluded areas [6]. However, there are still many key challenges in the use of light field for estimating depth maps, particularly for real-world images [2]. In this work we combine the sub-aperture image matching and defocusing cues on images using an occlusion aware method. This image combination was achieved by first using edge detection and edge orientation predictors to determine candidate occluded pixels and then applying our framework to the image, taking into account the candidate occluded pixels.

To further improve the depth map quality, we performed hole-filling and filtered the depth map using a fast weighted median filter. Our results are superior to previous works that used an unweighted median filter [7]. In addition, we used a joint histogram with median tracking that reduces the computational complexity [8] of the weighted median filter. This method provides fast access to data which can reduce the processing time when removing noise.

Our main contributions in depth map estimation (Chapter 4) are:

- By a comparison of light field data representations, we show evidence that using a 4D LF format leads to having a more accurate depth map compared to the lenslet representation and can help overcome the problem of noisy depth map estimates in low texture areas of LF images.

- Furthermore, we develop an occlusion-aware framework for combining sub-aperture image matching and defocusing cues, which results in both a reduction in the image search time, and more accurate results for real images.
- We also apply a fast and robust weighted median filter for noise removal on the initial depth map. We use the joint histogram to compute weights and then a median tracking strategy to obtain the final weighted median filter. The overview of the method for our proposed depth map estimation is shown in Figure 1.

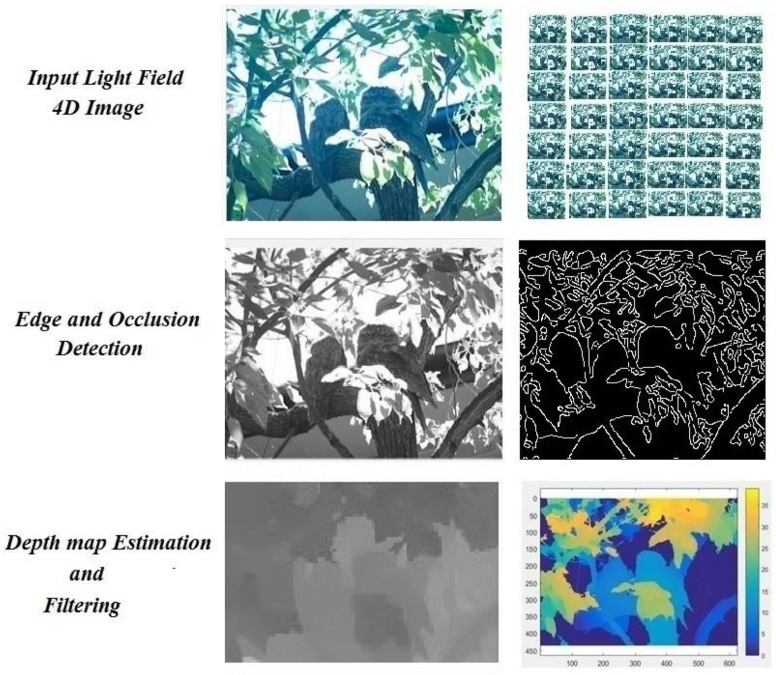


Figure 1. Overview of the proposed method on a real Light field image captured with a Lytro Illum camera. We used the 4D format of a light field image for the input images. This was followed by edge detection for occlusion identification. Then, initial depth map estimation using correspondence matching and de-focus cues, and finally, combination, filtering, normalization, and final depth estimation.

1.2 Creating the 3D Point Cloud

The issue of acquiring noise-less and complete 3D point clouds is of paramount importance to support advances in virtual, augmented reality and 3D printing. There is a significant demand for obtaining high quality 3D point clouds scanned from real objects for 3D rendering, computer graphics, 2D view extraction from 3D data, virtual reality, and object deformation [9]. A 3D point cloud can be obtained from various methods. Many existing methodologies for the reconstruction of 3D models are based on either Structure-From-Motion (SFM) or Dense Multi-View 3D Reconstruction (DMVR). However, these methods need multiple captures from different angles and significant user interaction when using an ordinary camera. For this reason, recent research has focused on the development of new strategies using less costly devices while continuing to minimize complexity [10].

Given the current state-of-the-art, one very effective solution is to create 3D models from a single image. A single conventional 2D image has limited information about the 3D nature of objects or scenes, so accurate estimation of 3D geometry from a single image is difficult to achieve [11]. Nevertheless, light field images with the ability to capture rich 3D information with a single capture offer an effective solution to this problem. The introduction of light field cameras (a type of plenoptic camera) has helped to reveal new solutions and insights into a wide variety of applications, including traditional computer vision and, image processing problems. Light field cameras capture rich information about the intensity, colour and direction of light, and can be used to estimate a depth map or 3D point cloud from a single captured frame. Light field camera technology has the potential to create 3D point cloud reconstructions in circumstances where standard multi-capture techniques can fail such as dynamic scenes or objects with complicated material appearance. The unique features of light field images – e.g., the capture of light rays from multiple directions – provides the ability to reconstruct 2D images at different focal planes. This feature can also aid in the reconstruction of scene depth maps [10]. A light field image can have multiple representations, but two LF representation formats are more common for computer vision and image processing problems; the lenslet format and the 4D LF format. In the work we report here, we used the 4D LF format as described below. For the reconstruction of a 3D point cloud of an object, we developed a new method which is based on the transformation of the point-plane correspondences. The input of our system is a 4D LF image which can be captured by light field cameras, such as the Lytro Illum [12] or created synthetically. The depth map of the LF can be produced by software bundled with the Lytro camera or similar third-party software or using our proposed depth map estimation method.

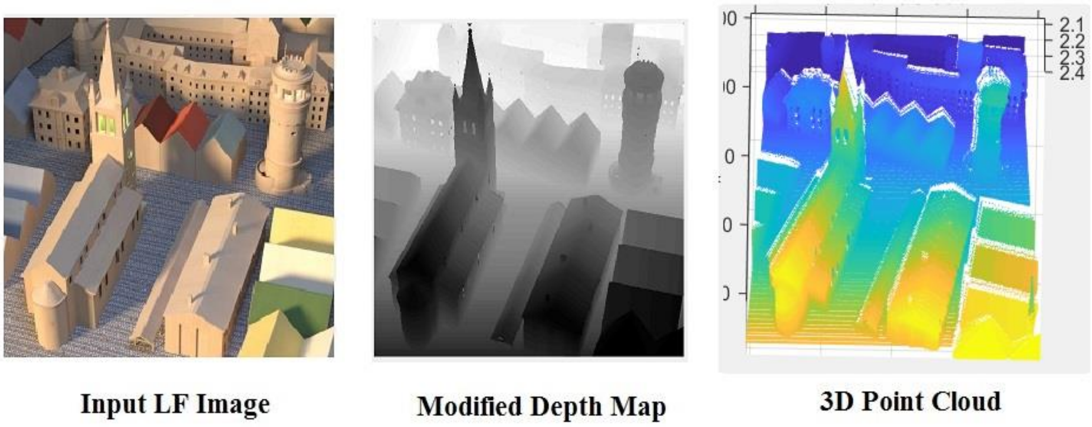


Figure 2. Using a light field image for creating a 3D point cloud.

Estimating a densely sampled depth map is essential for creating a 3D point cloud in this context. We enhance the depth map of the input image substantially by applying histogram equalization and histogram stretching followed by intelligently adding edge detection information determined from feature matching and fuzzy logic techniques.

Compared to adding linearly the ordinary edge detection information such as that obtained by Canny and Sobel filters, our approach obtains a more accurate depth map result. For this purpose, we calculate the gradient of the equalized depth map in the *x* and *y* axis separately by comparing the intensity of neighbouring pixels and then, by using a fuzzy logic method, we define which pixels belong to the edge of a region of uniform intensity. This kind of edge analysis can improve depth map estimation by introducing colour mapping information. In parallel, we detect and extract SURF features from Canny edge detection performed on the sub-aperture images of the LF and the original depth map. Then we match their features (matching features between the edge of the original depth map and the sub-aperture image) and add the matched pixels to the result of the fuzzy logic edge detection. Based on this process we can obtain edges of the depth map and sub-aperture images more adaptively without introducing noise caused by ordinary edge detection. Then, we combine this edge information with the equalized depth map by considering the level of the reference image (depth map). This kind of enhancement after estimation of depth map is one of the significant contributions of this work. We then transform the point-plane correspondence to acquire a 3D point cloud. Figure 2 shows the production of a 3D point cloud from a LF image

Our main contributions in the 3D point cloud part (Chapters 5 and 6) are:

- We acquire a 3D point cloud based on a single image from a light field camera which provides valuable information on the 3D structure of the scene.
- We design a novel approach for enhancing depth maps based on feature matching and fuzzy logic, which can overcome the common problem of introducing noise.
- We develop a method for converting an enhanced depth map to a 3D point cloud.

1.3 Reflectance Recovery

For navigating a physical environment and interacting with objects around the environment, individuals have the visual ability to perceive the complete shape of objects. However giving this ability to a computer is a challenging task. In computer vision and computer graphics, improving surface reflectance estimation has drawn considerable attention among researchers because of its relevance to the understanding of geometry and material properties of objects in a scene.

Recovery of reflectance information is a challenging problem due to the difficulty of modelling real-world material reflectance mathematically, and is in practice approximated by several different models for different types of material [13].

The visual perception of real-world materials depends on the illuminating environment as well as the specific viewpoint. Real world objects are complex and can be represented by complicated concepts such as glossiness, Lambertian aspects, specular and diffuse maps [14]. So, multiple observations are needed to enable us to fully specify the appearance of an object. Recently some methods have been introduced with regard to recovering the reflectance of Lambertian surfaces based on a single-view geometry. However, estimating complex material appearance based on a single view is difficult and cannot follow these methodologies. So recovering the 3D structure of objects with complex materials such as glossy surfaces, ceramics, metals and the skin of fruit or leaves is still a difficult problem [15]. In the chapter 7 we present a robust and efficient algorithm to estimate surface reflectance properties and remove the specular components of pixels of real-world objects using a light field camera. Using light-field cameras enables us to handle problem of reflectance by with a single capture with minimal effort. Our proposed method can be a practical and appropriate alternative technique to traditional multi-view methods. The aim of our theory is to detect specular pixels and specular highlights and remove them from the original light field image. One popular approach for handling the reflectance surface is separating specular pixels. Specular reflections are often related to bright pixels that can be perceived in captured images. Such reflection is informative in the terms of object reflectance and scene illumination. In fact, specular pixels indicate useful cues with regard to the light source direction [16]. In this way, we focused on finding specular pixels and highlights by using the advantages of light-field data to estimate and remove the specular parts. For finding the specular components more accurately we utilized two different approaches and each approach produces an estimate of the reflectance of the surface and provides data regarding the specular components of the image. Finally, we combine these methods and separate specular pixels from the original light field image. We consider both RGB and binarized light field images. By analysing the colour of pixels in the RGB light field image as well as analysing the arrangement of pixels of the light field binarized image we can estimate the type of appearance. The advantages of using a light field camera is that, it can provide us multiple views of a single point, which enables us to estimate the light source over the whole object [17]. Our first approach is based on the estimating light source colour and because specular pixels have different behaviour compared to diffuse pixels in different viewpoints of the light field image, we can estimate the specular pixels by rearranging the light-field data. The difference of behaviour of specular and diffuse pixels is based on the level of the colour consistency which in the case of specular pixels can be changed substantially compared to diffuse pixels. The second approach is based on an image binarization technique. This means, binarizing the light field data by thresholding on the pixel intensity of grayscale image. This approach gives us specular highlights of the image taking advantage of the fact that specular highlights in an image generally have the highest intensity compared to the whole image. We used Bernsen's local thresholding technique taking into account higher intensities to obtain specular highlights in the image. This information collected from different viewpoints of light field data enables us to find bright areas more accurately. This kind of detection for specular highlights can help recover reflectance information for many types of complicated material appearance. At the end we combine this data linearly to obtain a specular image estimate and by removing this information from an original image, create a specular-free light field image. Figure 3 shows a sample of our approach for finding specular component pixels on the skin of fruit and Figure 4 shows a block diagram of our approach.

Our main contributions to the estimation and extraction of specular components in light field images (Chapter 7) are:

- We develop a technique for extracting specular reflectance information using a single image capture from a light field camera as an alternative method to traditional multi-view techniques.
- We combined two different approaches based on the analysis of colour and brightness in binarized sub-aperture images which can robustly estimate and remove specular components.

The developed method can cover a wider variety of complicated material appearance compared to other methods.

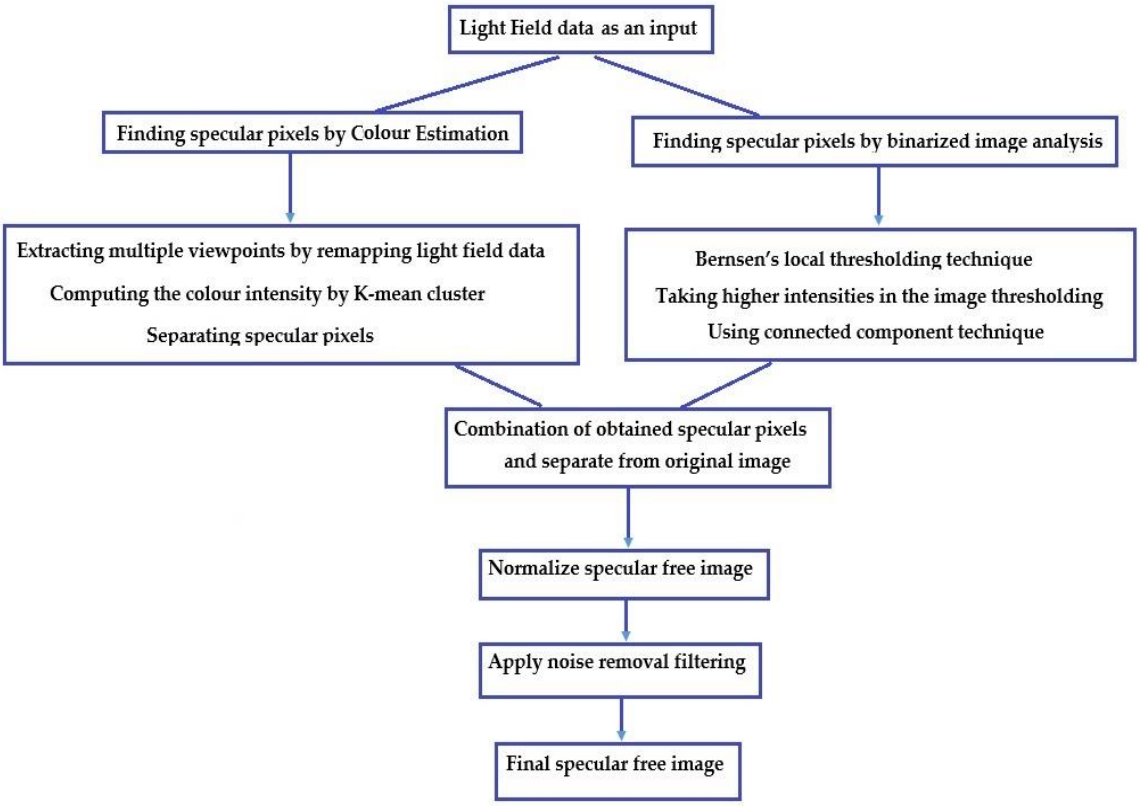

Light Field data as an input
Finding specular pixels by Colour Estimation
Finding specular pixels by binarized image analysis
Extracting multiple viewpoints by remapping light field data
Computing the colour intensity by K-mean cluster
Separating specular pixels
Bernsen's local thresholding technique
Taking higher intensities in the image thresholding
Using connected component technique
Combination of obtained specular pixels
and separate from original image
Normalize specular free image
Apply noise removal filtering
Final specular free image

Figure 3. Block diagram of our approach using two different techniques for finding specular components.

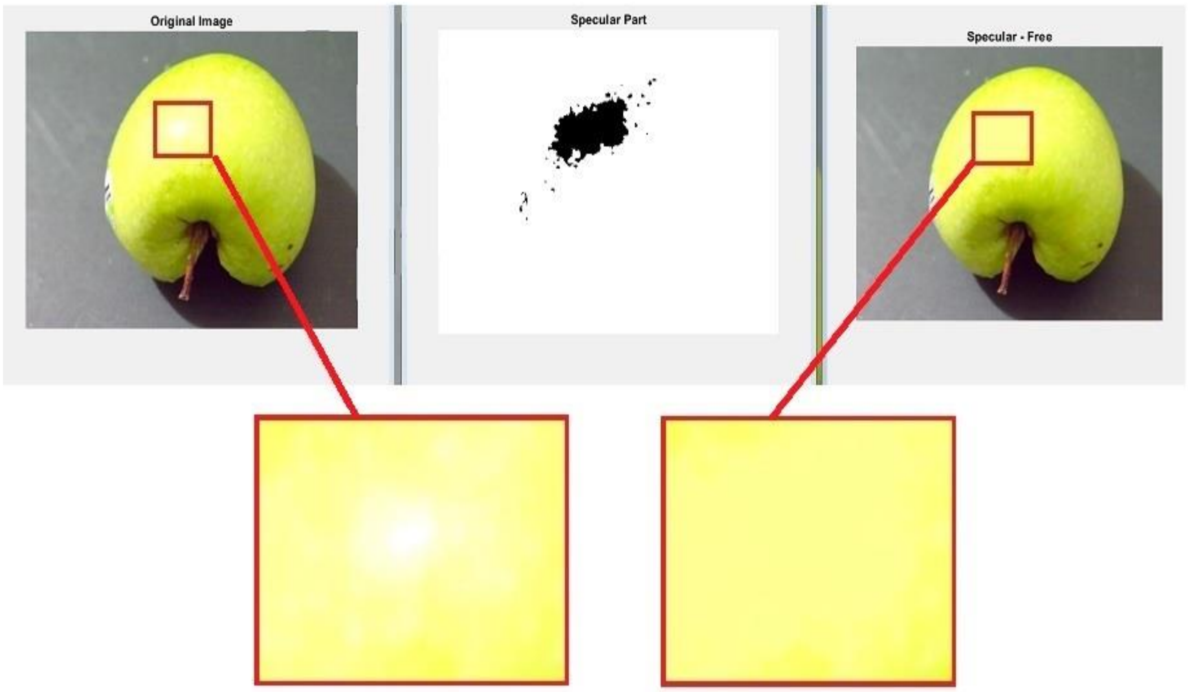


Figure 4. A sample of our proposed method applied to a real light field image captured by a Lytro camera.

Chapter 2: Light-Field Background

A light field camera is a type of Plenoptic camera that captures rich information about the intensity, colour and direction of light. These cameras can be used to improve the performance approaches to traditional computer vision problems, including depth estimation, post-capture refocusing, illumination estimation and material estimation. These problems are not easy to solve for traditional methods that capture standard 2D images.

2.1 Light Field Definition

As it is clear from the name, a light field is the visualization of the implication of considering light as a vector field. It defines a formal explanation of the intensities of a ray's collection flowing from and into each point in space [18]. From one viewpoint, a light field is a method of capturing a scene from multiple viewpoints which can be illustrated as a collection of views representing the same region of interest. Another way of considering a light field image is as a set of captures recorded by a moving camera, where photos were taken at different viewpoints or even two views of a stereo camera however, generally, the number of views is more than two. Comparing this to a standard image, a light field provides a great deal of additional information about the geometry of the scene which can be used for further processing. Probably, the most significant feature of light field cameras is to directly supply different viewpoints of the same scene. This feature allows a viewer to determine out depth and structure within a scene [18]. The distance between viewpoints and transitions between them can cause some problems, for instance when the distance between viewpoints is large. In this case, some information about the geometry of the scene needs to be leveraged for an accurate 3D reconstruction. When the position of multiple viewpoints is clear, the creation of a 3D model can be easily achieved. These features also help us estimate an accurate depth map based on the position and number of views. This depth map can be used for generating a 3D point cloud.

Having an accurate depth estimation from passive sensors is highly relevant to the 3D movie industry, as this reduces manual intervention. For some objects with complicated material appearance such as glossy objects, acquiring an accurate depth map is difficult to obtain with a setup of stereo or active depth sensors. This kind of geometry information can be applied for view combination, as well, with a hard or soft proxy which can be of benefit to Virtual Reality (VR) displays. Besides considering the benefit of having access to several views, light fields also have many other features such as known coordinates of light field rays that make it possible to repeat the ray integration procedure done by conventional cameras, post-capture with various parameter settings. This feature allows 2D image rendering with different depth and focus planes from light fields. During the past few years, light field cameras have become increasingly available, for both research and industry usage. An example of a commercial light field camera, the Lytro ILLUM, is shown in Figure 5. There are several light-field cameras on the market, including Pelican, Raytrix, and Lytro. However, the most well-known type for consumers is the Lytro camera. In this thesis we used the Lytro ILLUM camera for all our experimental results. An early light field camera was developed in 2005 and was called the Lytro [12], and a prosumer version of the Lytro was introduced in 2014 (called the Lytro Illum). Light field cameras have several features such as post-capture refocusing, depth map estimation and illumination estimation.

One of the main concerns in photography is the problem of focus. The clearest difficulty is the burden of focusing accurately on the object before capture [19]. A poorly focused photograph extracts a general sense of loss as the focus cannot be changed after taking a picture and focusing accurately is not easy. One of the best solutions to the focus problem is to leverage the abundance of a digital image sensor resolution to sample every single ray of light that provides the final image. This “super-sampling” of the light entering a digital camera offers a great deal of flexibility and quality in processing the final output images. This is defined as a “light field” in computer vision and computer graphics. A micro lens array in front of the photo sensor can act to capture the light field inside the light field camera. Each microlens can cover a tiny array of photo sensor pixels and also splits the light that impacts a small section of this array according to the angle of incidence. The section of a sensor array under a microlens can be considered as a macro-element of the scene where each value of the corresponding photosensor pixel can be considered as one of a range of light rays that emerge from the macro-element, each at a different angle. Ray-tracing techniques can then be used to render the final 2D images from the recorded light field [20].

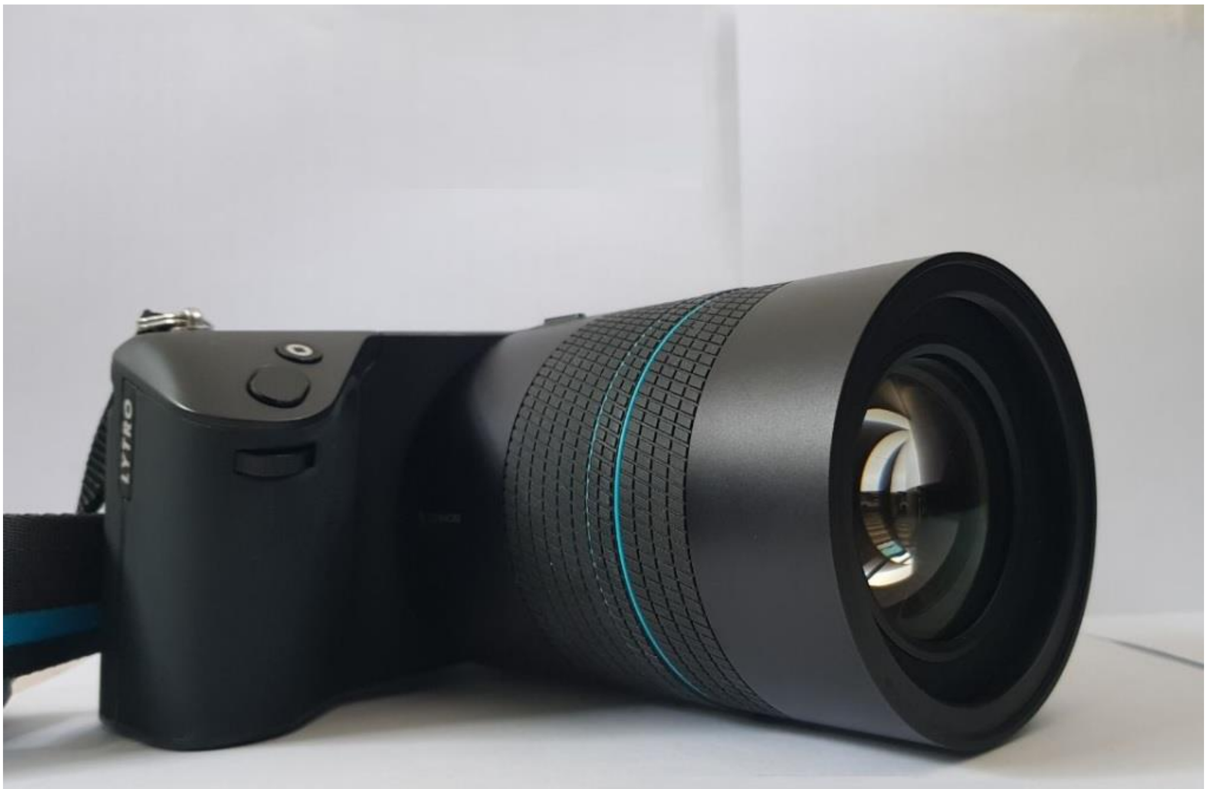


Figure 5. Lytro ILLUM Plenoptic Camera. In this thesis we used this Lytro ILLUM camera for some parts of our experimental results.

The aim of this configuration is to trace the recorded light rays via its optics to the plane of imaging is such a way as to generate a favourable photograph. In many implementations of light field camera technology, light passes through the main lens and then through an array of micro lenses before being detected by a sensor.

This kind of ray-tracing can contribute as a system to handle the unpleasant non-convergence of rays, which is an important part of the problem of focus. This feature provides the capability of refocusing an image after capture. Tracing the recorded light rays can produce images focused with different depths of field as well.

2.2 The Plenoptic Function

A light field can be defined as a vector function that indicates the amount of light flowing in every direction through every point in space. In comparison to conventional cameras, a light field camera captures not only a 2D image, but also information about how light intensity varies with direction of the incoming light rays at each pixel [21]. The LF function can be parameterized by the intersection of light rays with two arbitrarily placed parallel planes – a lens plane approximately located at the position of the camera lens and a sensor plane approximately located at the position of the sensor, as shown in Figure 6 and 7. The system of coordinates for each pixel in the light field image (u, v, s, t) is defined with (u, v) denoting the position on the lens plane and (s, t) denoting the position on the sensor plane. An aligned light ray is determined in a system when it first crosses the *uv* plane (lens plane) at coordinate (u, v) and then crosses the *st* plane (sensor plane) at coordinate (s, t), and can be encoded by a function $L(u, v, s, t)$ [22]. Each position on the sensor plane can be modelled as a pinhole camera viewing the scene from a position s,t on the sensor plane.

Each such view is termed a "sub-aperture image". Thus a 4D LF, $L(u, v, s, t)$ can be defined as a set of views (sub-aperture images) captured by a light field camera determined by the two parallel planes, *st* and *uv*, where (s,t) indexes the array of sub-aperture images and (u,v) indexes the pixels within each sub-aperture image as shown in Figure 6.

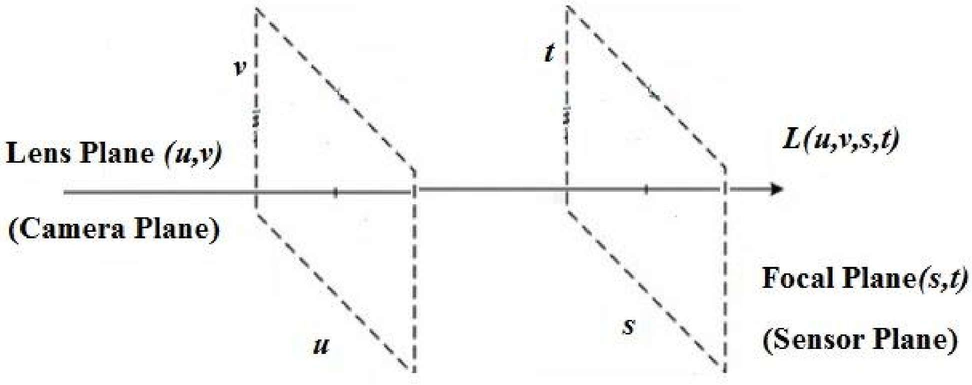


Figure 6. In a common parameterisation of light field images, light passes through the Lens plane (also called the Camera plane or Viewpoint or Angular plane) and then is captured at the Sensor Plane (also called the Focal plane, Image plane or Spatial plane). Each ray of light can be considered to pass along a line that enters the Lens plane at position (u,v) and ends at the Sensor plane at position (s,t).

2.3 Different Representations of the Light Field Data

Data from a light field camera may have multiple representations. However, two LF representation formats are common for image processing problems, these are the lenslet format and 4D LF format.

Lenslet format – Data in the lenslet format is the raw sensor data from lenslet-based light field cameras such as the Lytro Illum. Consider a sampling of the lens plane and sensor plane such that the lens plane is described by a grid of *Nu* by *Nv* samples and the sensor plane is described a grid of *Ns* by *Nt* samples. In the lenslet format, the 4D light field is mapped to a single 2D plane consisting of *Nu* by *Nv* grid of lenslet images. Each lenslet image consists of *Ns* by *Nt* pixels on the lens plane. An example is shown in Figure 7.

The 4D format $L(u, v, s, t)$ - Each camera accumulates the light rays departing the *uv* plane and entering at a specific point on the *st* plane (the gathering of light rays from a specific viewpoint). As a result, the 4D LF can be defined as a 2D array of images. The 2D slice

$I_{s*,t*}(u, v)$ (sub-aperture image) can be obtained by collecting the samples at a fixed coordinate (s^*, t^*) on the *st* plane. The slice $I_{s*,t*}(u, v)$ can be considered as a 2D snapshot captured by a pinhole camera located at (s^*, t^*) [22]. In practice, the 4D format is constructed from the uncompressed lenslet image by using information on lenslet positions usually stored in camera metadata to extract the sub-aperture images and re-arrange them into the 4D LF structure.

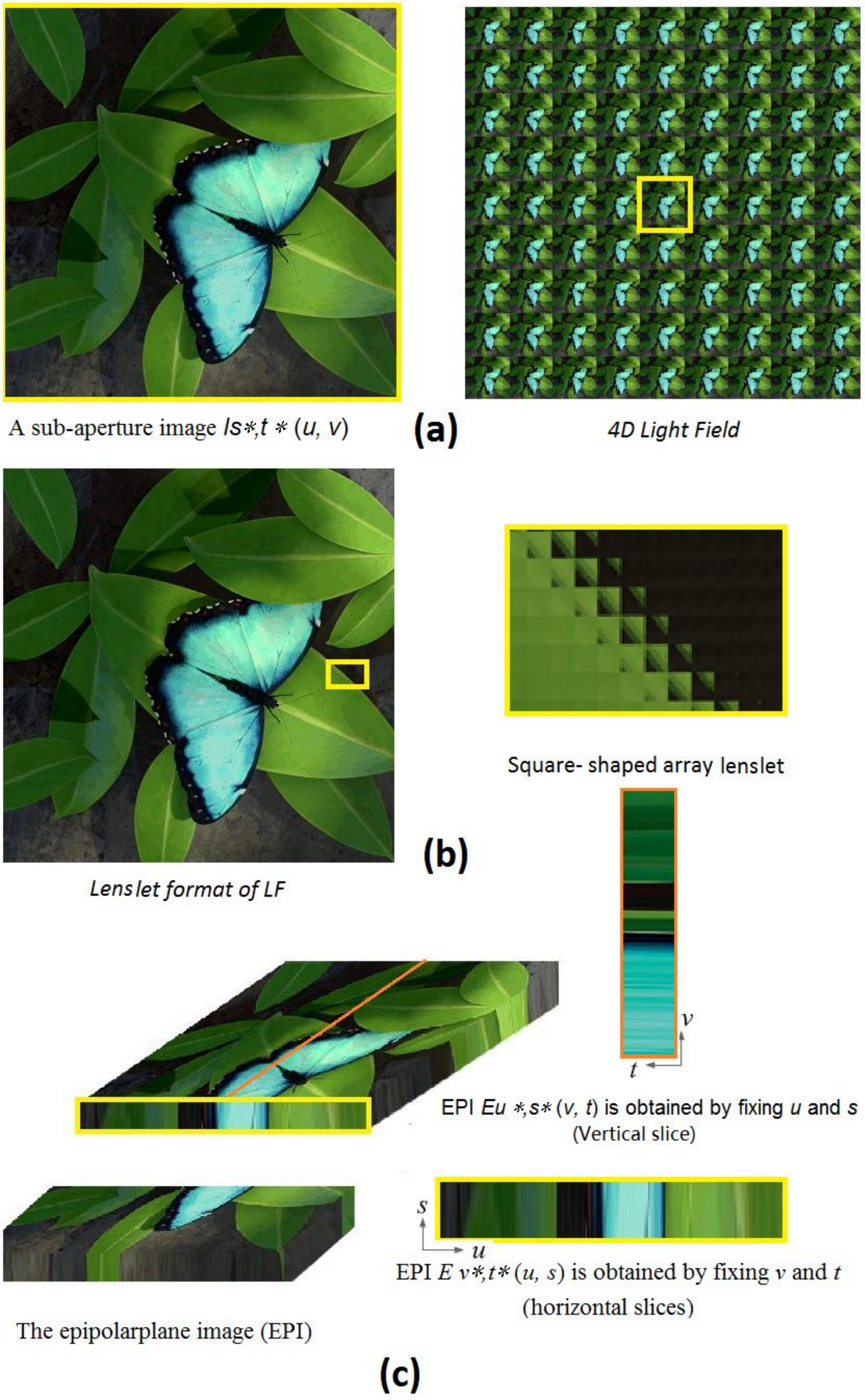

A sub-aperture image Is∗,t ∗ (u, v)
(a)
4D Light Field
Square- shaped array lenslet
(b)
Lenslet format of LF
v
t
EPI Eu ∗,s∗ (v, t) is obtained by fixing u and s
(Vertical slice)
s
u
EPI E v∗,t∗ (u, s) is obtained by fixing v and t
(horizontal slices)
The epipolarplane image (EPI)
(c)

Figure 7. Different formats for representations of light field images. (a) A single sub-aperture image from the 4D format of a light field image, I_(s,t*) (u,v), obtained by collecting the light field samples with fixed st coordinates s∗ and t∗. (b) A lenslet format light field image in which a LF sub-view I_(u*,v*) (s,t) is obtained by collecting the samples with fixed uv coordinates (u∗ and v∗). (c) An epipolar plane image (EPI) acquired by fixing the coordinates in both the angular and spatial dimension. For instance, the horizontal slice EPI E_(v*,t*) (u,s) is acquired by fixing v and t, and the vertical EPI E_(u*,s*) (v,t) is acquired by fixing u and s.*

The light field data can be visualised in a number of ways. It can be viewed as a series of sub-aperture images or as Epipolar Plane Images (EPI). The EPI include information across both uv and st planes by collecting the light field instances with a fixed spatial coordinate u and coordinate s (or v and t) into an alternate 2D image $E_{u*,s*}(v,t)$ (or $E_{v*,t*}(u,s)$ for example).

Some light field depth estimation methods use the lenslet format as an input which is a more compressed version of the LF data. These methods require additional camera metadata and image processing steps prior to depth map estimation [23]. A key difficulty in working with this format is that lenslet images can have a variety of different shapes of microlens arrays such as circular, triangular or diamond shaped arrays, and this variety makes it difficult to design a single algorithm to compute an accurate depth map. In contrast, the 4D format contains additional redundancy, but is more readily processed and will be shown to simplify the estimation of the depth maps. The 4D LF format essentially includes easily accessible multiple views of the scene, making depth map estimation possible.

2.4 Applications of Light Field Cameras

Light field cameras have several applications. The main applications can be categorized into three main groups. First, the application of geometry estimation that includes post-capture refocusing, depth map estimation and 3D point cloud estimation. The second group of main applications includes image rendering and the last important application is classifying material appearance [24].

2.4.1. Geometry Estimation

In the field of environmental research, there are a wide variety of techniques to obtain 3D information about real-world entities like buildings and plants. Most existing techniques need a high level of methodological complexity, but with the emergence of light field cameras, many problems have now been addressed in area of 3D modelling, measurement and monitoring. In the field of remote sensing, light field cameras are also suitable for generating 3D information for different kinds of monitoring applications, such as the monitoring of plant growth [25].

One of the significant features of light field cameras is illustrated in Figure 8 which shows post-capture refocusing in two different focal planes. Refocusing allows for changing the focal plane to a different position post-capture. This feature has a significant role in the generation of a depth map from a LF image. A light field can be considered as a vector function $I(u, v, s, t)$ between two planes (lens plane and sensor plane) [22] where u and v are coordinates on the lens plane and s and t are co-ordinates on the sensor plane as shown in Figure 8.

An aligned light ray is determined in a system when it first crosses the *uv* plane (lens plane) at coordinate (u, v) and then crosses the *st* plane (sensor plane) at coordinate (s, t), and can be encoded by the function $I(u, v, s, t)$ [22]. Each position on the sensor plane can be modelled as a pinhole camera viewing the scene from a position s, t on the sensor plane.

Depth estimation from two views using stereo cameras is one of the well-studied problems in the field of computer graphics and computer vision. By using a light field camera instead of utilizing a multi-image correlation-based method, estimation of the depth map is possible with a single snap-shot. The occlusion problem is one of the main concerns for the estimation of depth maps based on light field images, and in this thesis we overcome this problem by using an edge orientation predictor. Moreover, the difficulty of ill-posed depth map estimation problems in classical stereo becomes much easier to handle when utilizing a light field camera.

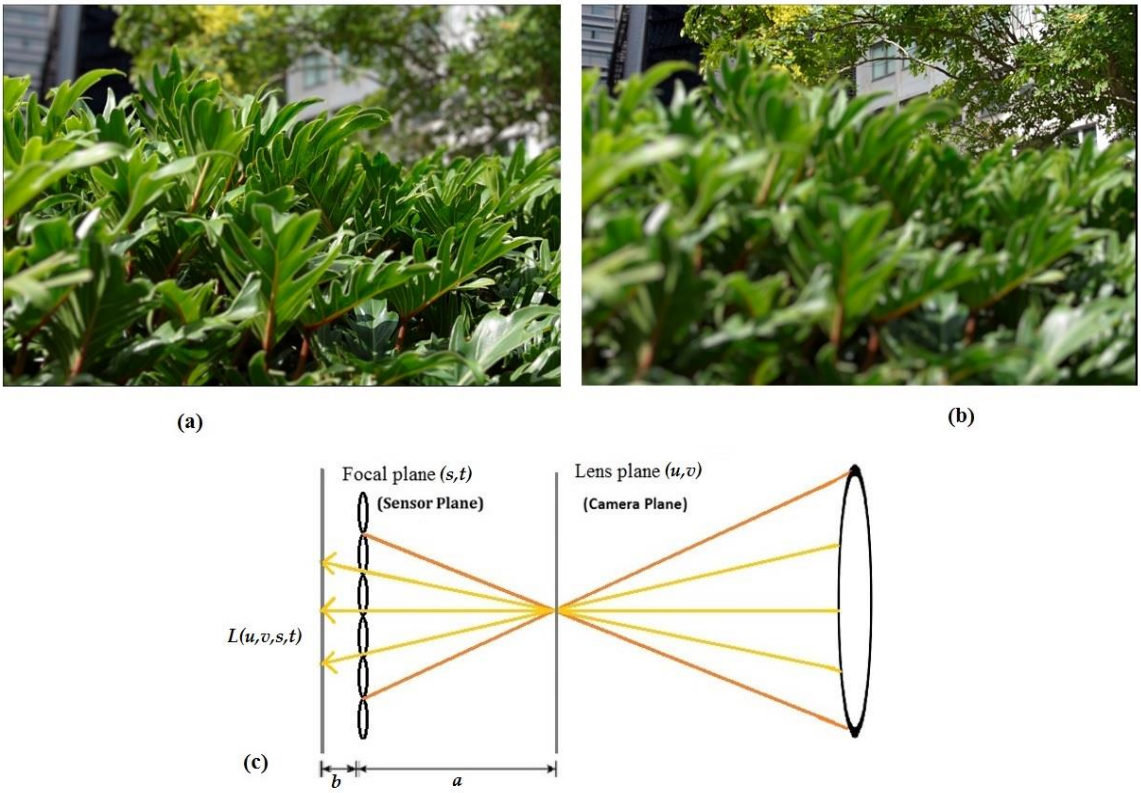


Figure 8. An illustration of the light field function and post-capture re-focusing. (a) Focus in the foreground. (b) Focus in the background. (c) A diagram of a light field camera setup where a is the distance from the camera plane to sensor plane and b is the distance between the sensor and the micro-lens array.

Considering sub-aperture matching of point correspondences and defocusing of the light field can aid in the creation of a depth map and based on the obtained depth map, creation of a 3D point cloud can be possible as will be shown in subsequent chapters.

2.4.2. Image Rendering

Generally, light field photography is related to the field of image-based rendering which focuses on creating images from computational methods as opposed to direct sampling. Creating new views after capture by straight incorporation can be possible without processing any sort of distinct proxy of geometry identified by a light field [18]. In contrast, the estimation of the depth map is based on the processing of corresponding rays of the obtained views and re-projections. The light field camera can capture light rays that are sorted by angle. This feature can allow replication of ray angular combinations produced by a number of different lens structures post-capture.

This means that, new images can be rendered at different planes of focus and aperture settings from the single light field capture [26, 27].

2.4.3. Classifying Material Appearance

The information about the angular distribution of the light field and defocusing can also be used for some other applications outside of depth map estimation and image rendering. For example, one significant application for light field in the field of computer vision is the identification of material appearance. Considering the densely sampled angular information of light fields, the difference between angular samples at a certain location can be exploited to classify material appearance. In the area of material appearance, non-Lambertian surfaces, such as glossy objects, provide changing intensity in the angular dimension which can be used for object extraction and detection [28, 29].

Furthermore, the advantages of using a light field camera are that it can provide us with multiple views of a single point. This enables us to estimate the light source over the whole object. This attribute can be useful for solving the problem of reflectance in a light field image.

Light fields can process a soft depth measure as well that, when combining other features of light field, makes overcoming the problem of disambiguation of scenes possible.

Considering the sharpness variation at different depth planes, it becomes possible to process a complex measurement of the depth map to help specify the salience of objects in a scene.

This can be especially useful for some situations in the image that colour is not sufficient to differentiate objects. For computing embedding information of the depth, the combination of gradients of angular intensity with gradients of spatial intensity is used. This also can be useful for the classification of, and differentiation between, 3D objects.

Chapter 3: Literature Review

There are several approaches available in the literature addressing the problem of the reconstruction of 3D surfaces with different methodologies. In this chapter, we first review related work on general concepts of 3D reconstruction. Then, we review literature on the specific steps for handling the problems of 3D model reconstruction. As we discussed in Chapter 1, for the reconstruction of 3D models, estimation of an accurate depth map is essential from which we can obtain a dense 3D point cloud followed by handling the problem of reflectance in light field images. We briefly review literature on these main steps.

3.1. General Concepts of 3D Reconstruction

There are a variety of publications looking at the problem of the reconstruction of 3D models with different approaches. Generally, these approaches can be divided to two groups. The first group works on solving the problem using multiple images and the second group focuses on solving the problem with just a single image. We discuss each group separately. Figure 9 shows the overall structure of 3D reconstruction with and without using a light field camera.

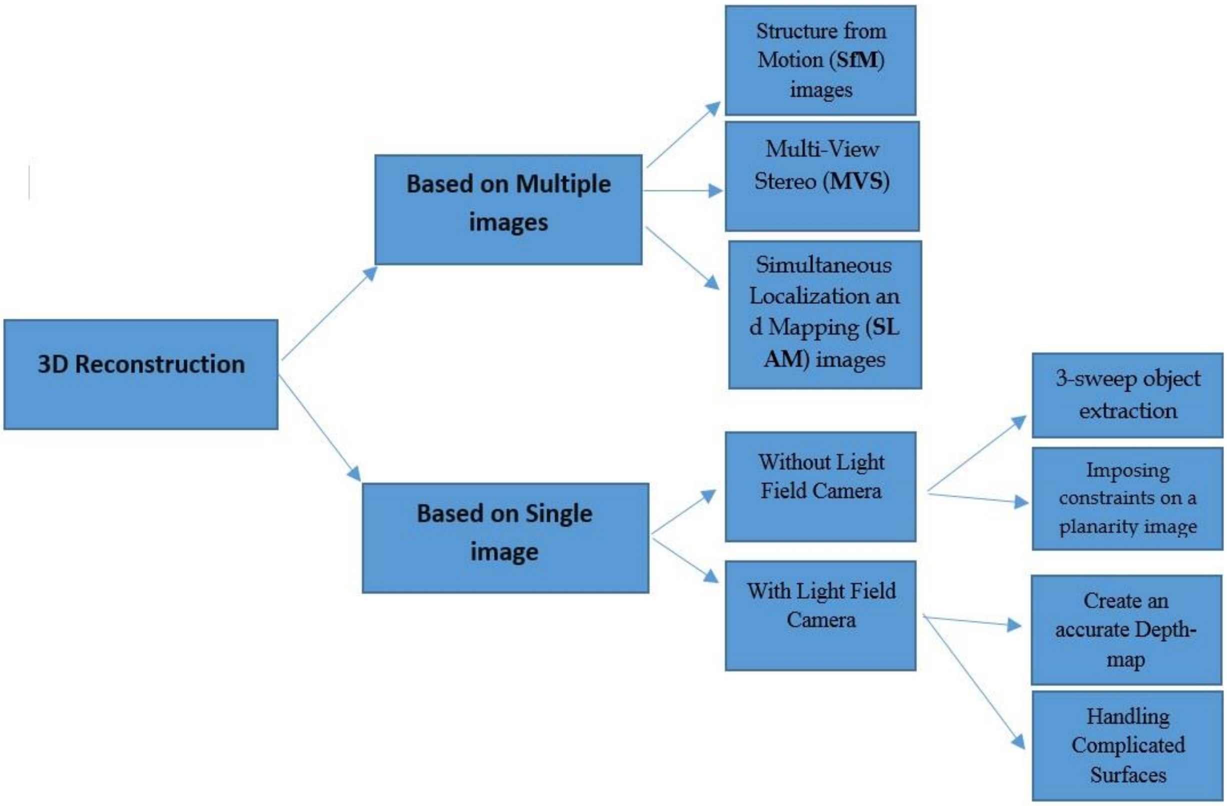

3D Reconstruction
Based on Multiple images
Based on Single image
Structure from Motion (SfM) images
Multi-View Stereo (MVS)
Simultaneous Localization an d Mapping (SL AM) images
Without Light Field Camera
With Light Field Camera
3-sweep object extraction
Imposing constraints on a planarity image
Create an accurate Depth-map
Handling Complicated Surfaces

Figure 9. Overall Structure of 3D Reconstruction. Generally there are two main approaches for recovery of a 3D model. One of them is based on a single image and the other uses multiple images.

3.1.1 3D Modelling from Multiple Images

The aim of a multi-image-based 3D model algorithm can be expressed as ”considering a set of photographs of a scene or an object, approximate the most likely 3D image that illustrates those pictures, under the hypothesis of known materials, viewpoints, and conditions of light” [30]. Initial methods for reconstruction of 3D models are mostly based on multi-capture techniques. Techniques for the reconstruction of 3D shape from multiple images is intended to create the structure of a scene or an object using solely geometric constraints in 2D images. In this field, several researchers have suggested different techniques and theories. These techniques and methods can be divided into three main categories [31]: Structure from Motion (SfM), Multiple View Stereo (MVS) algorithms and the Simultaneous Localization And Mapping (SLAM).

The Structure-from-Motion (SfM) algorithm is used to acquire the 3D object structure and the camera movement from a set of 2D images of static objects. This method estimates the localizations and direction of the camera and sparse features of images [31]. SfM requires repeatedly performing nonlinear optimization, which is similar to the SLAM algorithm.

Some early algorithms for various situations based on SfM have been described in [32, 33]. One of the typical SfM approaches was introduced by Snavely [34] and was applied to real-world objects for the creation of 3D models. They also used feature point matching and bundle adjustment to overcome the problems with 3D reconstruction of real scenes and objects like famous landmarks and cities. Since the SfM algorithm is slow and cannot eliminate outliers, it is hard to use for many applications because it involves time-consuming calculations, but it is able to provide 3D information from a series of images without utilizing any additional information. With developing hardware technologies, the SfM algorithm can be used in various fields. Some modified SfM approaches have been developed in [35, 36], which increase the speed of the calculation without a reduction in accuracy. Compared to hierarchical SfM and global SfM, the incremental SfM is a more common method for the 3D reconstruction for unordered images. Two main components in incremental SfM are the point matching of features between photos and bundle adjustment.

Multi View Stereo (MVS) is the popular phrase given to a group of methods that use stereo correspondence as their principal cue when the camera parameters (positions and orientations) are determined. The MVS algorithm reconstructs the 3D structure of a scene or an object by using more than two images. Furukawa *et al*. [37] introduced a method for reconstruction of 3D models based on MVS by matching the distribution of Gaussians and Harris corner points between several images. For matching other pixels between photos, they used patch expansion. Some other researchers have developed different 3D reconstruction techniques based on depth-map combination. These methods can acquire 3D model results with both higher compression and accuracy. The approach suggested by Shen *et al*. [38] is one of the most significant methods based on MVS. This approach besides using the position and orientation information of the cameras, utilizes the sparse feature points coordinates obtained from the calculation of the structure. In this way, they can obtain a depth map and followed by the creation of a 3D point cloud. However, Furukawa's method [37] depends on the texture of the images. In cases where the image has poor texture, it is hard for this approach to create a dense 3D point cloud. Moreover, this approach needs a long calculation time because the process involves the frequent use of patch expansion. In contrast, Shen's approach simply creates a compressed 3D point cloud using a depth-map combination. Qu *et al*. [31] developed an approach similar to Shen's method which can simply and rapidly create a dense point cloud.

Simultaneous Localization And Mapping (SLAM) involves the simultaneous modelling of the environment and estimation of the viewpoint position as that view moves through the environment. The structures are acquired by SLAM are mostly used to support other problems. SLAM approaches are commonly used for indoor applications such as Unmanned Aerial Vehicles (UAV) and mobile robotics. Early SLAM algorithms were based on Kalman filters and maximum likelihood estimation. Generally, the majority of SLAM approaches are based on iterative nonlinear optimization. One of the significant problems of SLAM algorithms is they can easily be trapped in a local minimum wherein they cannot estimate the structure accurately. For handling this problem some researchers used convex relaxation to fix the problem of convergence to a local minimum. Liu *et al*. [39] is one example of such an attempt. They introduced an improved SLAM approach to adjust to several applications such as vision-based navigation and mapping.

3.1.2 3D Modelling from a Single Image

The purpose of this approach is to create a 3D model solely by using a single image. These types of approaches can use a specific camera such as a light field camera or tackle the problem by applying user interaction. First, we will discuss the methods involving user interaction and then we will show that using a light field camera for creation of the 3D model will be much simpler and more accurate. Most recent research into overcoming the problems of 3D reconstruction has focused on using a single image. Chen *et al*. [40] have developed an interactive method to extract and manipulate simple 3D shapes based on a single image. Such extraction needs knowledge of the shape's components and relationships between the components of the image. Making these tasks automatic for a computer or machine is a difficult task. Therefore, their method combines the cognitive skills of humans with the computational accuracy of the machine to build a 3D model. In their approach, the individual needs to draw three strokes over the image to create a 3D model that snaps to the schema of the shape. Each stroke determines one of the component dimensions. Such manual intervention can divide a complex object completely into components. Jelinek *et al*. [41] tried to solve the problem of 3D reconstruction from a single image by using a camera of unknown focal length. They assumed the shape of object as a polyhedron where the coordinates of the vertices can be represented as a linear function of a dimension vector. They considered a set of correspondences between features of the image as an input of their system. They also considered a suitable projection model for the camera and the object dimension. In the case of perspective projection, the focal length of the camera is determined. They also used nonlinear optimization techniques by sampling the parameter space uniformly. Sturm *et al*. [42] proposed an approach for creation of 3D model of objects from a single panoramic image. Their method is based on user-provided coplanarity, perpendicularity and parallelism constraints. However, the application of their proposed method is for a parabolic mirror-based omnidirectional sensor. Xue *et al*. [43] dealt with the problem of 3D reconstruction by using single-view modelling. They recovered the 3-D geometry of a symmetric object with minimal user interaction using symmetry as the most common attribute of natural or manmade of objects. Considering a single view of a symmetric object, the user can mark some symmetric lines and depth dissimilarity regions in the image. Their method first detects a set of planes as a rough approximation to the object, and then a rough 3D point cloud is created by an optimization method.

Zhang, [44] presented a method for 3D reconstruction from a single photograph. They used free-form curved surfaces with arbitrary reflectance properties for dealing with this problem. The main point of their method is a hierarchical transformation structure for accelerating convergence on a non-uniform, piecewise continuous grid.

Zou *et al*. [45] suggested a semi-automatic 3-D modelling method to create a 3D geometry from a single photograph of a piecewise planar object by using minimal user interaction. Their method included three main steps. Firstly, a rough sketch is taken as an input. Secondly, an estimate of a rough 3D model for valid for a large class of objects is estimated and thirdly, the hidden part of an object is recovered and a complete 3D model created. Jiang *et al*. [46] proposed a method to reconstruct a 3D texture-mapped architecture model again based on a single image. They approached the problem by applying limitations derived from symmetries of shape, which are common in architecture. They first calibrate the camera from a single photograph by exploiting symmetry. Then they create a 3D point cloud based on the calibration and the symmetry of underlying structure. After creating the initial point cloud, a user needs to mark out the architecture structure of shape and the positions of the 3D point cloud. At the end they enhanced the quality of texture in occluded regions. They also claimed that their method can act faster than some other methods using a single image.

Toppe [47] proposed the concept of relative volume constraints for solving the difficult problem of 3D reconstruction from a single image. The main idea of the approach is to formulate a variational reconstruction method with shape priors in the form of relative depth map or volume ratio information regarding object components. This information can easily be acquired from a user sketch or from the segmentation of the object's shading in the photograph. They used error propagation to mitigate the problem of shadows in the image. They also tried to solve the difficulties of occlusions and holes.

Yan *et al*. [48] introduced an approach for the creation of 3D models of flowers based on a single photograph. Recovery of 3D shape from a picture of flowers is a challenging topic because of the ambiguous shape and the head of flowers includes petals embedded in 3D space. Their technique first applies the shape of a cone and subsequently a surface of revolution to the structure of the flower. Then single shapes of petals are calculated from their projection in the image. For flowers with diverse layers of petals, they proceed with different layers separately. The occlusions problem is handled by both within and between petal layers.

Most 3D modelling approaches that directly recover 3D geometry from a single image need significant user interactions during the process. However, a few recent works use light field images for recovering 3D models such as Perra *et al*. [10] which tried to reconstruct a 3D model based on a light field image. Their method works based on the original depth map acquired from the Lytro software. As they do not have any modification of the depth map of the light field image, when their algorithm is applied to real-world images, it cannot produce an accurate result, especially in the case of occluded images. In contrast, we developed an approach to estimate the depth map more accurately compared to the default Lytro software and then modified it by using two different approaches. The obtained point cloud is denser compared to Perra's method.

3.2. Light Field Depth Estimation

Existing LF depth estimation methods can be separated into three groups: (i) approaches based on sub-aperture image matching, (ii) Epipolar Plane Image (EPI) based methods, and (iii) LF data machine learning-based approaches [22].

3.2.1 Sub-Aperture Image Matching Based Methods:

In 4D light field images, the set of sub-aperture images is arranged in a 2D array. A single scene point corresponds to a pixel in multiple sub-aperture images. Corresponding pixels across sub-aperture views are related by non-integral (subpixel) shifts between pairs of sub-aperture images [49]. Pairs of sub-aperture images hence represent very small baseline stereo views of the scene which can be used with standard stereo matching algorithms to estimate a depth map for the scene. To account for the very narrow baseline, Jeon *et al*. [50] introduced a method that relies on a cost volume per-pixel. A cost volume can improve the accuracy of depth maps in poorly textured areas by using multi-label optimization. Inoue and Cho [51] used a pixel rearrangement method to improve depth map estimation. Their method is based on calculating the pixel blink rate (instead of using block matching, they measured the depths of pixels directly) and they used the total number of pixels in an image sensor to improve the quality of the recovered 3D image. However, this method has problems with images with occlusions.

To address the occlusion problem, Wang *et al.* [52] proposed a strategy for finding occluded edges based on dividing the image into angular patches and using photo consistency in the regions that solely contain occlusions. However, their approach has some limitations in overcrowded and noisy image regions. Anisimov *et al.* [53], proposed a very fast technique to estimate the depth map using a methodology based on stereo matching. Their method used enlargement in the sampling of multi-view light field images with a correspondence search across viewpoints. Wang *et al.* [54] extended the classical method of semi-global matching to obtain a real time depth map estimate. They used a multi-view stereo matching framework and calculated the level of disparity at a sub-pixel level to partly overcome the problems of narrow baselines. However, this method produces poor depth map estimates in noisy image areas. The coordinates of the viewpoint in the physical space corresponding to the sub-aperture image (s,t) are $\binom{S}{T} = \frac{D}{d}(D+d)\binom{s/l_u}{t/l_v}$, where D is the distance between the viewpoint and the middle of main lens, d is the distance between the viewpoint and imaging sensor, and l is defined as a focal length of the main lens. When we consider a uniform focal length (i.e. $l_u = l_v = l$), the baseline between two adjacent sub aperture images can be computed as $Baseline = \frac{(D+d)D}{dl}$ [22].

3.2.2. Epipolar Plane Image (EPI) Based Methods

In addition to the methods described above based on matching of sub-aperture images, another class of methods estimate the depth map of light field images using EPIs created directly from the microlens sub images. Several EPI-based methods for estimating depth maps are based on using the horizontal EPI ($I_{s*,t*}(u, v)$ acquired by fixing v and t) and the vertical EPI ($I_{u*,v*}(s, t)$ acquired by fixing u and s) slices, because these two EPI slices are simple to extract and manipulate [22]. Wanner *et al*. [3] used the local structure tensor of an epipolar plane image in the spatial and angular direction for orientation estimation. However, the tensor structure depends on a high angular resolution, and becomes inaccurate when occlusions appear. Houben *et al*. [55] worked on how to best process noisy light field images for depth map estimation. They combined a fast de-noising structure in a depth map estimation framework that is applied to the EPI domain. Chantara *et al*. [56] used the Modified Structure Tensor (MST) method for estimating the depth map and obtaining correspondence information in EPI images. Ziegler *et al*. [57] improved depth estimation by converting the EPI imaging space to a 4D EPI volume as a holographic representation of the same light field data. They claimed there were benefits in depth estimation using both standard light field and holographic representations. For handling occlusion problems, Zhang *et al*. [58] developed an occlusion removal framework based on EPIs to handle occlusion problems in integral imaging. They developed a contour-based method for object extraction by removing the occlusion that in turn allows for efficient extraction of 3D objects. Tosic *et al*. [59] developed an approach for estimating the depth by using the normalized second derivative of the Ray Gaussian kernel. Johannsen *et al*. [60] developed a method for one-by-one encoding the upper and lower parts of the EPIs to handle occlusions. The orientations were separated into eight classes which matched any of four directions, but based on horizontal and vertical EPI patches. However, all the aforementioned methods calculated the local disparity by relying on the horizontal and vertical EPIs that use less angular resolution information.

3.2.3. LF Data Machine Learning-Based Approaches

Learning-based methods are another approach to depth estimation aside from sub-aperture image matching and EPI methods. Convolutional Neural Networks (CNNs) have recently been successfully applied to various computer vision applications. Despite the success of CNNs, there are also some drawbacks. One main drawback is that most deep learning methods need huge, labelled datasets for training purposes. Unsupervised methods such as Jiayong Peng et al. [93] generally perform poorly compared to supervised methods and in some applications such as remote sensing and agriculture where point clouds and depth maps are used to extract quantitative data, the accuracy of deep learning techniques can be unclear due to the inherent difficulties in understanding the processing applied to the data and the types of distortion present. Although CNN techniques work well for detection and recognition when preparing a training database, they do not work accurately for estimation of the depth map of a scene as every scene has different details making it difficult to train a convolutional neural network. Zhou *et al.* [61] used multimodal methods from different formats of light field and three different architectures of convolutional neural networks based on pixel-wise classification for estimating the depth map. They found that discrete classification could potentially help to create a multimodal architecture through a probability distribution, but existing LF datasets are not sufficient for the purpose of training CNN methods. In order to increase the dataset size, some researchers have used computer graphics software such as POV-Ray to synthesize LF datasets [62].

3.2.4. Problems of Existing Approaches and Our Solutions

Each of the three different types of methods for depth map estimation have some limitations and most of methods require dense sampling for generating the light field image. Methods that solely use sub-aperture image matching suffer the problem of a narrow baseline for depth map estimation that results in the depth space being discretised. As a result, these methods do not produce accurate depth maps in noisy areas of the light field. In contrast, for overcoming the problem of a narrow baseline between sub-aperture images within a light field, we increased the distance between the viewpoint and the middle of the main lens to produce a wider baseline between sub-aperture images. Unlike stereo matching based on two adjacent views, our approach uses all views as represented by sub-aperture images. Furthermore, we combined this method with defocusing techniques to estimate the depth map with greater accuracy. In our work, we circumvented reliance on EPI based methods because they do not provide geometry and angular resolution information on pixels since they calculate the local disparity of the centre view based on the horizontal and vertical EPIs only. Deep learning approaches to depth map estimation cannot account for classification and object recognition applications, and any improvement attained with pre- and post- processing will come at an impractical performance cost.

3.3. Creation of a 3D Point Cloud

Existing 3D reconstruction approaches can be divided into three different groups: methods based on capturing multiple images, creating 3D point clouds based on deep learning and approaches based on 3D reconstruction from a single image.

3.3.1. 3D Reconstruction from Capturing Multiple Images

In general, the reconstruction of 3D point clouds from multiple image captures of the same scene is a computationally expensive process and requires significant user interaction [63]. One of the common methods in this field is Structure from Motion (SfM) which requires the capture of photos of a scene from all feasible angles around the object especially for fused aerial images and LiDAR (Light Detection And Ranging) data. d reconstructed three-dimensional volumes of rural buildings from groups of 2-D images by using SFM methods. For working on Unmanned Aerial Vehicle (UAV) images, Weiss *et al.* [64] utilized RGB colour model imagery for describing the vineyard 3D macro-structure based on the SFM method. Bae *et al.* [65] proposed an image-based modelling technique as a faster method for 3D reconstruction. For image capture, they utilized cameras on mobile devices. One of the benefits of using image-based modelling is the accessibility of texture information that can enable material recognition and 3D CAD model object recognition [66, 67] . Moreover, some approaches used the reconstruction of 3D scene geometry for purpose of controlling and management of energy in the field of 3D modelling of buildings [68]. Pileun *et al.* [69] estimated the positions and orientations of the object by using Simultaneous Localization And Mapping (SLAM). The 2D localization information is utilized for creating 3D point clouds. This reduces the time of scanning and requires less effort for collecting accurate 3D point cloud data but still needs user interaction.

3.3.2. Creating 3D Point Clouds Based on Deep Learning Techniques

Recently, approaches based on deep learning have drawn attention for solving many computer vision problems. A wide variety of deep learning models have been developed to create 3D point clouds but most of them require images capturing an object with an uncluttered background and a fixed viewpoint [70]. Current techniques have limited application to real-world objects. Yang *et al*. [71] generated a point cloud based on a specific deep model named PointFlow. This model has the advantage of having two levels of continuous flows for normalizing the point cloud. The first level is for creating the shape and the second level is for distributing the points. For handling large scale 3D point clouds, Wang *et al*. [72] developed a method based on the Feature Description Matrix (FDM) combining traditional feature extraction with a deep leaning approach. As deep learning alone is not efficient for creating a 3D point cloud, Vetrivel *et al*. [73] combined a convolutional neural networks approach with 3D features to improve results. Wang *et al*. [74] used deep learning for fast segmentation of 3D point clouds. They introduced a new framework called Similarity Group Proposal Network (SGPN). However, this method is still not efficient for real-world objects.

3.3.3. 3D Reconstruction from a Single Image Approaches

Creating 3D point clouds from a single image has received significant attention from the research community. However, 3D reconstruction from a single projection still has many problems and is a challenging topic. Mandika *et al*. [75] estimated 3D point clouds from a single input view by training an auto-encoder to learn a mapping from 2D input images to 3D point clouds. However this type of estimate is not very accurate and requires extensive training [76]. To overcome the drawbacks of estimation of 3D point clouds from a single capture, light field cameras have been proposed. Using light field images as an input can lead to 3D point cloud estimates with low cost and complexity. Perra *et al*. [10] used light field images as an input image to determine depth maps of scenes and then estimated 3D point clouds. The depth maps that are automatically acquired from light field cameras have some potential limitations when dealing with real objects. To tackle this problem, we propose a novel algorithm for enhancing the depth map by intelligently adding edge detection information which provides more information about the depth map. Moreover, compared to the Perra *et al*. [5] method, we used a transforming method for converting the depth map to 3D, which is more reliable and does not need the segmentation and the extraction of occlusion masks which was required by their method.

3.3.4. Problems of Existing Approaches and Our Solutions

Each of the three aforementioned method types have some limitations for creating dense and accurate 3D point clouds and most of them require considerable effort to obtain 3D data. Methods that make use of multiple images, such as SfM and SLAM, usually require finely textured objects without specular reflections. Moreover, if the baselines used for separating the viewpoints are chosen to be large, it causes many problems for feature correspondences on account of occlusion and changes in local appearance [70]. For deep learning approaches, despite the recent favourable results of deep learning models in some machine learning tasks, creating 3D point clouds remains challenging [71]. This difficulty can be attributed to the lack of order in the 3D point cloud, so no static structure of topology can be found for recognition and classification of the scene based on the deep learning. This means it will be problematic to use deep neural networks directly on the point clouds because points will not be arranged in a stable order like pixels are in 2D images.

Many attempts have been made to address the problem of 3D reconstruction from a single snapshot, however, when those methods are applied on real images, they will suffer from a sparsity of information. Since in the one snapshot there will be only a single view of the image, some parts of the scene will be invisible. This shortage causes the generation of a 3D point cloud to be problematic. In contrast, we used a light field camera in this work, which by one snapshot provides multiple sub-aperture images, providing sufficient data for depth map estimation. As a result, compared to other single image methods, we can overcome the problem of this kind of deficiency. Moreover, we enhanced the depth map intelligently by extracting SURF features from central sub-aperture matching and depth map images as well as fuzzy logic. This kind of enhancement provides for our work a more accurate point cloud compared to other methods in this area, and also makes the reconstruction of 3D point clouds much easier.

3.4. Light Field Reflectance Recovery

Over the last decade, many approaches have looked at the issues of recovering reflectance from imagery. However, dealing with complex material appearance, especially solving this problem with a single image capture, is still hard and challenging. To review the related work more specifically, we summarize works on illumination and reflectance separation in this section into two categories based on whether the work uses a conventional camera or light field camera.

3.4.1. Handing Reflectance without Using a Light Field Camera

Recovery of reflectance is an attractive research field and some researchers have utilized multi-capture imagery based on rotating platforms such as Pham *et al*. [77] applied to a hyperspectral imaging system. To estimate the relative reflectance, they take a white diffuse reflectance image and create dark reference images by turning off illuminating lamps at regular time intervals and obtained a diffuse reflectance estimate in doing so. This technique is related to the method of projecting a black-and-white moving grating while taking images to estimate the surface reflectance. Similarly, Rahman *et al*. [78] used a single hyperspectral image and considered the radiance of the image as a combination of diffuse and specular reflection components and introduced a cost function optimized by an iterative least squares algorithm. Jiddi *et al*. [16], developed a strategy for estimating and detecting the specular reflectance of real scene surfaces based on the both the colour and position of light sources. They introduced a method for solving the problem of reflectance estimation by incorporating and detecting the information from cast shadows and specular pixels. However, this method needs to capture the scene under near-ambient lighting to separate texture from illumination in coloured images. For estimating reflectance under conditions of multiple illumination sources, Chen *et al*. [79] utilized a Conditional Random Field (CRF) model and in each local patch they separated reflectance by incorporating spatial information. The CRF model is used for segmenting diversely illuminated patches. However, this reflectance model can cover only limited material appearance. Oxholm *et al*. [80] recover both reflectance of the surface and shape using a probabilistic geometry estimation method for natural lighting based on an expectation-maximization framework. However, this algorithm is limited by the difficulty of accurately modelling the Bidirectional Reflectance Distribution Functions (BRDF).

3.4.2. Reflectance Recovery Using Light Field Camera

It is desirable that the reflectance information be recovered by using one single capture which does not need user interaction. Using a light field camera is one effective solution to this problem. Kim and Ghosh [81] applied polarized light-field imaging solutions for separating specular and diffusion pixels for handling the problem of human skin reflection estimation, but their technique could not acquire accurate information on microlens positions. Ngo *et al*. [13] used both a light field camera and a 360-degree camera to improve the results. They captured the illumination with a 360-degree camera and they used a Directional Statistics Bidirectional Reflectance Distribution Function (DSBRDF) model to estimate the reflectance. Similarly, Jeong *et al*. [82] worked with the 360-degree light field image. For obtaining a 360-degree light field image, they used two mirrors and a light field camera. One key application of reflectance recovery is for creating an accurate depth map and reconstruction of a 3D point cloud [83, 84]. To this goal, Wang *et al*. [15] introduce a method for analysing diffuse pixels by using a Spatially-Varying (SV)BRDF-invariant theory. They derived an equation relating depths and normals which can estimate the depth map more accurately. Tao *et al*. [17] developed a solution for both Lambertian and glossy objects. However, there are no effective techniques using a light field camera that can cover more general reflectance.

3.4.3. Problems of Existing Approaches and Our Solutions

Among the methods reviewed above, each has some limitations in recovering the reflectance, especially in face of the challenge of real-world data. On the one hand, the main drawbacks of the methods that make use of multiple images (methods without using light field capture), consist in requiring considerable effort and extensive user intervention. Moreover, some of these methods even require a video sequence of object motion as an input which could be considered as an excessive number of views. On the other hand, some attempts have been made to address the problem of reflectance from a single capture by using a light field camera. However, when those methods are applied on real images, they cover only limited types of material appearance and most of them assume a Lambertian model for the reflectance of the object and dealing with the complex behaviour of realistic BRDF is difficult based on their methods. Hence, they focus on optimizing just the consistency between the viewpoints which lead to challenging or poorly defined optimization problems.

For solving both problems, in this work we used a light field camera to reduce the expensive effort of user interaction. Furthermore, some methods that used only light source colour analysis on light field image, suffer from the problem of the small-baseline of sub-aperture images of light-field data. Therefore, using solely analysis of pixel value in the colour space of the light field image cannot detect all specular components and highlights [85]. In contrast, we combined two different approaches for finding the specular components. Concentrating on both analysing the values of pixels in colour space and binarized space in different viewpoints can help us to cover wide range of real-world complicated material appearance. Moreover, we utilized the combination of Bernsen's local thresholding technique with a higher intensities technique for finding specular highlights in binarized space. This means our proposed model can be considered as a kind of general reflectance model. As a result, compared to other methods, we can overcome the problem of reflectance estimation more accurately without excessive complexity. Other researches trying to handle this problem by using a single image from a conventional camera have suffered from a shortage of the required information because in a single view of the scene some parts of the scene will be invisible. This lost information causes serious problems in recovering the reflectance. In contrast, in this work, we used a light field camera, which in one capture can act as a multi-camera array and provides sufficient data for recovering the reflectance.

Chapter 4: Proposed Depth Map Estimation Method

By using a captured light field for estimating the depth map of a scene, there is no need for labour intensive user interaction characteristic of photogrammetry or expensive equipment such as LIDAR systems. Figure 10 shows a block diagram of the proposed method. Our approach is based on combining sub-aperture image matching with defocus information utilizing the 4D LF format. We use this combined structure to address some of the problems discussed in Section 3.2 overcoming the weaknesses of each component method in realistic time. We compared the LF data types and found that the best format for producing accurate depth map estimates is the 4D format rather than the lenslet format. The four-dimensional light field description contains multiple views of the scene in an easily accessed format and can be generated from lenslet format images.

To create a better result for occlusions, our approach uses Canny [86] edge detection applied to the central sub-aperture image to identify occluded edges. We used the MATLAB implementation of the Canny edge detector. This implementation makes use of two thresholds that in our work we allowed MATLAB to set automatically. Based on the detected occlusions we can solve the problem of accurate depth estimation for occluded areas by extending the approach of Wang *et al.* [52]. Moreover, we combine correspondence matching by using defocusing features enabled by the light field format. Finally, for handling noise and improving accuracy, we used a fast-weighted median filter to further improve the depth map. The operation of combining defocus cues and correspondence matching cues was followed by a noise filtering operation. The rationale for the combination of cues is the fact that defocus cues are dense, but tend to be affected by noise, while correspondence cues are sparse, but when correspondences are found, they are quite accurate. By combining both we can take the advantages of both approaches. Even with the combination, residual noise in the depth estimate remains, and a filter is applied to reduce this noise. We illustrate the effectiveness of our approach on a number of synthetic LF image examples and real-world light field data sets. Our experimental results show higher performance than the classic and recent state-of-the-art light field depth estimation approaches. Our method was tested on real-world data, which was collected using a Lytro Illum, one of the current two major light field cameras (Lytro and Raytrix) available in the market.

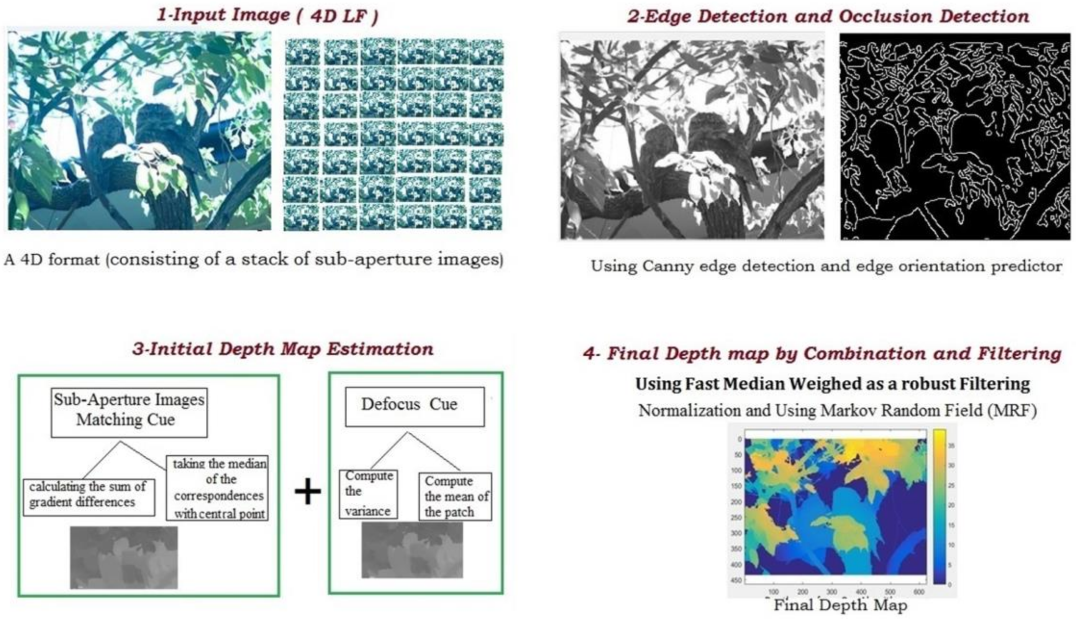

1-Input Image (4D LF)
A 4D format (consisting of a stack of sub-aperture images)
2-Edge Detection and Occlusion Detection
Using Canny edge detection and edge orientation predictor
3-Initial Depth Map Estimation
Sub-Aperture Images Matching Cue
calculating the sum of gradient differences
taking the median of the correspondences with central point
+
Defocus Cue
Compute the variance
Compute the mean of the patch
4- Final Depth map by Combination and Filtering
Using Fast Median Weighed as a robust Filtering
Normalization and Using Markov Random Field (MRF)
Final Depth Map

Figure 10 . Overview of the proposed method on a real Light field image captured with a Lytro Illum camera. Our method consists of 4 stages: 1. 4D format of light field image as input images, 2. Edge detection for occlusion identification, 3. Initial depth map estimation using correspondence matching and de-focus cues, 4. Combination, filtering, normalization, and final depth estimation.

Our algorithm is structured as: 4D format of light field image as input images, edge detection (Canny edge detection, occlusion detection), initial depth map estimation, filtering and normalization, and final depth estimation. We will discuss each part below.

4.1 Using 4D LF as an Input LF Format

For solving problems in state-of-the-art LF image processing, one of two LF data representation formats are common: lenslet or 4D LF. For purpose of depth map estimation, several methods make use of the lenslet format as it is a more compressed version of the LF. However, this format needs additional camera metadata and processing steps prior to image processing. In contrast, 4D LF data can be viewed as a stack of sub-aperture images, requiring only a small amount of side information to be used in subsequent image processing operations [23]. Many previous approaches used the lenslet format as an input, but as shown in Figure 11, lenslet format images can have different lenslet shapes that can cause undesired effects in pixels near the lenslet edges. This situation is compounded by the fact that the size and shape of gap regions in the representation between lenslet images can vary substantially according to the type of lenslet image.

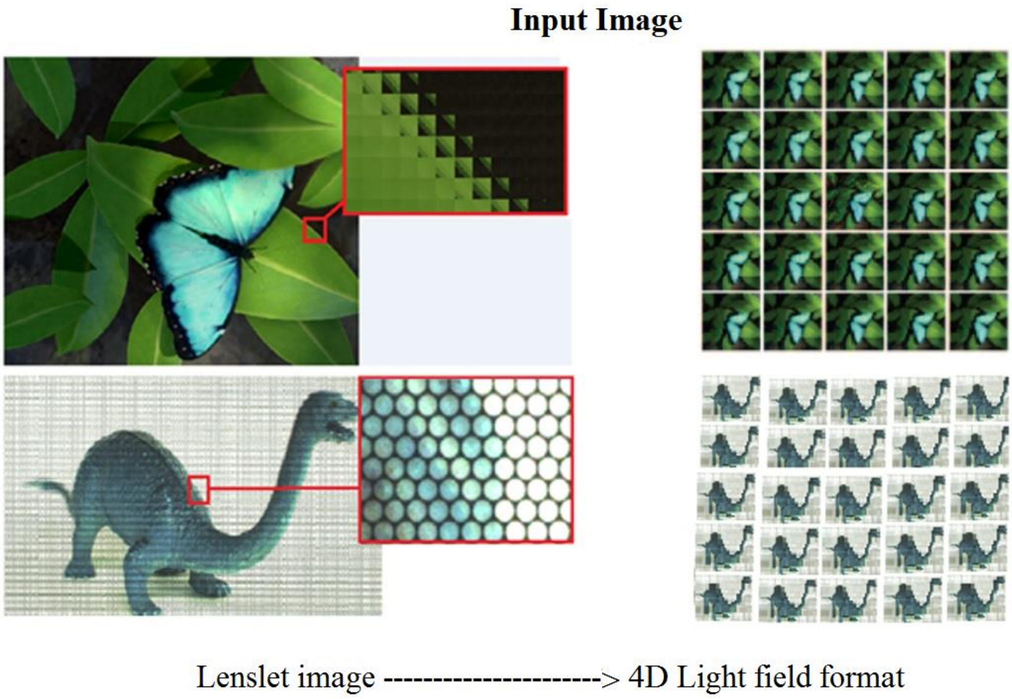


Figure 11. Examples of different shapes of lenslets used in light field images (a lenslet grid can be a hexagonal, circular or square shape). In comparison, the 4D light field format is consistent across different lenslet designs.

Our justification for using the 4D LF format is based on a comprehensive experimental validation that we have performed to compare these two types of image representations. For the reasons above we used the 4D light field format for our approach. A digital coloured 4D LF, which is a collection of sub-aperture images, can be obtained by sampling the 4D LF function defined as $L(u, v, s, t)$, Given colour image formation, it can be seen that each pixel in the LF is a three element vector with radiance information represented as red, green, and blue elements or any corresponding wavelengths [87].

A 4D LF is obtained from raw light field images saved in the Lytro Illum native file type (LFR), with a size of about 50 MB/image.

4.2 Edge Detection and Occlusion Detection

For edge detection, we used the Canny edge detector in MATLAB 2017b [86] to extract edge information. Edge detection is used on the central view (sub-aperture image) to detect the edges. Then for obtaining the orientation angles at each edge pixel we used an edge orientation predictor. These pixels (edge pixels) are counted as candidate occlusion pixels in the central view similar to Wang *et al*. [52]. Since it is likely some pixels occluded in non-central views are not occluded in the central view, we dilate the edges found in the central view. This helps to avoid missed candidate occlusion pixels due to noise or other artefacts. The benefit of this occlusion prediction framework is to ensure constraints of visibility while estimating the depth map by avoiding missing pixels occluded in non-central views, but not the central view. A robust depth estimation technique which explicitly takes occlusions into account can modify angular consistency and the visual compatibility across sub-images and estimate the depth map more accurately. This is especially needed in some scenes with complex occlusions that do not have smooth object boundaries estimation of depth map, where not considering the problem of occlusion can reduce the accuracy of depth estimation.

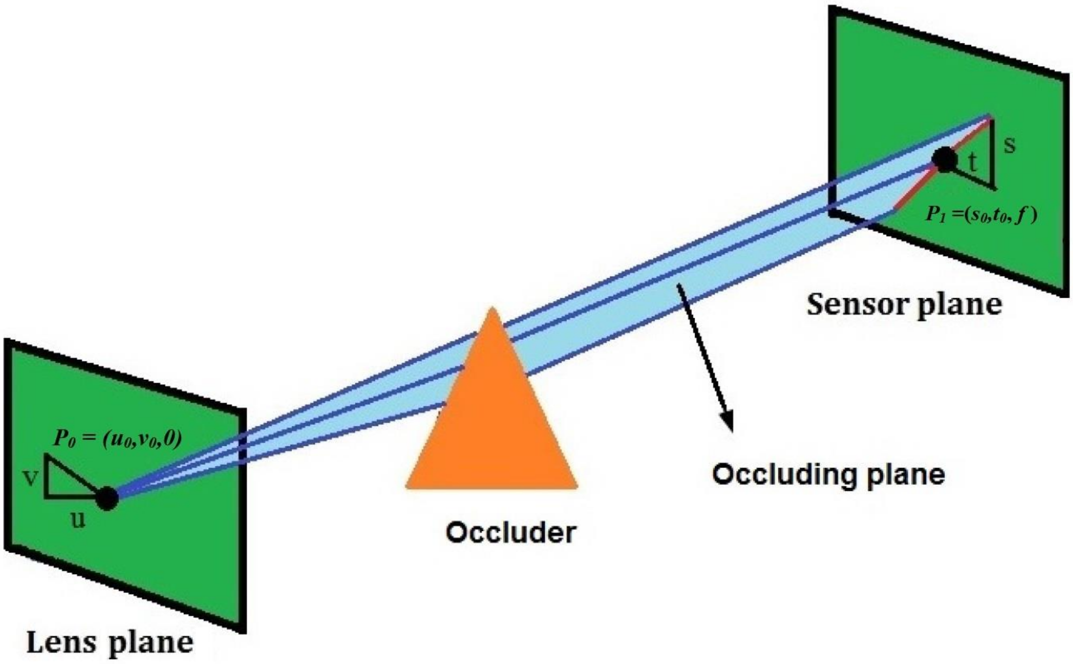


Figure 12. An occlusion edge on the imaging planes (Sensor plane and lens plane) corresponds to an occluding plane in the 3D space K plane (the light blue shape).

Following the theory of Wang *et al.* [52], each sub-aperture view can be divided into a grid of equal sized square patches termed angular patches. Given this theory, considering a pixel at (s_0, t_0, f) on the imaging sensor plane and an arbitrary point (u_0, v_0, 0) on the lens plane, if this pixel is on an occlusion edge, then it can be described as the occlusion edge via an occluding plane *P(s, t, z)* as shown in Figure 12. The plane takes the form of a 2D triangular patch in 4D space where the apex of the triangle is at (u_0, v_0, 0) in the lens plane and the base of the triangle passes through (s_0, t_0, f) on the sensor planes as indicated in Figure 12. Where f is used to show plane P in 3D space. The occluder intersects *P(s, t, z)* with z ∈ (0, f) and lies on one side of the planes. The aim of this theory is that in the angular patches, the edge that separates the unoccluded and occluded pixels would have the same orientation as the occlusion edge in the domain of spatial.

For this purpose we have used the code [111] from Wang *et al.* [52]. As shown in Figure 12 considering a pixel at (s_0, t_0,*f*), on the focal plane, where $s0$ and $t0$ are the coordinates of the central sub-aperture image (*s, t*). first, the normal **n** of a pixel at the sensor plane (s_0,t_0, f), can be obtained from using Eq (1). An edge in the central view with 2D slope perpendicular to **n** corresponds to a plane K in 3D space:

$$\mathbf{n} = (s_0, t_0, \mathrm{f}) \times (s_0 + 1, t_0 + \mu, f) = (-\mu f, f, \mu\, s_0 - t_0) \qquad (1)$$

where × denotes the cross product and $\mu = v_0/u_0$ is the slope that corresponds to the plane *K* in 3D space which is shown as the blue plane in Figure 12 similar to Wang *et al.* [52]. Once at this stage we have an equation for the occluding plane for any value of (u_0,v_0, 0) across all sub-aperture images, but we do not know whether the plane passes through an occlusion or not. Consider a position, $z \in (0,f)$, between the lens and sensor planes on the occluding plane *K*. Then points on the occluding plane have the relationships given by Eqs. (2) and (3) below:

$$K(s,t,z) \equiv \langle \mathbf{n}, (s_0 - s, t_0 - t, f - z) \rangle = 0, \qquad (2)$$

Expanding the dot product and collecting terms we obtain:

$$K(s,t,z) \equiv \mu f s - f t + (t_0 - \mu s_0) z = 0 \qquad (3)$$

For a pixel (s_0, t_0,*f*), (u_0,v_0,0) to not be part of an occlusion, the line segment connecting the Point P_0 = (u_0,v_0,0) on the lens plane and the point P_1 =(s_0, t_0,*f*) on the sensor plane must not pass through the occluder.

From Wang *et al*. [52], this condition was met when Eq. (4) held:

$$\mathrm{K}\big((u_0 + s_0 b - u_0 b\ ), (v_0 + t_0 b - v_0 b\ ), fb\big) \leq 0\ \forall b \in [0,1], \tag{4}$$

where $P_0 = (u_0, v_0, 0)$ and $P_1 = (s_0,\ t_0, f)$.

$$\text{when } b = 1\ ,\quad K(P_1) = 0. \tag{5}$$

$$\text{when } b = 0,\ K(P_0) = \mu f u_0\ -\ f v_0\ \leq\ 0.$$

The last condition ($b = 0$) will be valid if $v_0 \geq \mu u_0$. When (s_0, t_0, f), $(u_0, v_0, 0)$ lies on an occluder, then Eq. (4) will not hold and $K(s, t, u, v) \geq 0$. Therefore, that means that the critical slope on the angular patch $v_0 \geq \mu u_0$ is equal to the edge orientation in the spatial domain. This is a general result is not dependent on the specific value of f.

The combined occlusion map across all of the sub-aperture images given by:

$$OC(u, v) = \sum_{s,t} K(s, t, u, v) \tag{6}$$

Where the $OC(u, v)$ is the map of occlusion.

4.3 Initial Depth Map Estimation

In this section, we show how to compute the initial depth estimation. Our method is based on combining defocusing and sub-aperture image matching on 4D images which results in both a reduction in the image search time, and more accurate results for real images. In [50], the work was applied to lenslet images and did not pay significant attention to occlusions. In contrast to [50], we first estimate the occlusion pixels then combine sub-aperture image matching with defocus information to a get more accurate result. The authors of [5] combined defocus with correspondence matching using an Epipolar Plane Image (EPI) method. However, we use correspondence matching across sub-aperture images as opposed to correspondence matching using the EPI. The EPI is acquired by fixing the coordinates in both a spatial and an angular dimension and needs a high angular resolution of sub-aperture images to be accurate. Thus, we used multiple sub-aperture images instead of the EPI-based approach.

4.3.1. Sub-Aperture Image Matching Cue

As described above, a light field image can be decomposed into an array of sub-aperture images. A sub-aperture image can be created by light rays coming at the sensor plane from one spot in the lens plane. Each pair of sub-aperture images represent a narrow baseline difference in apparent position on the lens plane [22]. Due to the narrow baselines, the sub-pixel difference in the spatial domain generally includes interpolation with resultant uncertainties, which yields weak results for correspondence-matching methods. For our method, rather than using stereo matching, which is based on two views, all views as represented by sub-aperture images are contained in the constraints. Since we are using the 4D format then the 4D parameterization is used where the pixel coordinates of a light field image *I* are defined using the 4D parameters of *(s, t, u, v)*.

In order to match sub-aperture images, we calculate the sum of gradients. Sub-aperture images matching with using sum of gradients further enhanced the accuracy of depth map [50]. We start by defining P, a 2D vector that contains the s-t coordinates of the non-centre view containing the correspondence and Pc a 2D vector that contains the s-t coordinate of centre view. Ps and Pt are the s and t components of P, and *I(P)* is the grayscale value of P. We used the SIFT algorithm for the correspondence matching [88]. Similar to Jean et al. [50], we define the 2D shift vector $\Delta U(P,l)$ as below:

$$\Delta U(P, l) = lh(P - Pc) \tag{7}$$

Where l is depth cost label, which is related to number of depth labels in the final depth map [50] and h is the shift unit of the label in pixels. The pixel shift unit, h, varied according to the dataset [50]. Figure 13 shows a sample of calculating the 2D shift vector $\Delta U(P,l)$. The obtained 2D shift vector $\Delta U(P,l)$, contains two elements, indicating the shift in both the s and t directions. In other words, $\Delta U(P,l)$ includes both $\Delta U(Ps,l)$ and $\Delta U(Pt,l)$. The gradients are then given by:

$$Diff_s(Pc, P, s, l) = |I_s(Pc, s) - I_s(P, s + \Delta U(Ps, l))| \tag{8}$$

$$Diff_t(Pc, P, t, l) = |I_t(Pc, t) - I_t(P, t + \Delta U(Pt, l))| \tag{9}$$

$Diff_s$ defines the s-directional gradient of the sub-aperture images and $Diff_t$, defines t-directional gradient of the sub-aperture images.

In addition, we define,

$$\gamma(P) = |s - s_c|/(|s - s_c| + |t - t_c|) \tag{10}$$

where $\gamma(P)$ controls the relative importance of the two directional gradient differences based on the distance in (s, t) space from the currently considered sub-aperture image (s,t) and the centre sub-aperture image (s_c,t_c).

For each correspondence, the sum of gradients $GD(s, t, P, Pc)$ is obtained by a weighted sum of the *s* and *t* gradients:

$$GD(s,t,P,Pc) = \gamma(P)\min(Diff_s(Pc,P,s,t),\delta) + \left(1 - \gamma(p)\right)\min(Diff_t\ (Pc,P,s,t),\delta) \tag{11}$$

where δ is a small constant value that helps to remove outliers.

From a pair of matched feature positions (P, Pc), the positional deviation ΔP_d in the s, t coordinates is computed as

$$\Delta P_d = \sqrt{\Delta \mathrm{U}(\mathrm{P},\mathrm{s})^2 + \Delta \mathrm{U}(\mathrm{P},\mathrm{t})^2} \tag{12}$$

If the amount of deviation ΔP_d exceeds the maximum disparity range of the light field camera, (P, Pc) are rejected as outliers and $GD(s, t, P, Pc)$ is set to 0.

To obtain values of $GD(s, t, P, Pc)$ for the central view (s_c, t_c), we take the median of the correspondences with *Pc* across all of the non-central sub-aperture views.

Each correspondence is then modified by the intensity difference and the value of $GD(s, t, P, Pc)$ can be obtained as below:

$$GD(s,t,P,Pc) = GD(s,t,P,Pc) + \|I(P) - I(P_c)\| + \left\|I(P) - \overline{I(P)}\right\| \tag{13}$$

where $\overline{I(P)}$ is the median intensity value of the correspondences in s, t and the double vertical line here means, the absolute value of $I(P) - I(P_c)$.

A full-size map $\acute{GD}(s, t, u, v)$ is created by allocating an array for each sub-aperture image of the same size as the sub-aperture images containing only zero values at each pixel location. For each correspondence, $\acute{GD}(s, t, u, v)$ is set to $GD(s, t, P, Pc)$. This results in a map of gradients for each sub-aperture image which contains the value of $GD(s, t, P, Pc)$ for locations where a correspondence was found and zeros elsewhere.

To obtain the final depth map based on correspondence matching we sum across all sub-aperture gradient maps, as represented in Eq. (14).

$$DM(u, v) = \sum_{s,t} \acute{GD}(s, t, u, v) \tag{14}$$

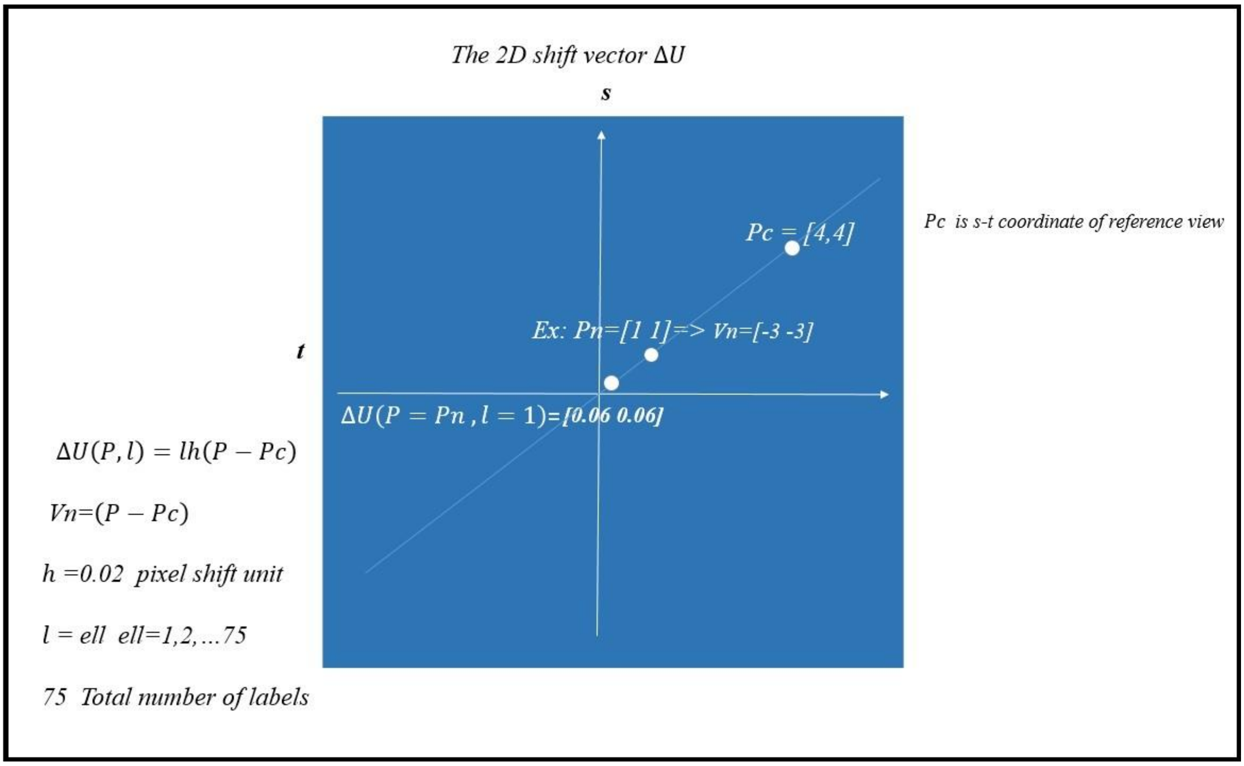

The 2D shift vector ΔU
s
Pc = [4,4]
Pc is s-t coordinate of reference view
Ex: Pn=[1 1]=> Vn=[-3 -3]
t
ΔU(P = Pn , l = 1)=[0.06 0.06]
ΔU(P, l) = lh(P − Pc)
Vn=(P − Pc)
h =0.02 pixel shift unit
l = ell ell=1,2,…75
75 Total number of labels

Figure 13: A sample of calculating the 2D shift vector $\Delta U(P,l)$*.*

4.3.2. Defocus Cue

Spatial and angular information captured by the light field camera supplies enough information to use defocus cues for depth estimation. In using defocusing cues for depth estimation, the optimal contrast is computed, and occlusions can have a strong effect on the outcome of the measure. However, since we detected the occlusions before this step our method will have improved stability across regions with occlusions. For each pixel, we refocus to various depths by exploiting the 4D format of the light-field data. For computing the depth value from defocusing information, DF_α, we first remap the 4D light field input image as follows, where a pixel (u, v) in the re-focused sub-aperture image (s, t) is given by:

$$I\alpha(s,t,u,v) = I(s + u(1 - (1/\alpha)\,), t + v(1 - (1/\alpha)), u, v) \tag{15}$$

where $I\alpha(s,t,u,v)$ is a pixel at position (u,v) within sub-aperture image (s,t) as described in Section 2.3. The coordinates (s,t) are the spatial coordinates and (u,v) are the angular coordinates. Angular information of the LF is obtained from the lens plane and the spatial information is acquired from the sensor plane.

A contrast-based measure can be used to find the optimal α with the highest contrast at each pixel. Following the method of [5] αmin = 0.2, αmax = 2, and αstep = 0.007 Tao et al. [5].

We calculate the variance and the mean of the patch to get the depth value from the defocus cues, following the method of Tao *et al.* [5]. However, we are working with a 4D light field image as input and working with the entire angular patch. First, we compute the means and variance of each patch and then calculate the minimum variance to obtain the depth from defocus cue. Define $I_{\chi}^{\alpha}(s,t,u,v)$ as a pixel at position (u,v) within patch χ in sub-aperture image (s,t) for the sub-aperture image refocused by ratio *α*. The central pixel in a sub-aperture image is located at (u,v) = (0, 0). Each sub-aperture image within the refocused light field $I\alpha(s,t,u,v)$ is then divided into a grid of *K* angular patches, χ_k, $(\chi_k = [(u_1,v_1),(u_2,v_2),(u_3,v_3),\ldots,(u_k,v_k),\ldots.,(u_K,v_K)])$, where (u_k,v_k) is the angular coordinate of the central pixel in the *k^th^* angular patch. Since angular patches contain noise and depth discontinuities, we process each patch and its neighbourhood to obtain an average, smoothed, depth value from defocusing information for each patch.

We first compute the means and then variances of the *K* angular patches, (χ_k). The mean of the *k^th^* angular patch is given by:

$$\overline{I_{\chi_k}^{\alpha}}(s,t) = 1/N_k \sum_{u_k,v_k \in N\chi_k} \left(I_{\chi}^{\alpha}(s,t,u_k,v_k)\right) \qquad (16)$$

where $N_k$is the number of pixels in *kth* patch. $N\chi_k$ is the set of pixels belonging to the *kth* patch. The variance of the *kth* angular patch is given by:

$$V_{\chi_k}^{\alpha}(s,t) = 1/(N_k - 1) \sum_{u_k,v_k \in \chi_k} \left(I_{\chi}^{\alpha}(s,t,u_k,u_k) - \bar{I}_{\chi}^{\alpha}(s,t)\right)^2 \tag{17}$$

For patch k, the variances of each angular patch within the 8-neighbourhood of patch *k* are compared and patch k is replaced by the neighbouring patch, *i*, with the minimum variance.

$$i = \arg min_k \ (V_{\chi_k}^{\alpha}(s,t))$$

(18)

the depth from defocus in patch χ_k is given by:

$$DF_{\chi_k}^{\alpha}(s,t) = \left(\overline{I_{\chi_k}^{\alpha}}(s,t) - \overline{I_{\chi_c}}(s,t)\right)^2 \tag{19}$$

where $\overline{I_{\chi_c}}(s,t)$ is the mean of the central patch of the current un-refocused sub-aperture image, $\overline{I_{\chi_i}^{\alpha}}(s,t)$ computed in a similar way to Eq. (16). Denote $DF^{\alpha}(s,t,u,v)$ as the depth from defocus estimate indexed by pixels, rather than patch by patch, for the (u,v)th pixel in the (s,t)th sub-aperture image [3].

The final depth from defocusing information is the average depth from defocusing cue across all of the sub-aperture images given by:

$$DF^{\alpha}(u,v) = 1/\left|N_{s,t}\right| \sum_{s,t \in N_{s,t}} |DF^{\alpha}(s,t,u,v)| \tag{20}$$

where $N_{s,t}$ is the set of sub-aperture images in whole image *I*.

4.3.3. Combination and Filtering

Given both the depth from defocus cue and the depth from correspondence matching cue, we combine them to obtain a more accurate result then refine the final depth map with a Markov Random Field (MRF). We combine the two estimations as follows:

This process starts by computing a combined depth map, $d_{comb}(u,v)$ by merging $DF^{\alpha}(u,v)$ (where α = 0.1) and $DM(u,v)$. Since the depth map from correspondence matching may not be valid for all pixels, the merged depth map, $d_{comb}(u,v)$, is created by choosing the depth map value from the defocusing cue when the acquired depth map value from correspondence matching is not valid (that is, it is zero), otherwise the combined depth map at (u,v) is the arithmetic average of $DF^{\alpha}(u,v)$ and $DM(u,v)$.

Then we compute an occlusion weighted depth map by merging the combined occlusion response $OC(u,v)$ (Eq. 6) and combined depth map d_{comb}.

$$\mathrm{d_{ocl}(u,v)} = \mathrm{N}\left(\mathrm{d_{comb}}(u,v)\right).\mathrm{N(OC(u,v))} \tag{21}$$

where $N(.)$ is a normalization function:

$$N(x) = (x - \mu_x)/\sigma_x \tag{22}$$

where μ_x is the mean of x and σ_x is the standard deviation of x.

In order to propagate the local estimation to regions with low confidence, we regularize $d_{ocl}(u, v)$ with a Markov Random Field (MRF) for a final depth map using the technique in Wang *et al*. [52]. The energy function used by the MRF regularization is given by:

$$E = \sum_p E_{binary}\left(\mathrm{p}, \mathrm{d_{ocl}}(\mathrm{p})\right) + \sum_{p,q} E_{binary}(\mathrm{p}, \mathrm{q}, \mathrm{d_{ocl}}(\mathrm{p}), \mathrm{d_{ocl}}(q)) \tag{23}$$

where d_{ocl} is the final depth, *p, q* are neighbouring pixels and E_{binary} is based on the gradient of the central pinhole image as defined in Wang *et al*. [52].

Equation (23), followed by the use of a fast-weighted median filter is consistent with the technique of Zhang [88] where this method was shown to improve the depth estimate while providing for the reduction of residual noise.

Finally, for further noise reduction we used the fast-weighted median filter described by Zhang *et al*. [89] applied to the final depth map d_{ocl}. The main concept of the weighted median filter is to substitute the current pixel by the value of the weighted median of neighbouring pixels. The bilateral filter is an example of a weighted median filter. However, rather than using the bilateral filter, Zhang *et al*'s approach used the joint-histogram and median tracking [89].

For this purpose, first we measure the joint histogram as below:

J is defined as a 2D joint histogram for the depth map where, *J(i,j)* is the frequency of the i^{th} intensity level occurring with the j^{th} value of the feature. Therefore, the total joint-histogram is produced as:

$$J(i,j) = \#\{p \in R(p) |\, I(p) = I_{i,} f(p) = jf_j\} \tag{24}$$

We consider pixels within the local window *R(p)* of radius *r*, where each pixel p in this filter is indicated by its value $I(p)$ and feature $f(p)$. $f(p)$ is the intensity component. For computing the weights,

$$W_i = \sum_{j=0}^{N_j-1} J(i,j) g(f_i, f(p)) \tag{25}$$

where g is a typical influence function between neighbouring pixels. For computing the weights as Eq. (25) measurement of the joint histogram is needed. This is performed to calculate the total weight W_t in the first iteration. Then in the second transmission, we aggregate weights up to half of W_t and output that pixel value [30].

We then obtain the median by shifting the cut point and checking the balance in Eq. (26). This process is the same as median tracking.

$$c = \sum_{j=0}^{N_j-1} C(f) g(f_i, f(p)) \tag{26}$$

where $C(f)$ defines the imbalance of pixel numbers with regard to features [89]. Every time the cut point shifts, the amount of balance is updated by the joint-histogram.

4.4 Experimental Results

To show the performance of our proposed depth estimation method we evaluated different aspects and different situations, such as images with shadow and shading, low-texture images, and challenging images with lots of light (bright images) and occluded pixels.

4.4.1. Datasets

We tested our method with three different databases on several images: one synthetic database and two real world image databases. We performed a wide variety of tests using synthetic images created by Wanner *et al*. [90] with depth maps from prior algorithms and ground truth depth maps as a part of the database. For the real-world image databases, we used the real-world LF database from JPEG (JPEG Pleno Database) [91] which includes the result of their depth map estimation and a third database that we captured using a Lytro Illum camera.

4.4.2. Methods compared

To evaluate the accuracy of our method, we compared our result with those of Jeon *et al.* [50], Wang *et al*. [52] and Wanner *et al*. [92]. For [50] and [52] we used source code supplied from the authors and implemented the code in MATLAB. For the method of Wanner, source code was not available. However; the results of their algorithm were supplied with their database [90] and we compared our results against those supplied. The results confirm that our method is faster than the method of Jeon *et al*.[50] and Wang *et al*. [52] which is shown in Table 1.

4.4.3. Evaluation methods

For estimating the error of our method, we compared the results of our method in terms of MSE as shown in Eq. (27) for light field imagery with ground truth depth maps from the Wanner synthetic database [90] as shown in
Table 2. For calculating MSE we used the Eq. (27) from [93] :

$$MSE_N = (\sum_{i \in N}(d(i) - gt(i))^2)/(N * 100) \qquad (27)$$

where, gt is the ground truth, and d is the depth map which is estimated. N is the number of pixels in the depth map. We divided by 100 to convert MSE into a percentage. We understand that this is a non-standard processing of MSE values, but this method was used in [93] and we have taken this approach here to make our work comparable with other works [4,49,93].

Table 1. The time to compute the depth map calculated on a CORE i5-6300U CPU @2.40GHz , 8 GB RAM running MATLAB R2018a.

Image	Jeon *et al.*[50]	Wang *et al.*[52]	Our method
Papillion	49.98min	16.59 min	**14.95 min**
Buddha	44.65min	14.27 min	**11.81 min**
Horses	45.98min	14.89min	**12.16min**

Table 2. The MSE error of the estimated depth map compared with ground truth (%).

Image	Wanner *et al.*[92]	Chen *et al.*[4]	Jeon *et al.*[50]	Zhang *et al.*[94]	Wang *et al.*[15]	Our method
Papillion	24.44	12.83	36.23	7.50	17.9	**5.91**
Buddha	15.01	8.34	28.90	7.99	13.524	**5.70**
Cube	13.29	9.72	26.03	9.96	11.46	**9.02**

4.4.4. Depth Maps for Synthetic Images

We used the dataset of Wanner et al. [90] for evaluating our method based on synthetic images which have less noise and are closer to ideal multi-view images [6]. The MSE values between the ground truth depth maps and our estimation of the depth maps are illustrated in

Table 2. The MSE values for [4, 15, 92] were collected from [6]. For calculating the MSE we compared the result of the depth map with ground truth provided by Wanner [90]. The detailed comparisons on depth maps with ground truth for two images are shown in Figure 15. Furthermore, Figure 14 shows an example of our result compared with the method of Jeon et al. [50], Wang et al. [52] and Wanner et al. [90]. It can be seen that our method compared to Wang et al. [52] can better handle noise by using a fast weighted median filter, the method of Jeon et al. [50], doesn't work well for synthetic images and comparing with Wanner et al. [90], we obtain a better result. In particular, our results demonstrate clearer edges in regions of occluded pixels. Note that the ground-truth depth maps in this figure are represented as the inverse of the depth maps in Figure 14; we have inverted our depth map to allow a direct comparison.

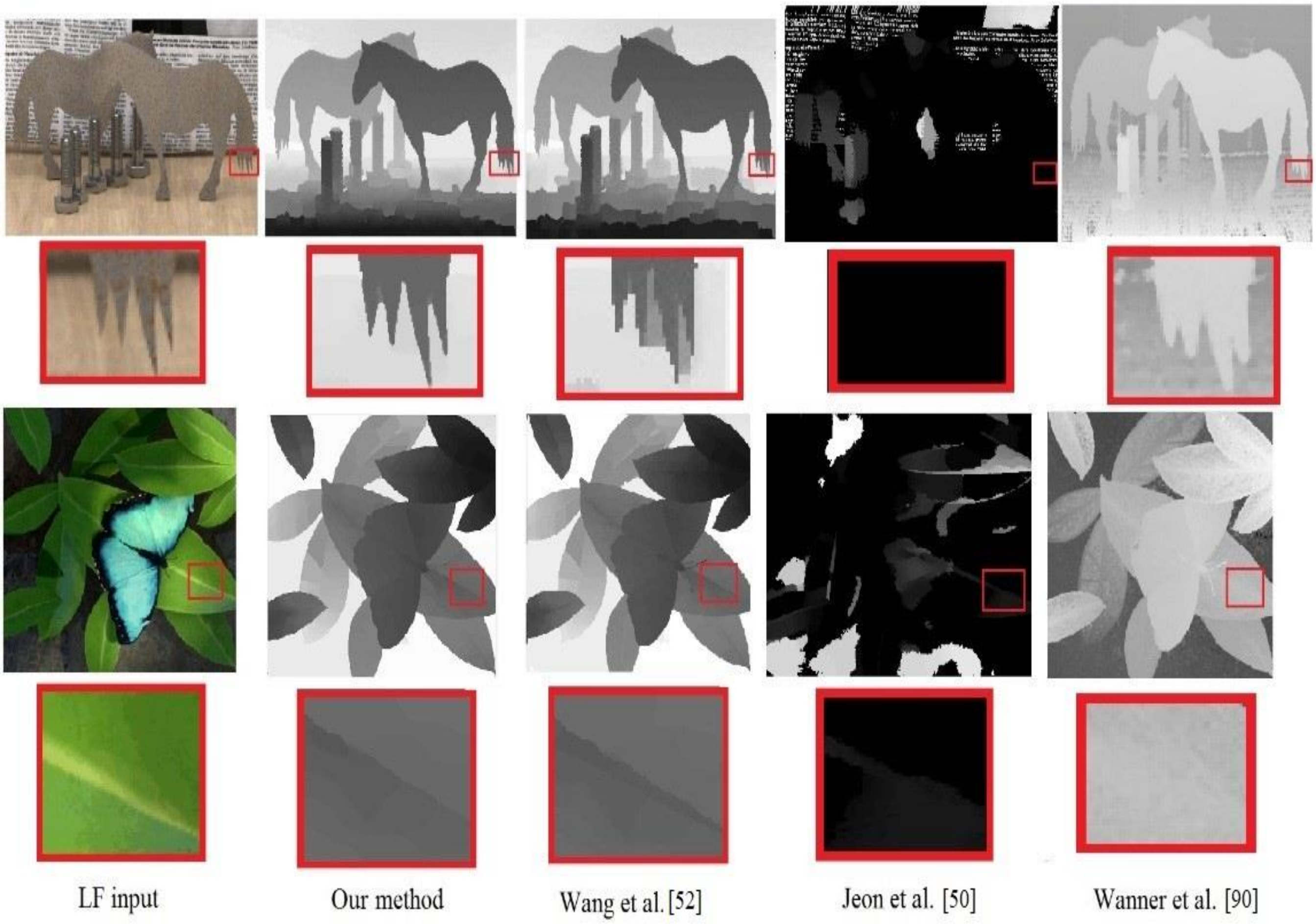


Figure 14. Comparing our approach result with the results of other approaches for two synthetic images.

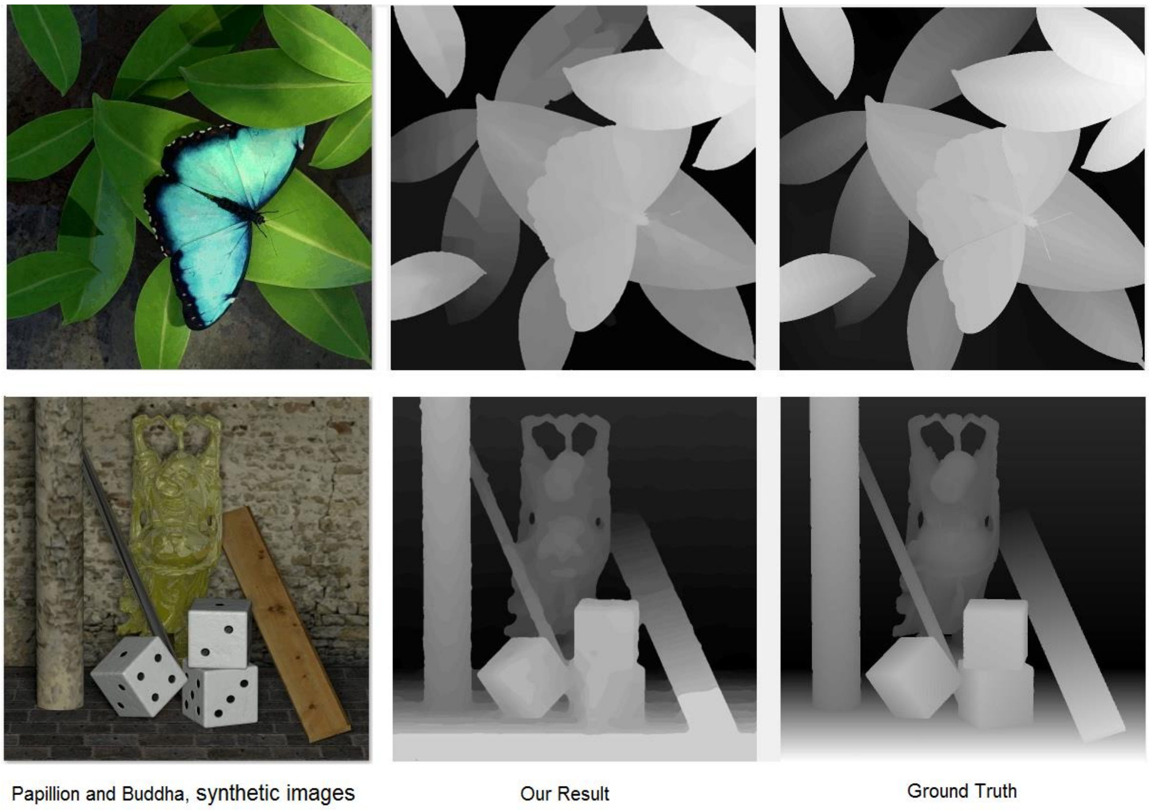


Figure 15. Comparing the results of our approach with ground truth on the synthetic images "Papillion" and "Buddha".

Some synthetic light images such as Figure 14 (butterfly and leaves) have the problem of synthetic shadows which appear incorrectly in the associated ground truth data. This makes comparison of our algorithm to the ground truth for this data to be problematic as the shadows are an artefact of the synthetic data, and do not reflect a real light field image. We have performed another experiment on real images with natural shadows to show the ability of our proposed method to handle real world shadowing. Figure 16 shows an image captured by a Lytro camera, and shows the step by step results of our proposed method. As explained above, the four main steps in our work for estimating depth maps are: edge and occlusion detection, sub-aperture image match computation, defocus cue computation and filtering. Each step has an important role in the accurate estimation of the depth map. Without the edge and occlusion detection step, we cannot overcome the problem of occlusion between the sub-aperture images, especially in challenging lighting conditions. Sub-aperture image matching and defocus cue computation complement each other to obtain a more accurate depth map, even though each of them has inherent limitations. The combination of these approaches is thus critical to obtaining better results. Filtering has an important role in noise removal, especially for converting the depth map to a 3D point cloud, and this step enables significant improvement in the result.

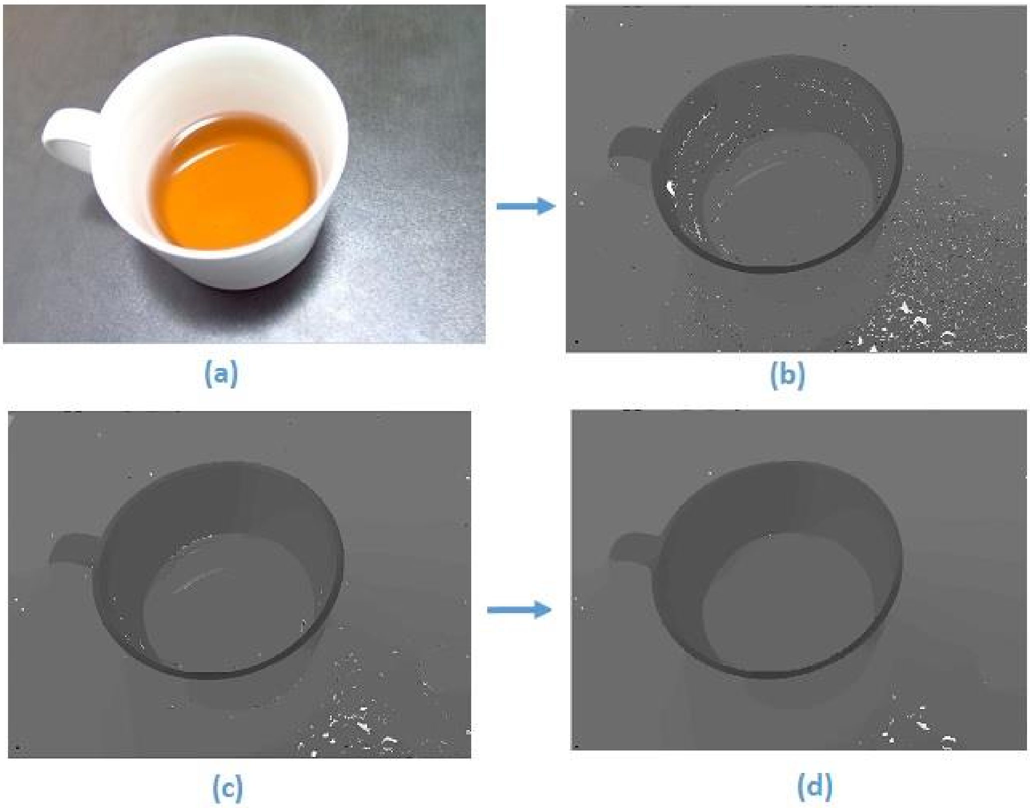
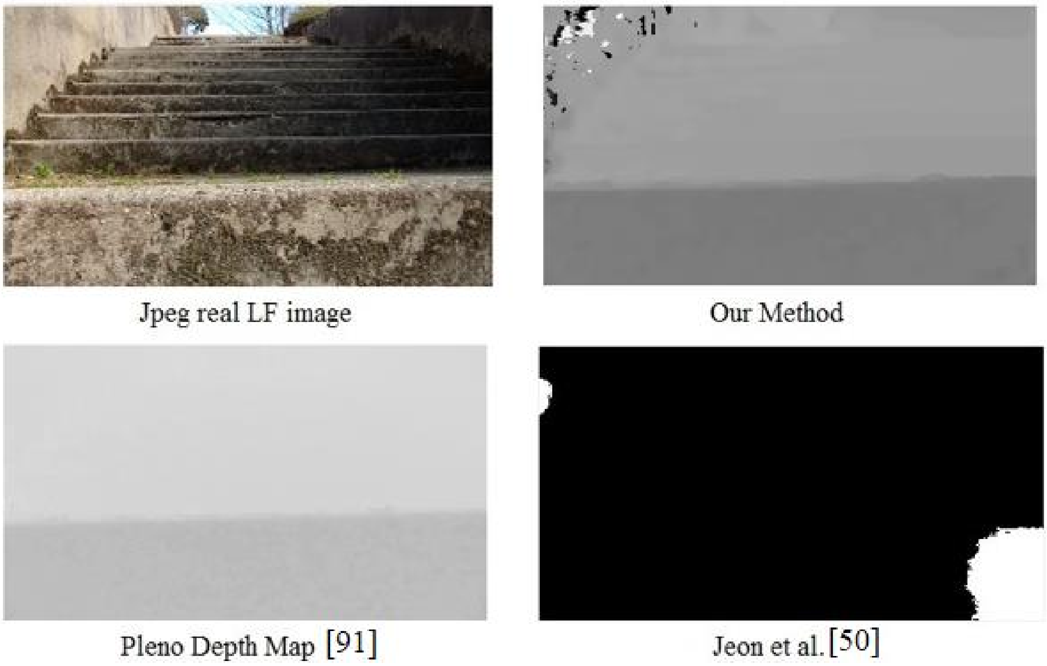


Figure 16: Progression of our proposed method to estimate a final depth map. (a) shows the light field image captured by ourselves with a Lytro camera. (b) shows the initial depth map estimation based only on sub-aperture matching. (c) shows the combined depth map estimation including defocus cues. (d) illustrates the final depth map after applying normalization and filtering.

4.4.5. Depth Maps for Real Images

In this section, our method is evaluated using real images captured by the Lytro Illum camera. These images are noisier than synthetic images. We have used two different datasets of real images, one collected by JPEG and one collected by us. Figure17 and Figure 18 show the result using real images from the JPEG dataset [91]. It is clear that our result has greater accuracy compared to other methods. If we observe the background of the image in Figure 18 for instance, in our result the depth variation of the grass is apparent, however in the Jeon et al. [50] result the background of the image is totally black and compared to Wang's result our result has less noise on sharp edges in the image.

Figure17 illustrates an image with very challenging light conditions from the same dataset. To further evaluate our method, we prepared a dataset by capturing images using a Lytro Illum camera. Two different samples of our images are shown in Figure 19 and Figure 10. It is clear from the Figure 19, the proposed method is more robust to noise by the using fast weighed median filter and also is able to maintain more image details, especially for scenes with occlusions.


Jpeg real LF image

Our Method

Pleno Depth Map [91]

Jeon et al. [50]

Figure 17.Depth estimates on real images from the JPEG dataset with very challenging conditions (lots of shadow and low-light areas).

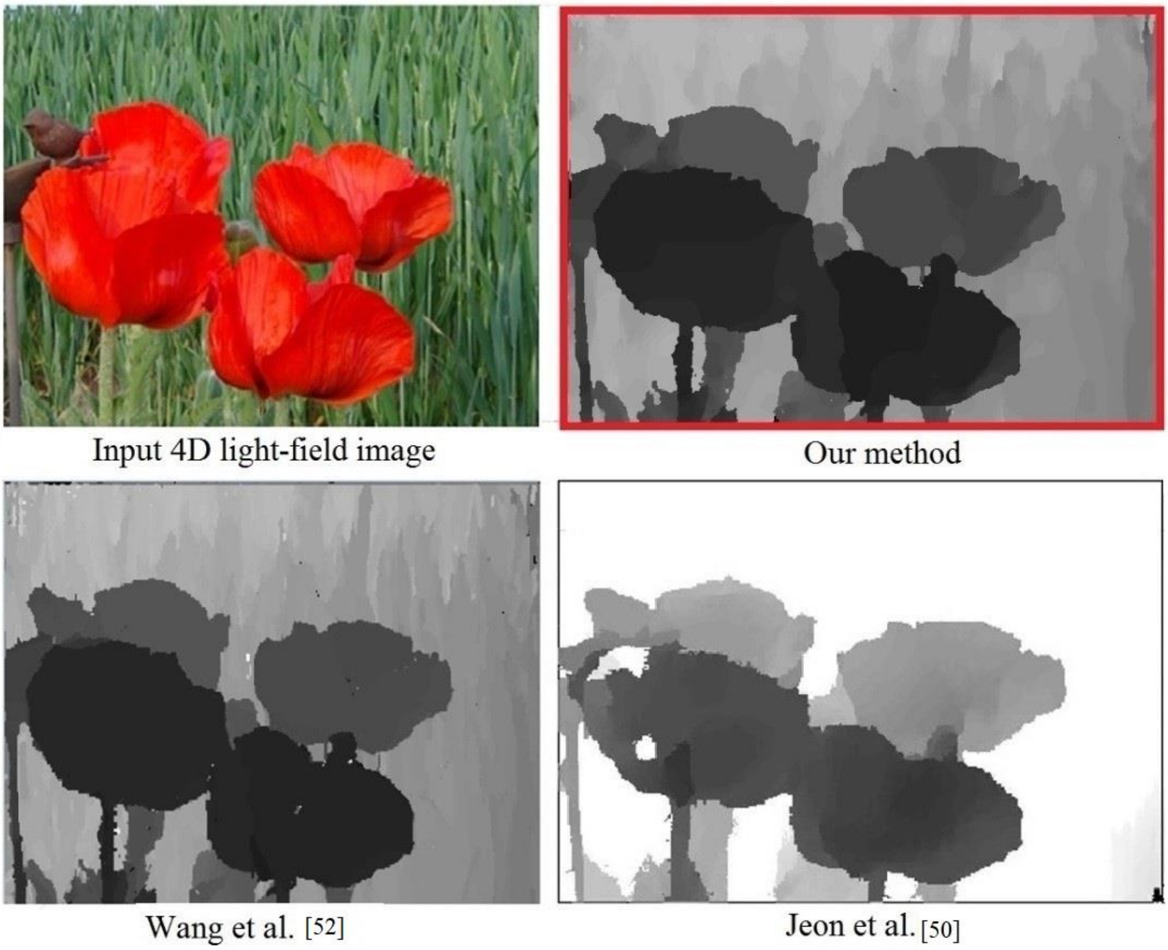


Figure 18. Comparison of depth estimation results of different methods from a 4D light-field input image. The top-left image shows a single sub-aperture image from a LF dataset. The top-right image shows the result of our depth-map estimation algorithm, while the bottom two images show the results of two prior art depth map estimation algorithms.

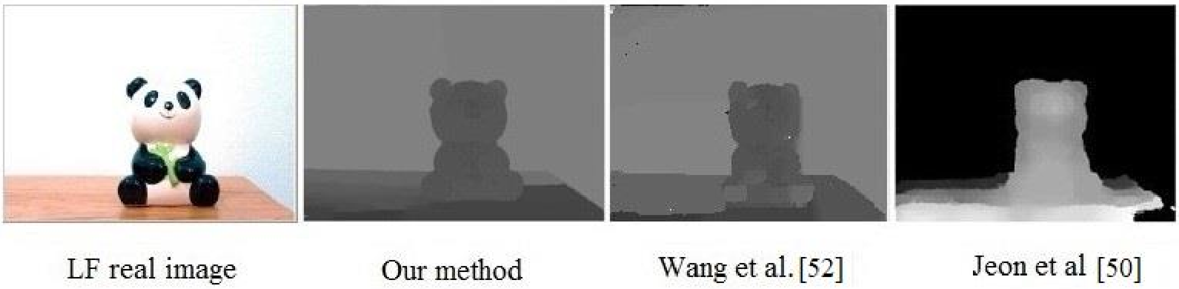


Figure 19. Depth estimation on a real image captured with a Lytro Illum camera.

Chapter 5: Proposed 3D Point Cloud Approach

In this section, we present the proposed method for creating a 3D point cloud based on the proposed depth map. For the reconstruction of a 3D point cloud of an object, we develop a new method that is based on the transformation of the point-plane correspondences. The input of our system is a 4D LF image which can be captured by light field cameras, such as the Lytro [12] or created synthetically. As a first step we need the estimated depth map, which we discussed in Chapter 4. As having a densely sampled depth map is essential for creating a 3D point cloud, in this chapter we enhance the depth map substantially by applying histogram equalization and histogram stretching followed by adding edge detection information from the original image. This kind of enhancement after estimation of the depth map is one of the significant contributions of our point cloud reconstruction work. These steps also can increase the distance between adjacent depth layers. This kind of increase between the layers can contribute to deduction of errors for estimation of depth map. Therefore, we linearly combined the resultant stretched depth map with Canny edge detection results and then transformed the point-plane correspondences to acquire a 3D point cloud. This operation enhances fine detail in the depth map by leveraging the sharp information in the edge image. We used a Lytro Illum camera in this work. After comparing different types of LF formats we decided to use the 4D LF format as an input image for more consistency and reliability.

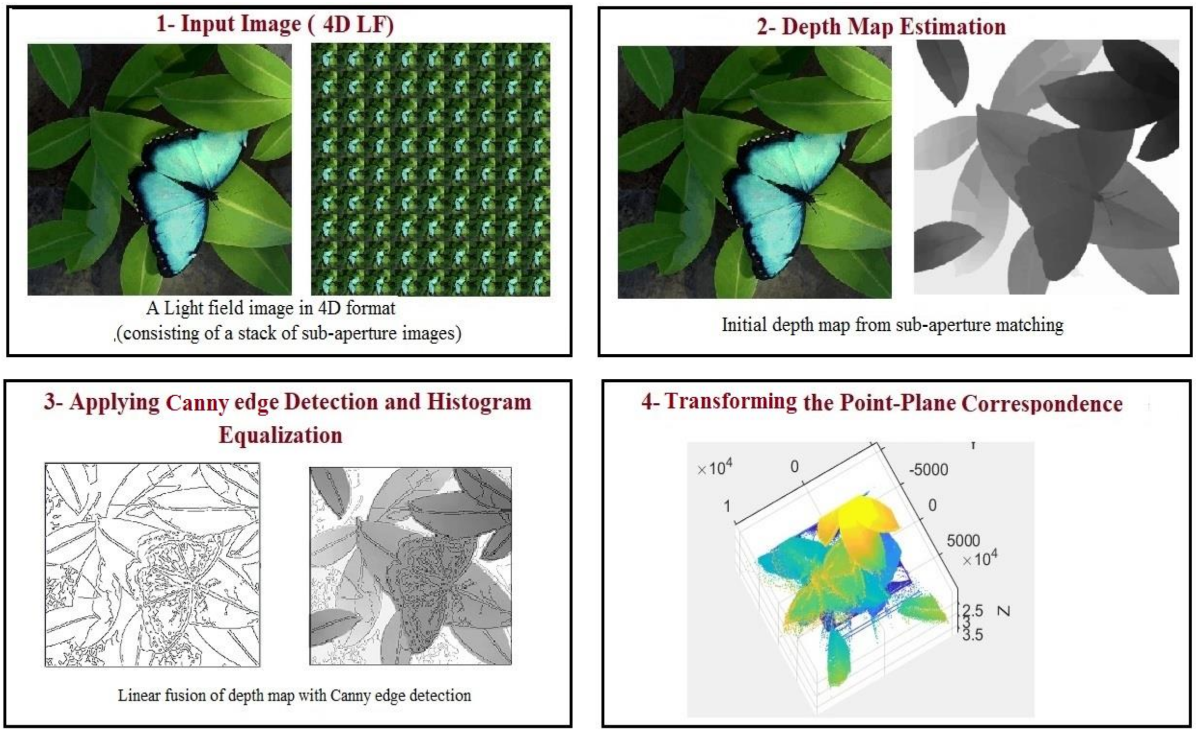

1- Input Image (4D LF)
A Light field image in 4D format
,(consisting of a stack of sub-aperture images)
2- Depth Map Estimation
Initial depth map from sub-aperture matching
3- Applying Canny edge Detection and Histogram Equalization
Linear fusion of depth map with Canny edge detection
4- Transforming the Point-Plane Correspondence
×10^4
0
-5000
1
0
5000
×10^4
2.5
3
3.5
Z

Figure 20. Overview of the proposed method. Our method consists of 4 main stages: 1. 4D format of light field image as input images, 2. Depth map estimation based on using our proposed method for depth map, 3. Canny edge detection results are combined linearly with the histogram stretched depth map, 4. Transforming the point plane correspondences to create a final 3D point cloud.

Our approach is based on transforming the point-plane correspondences and in the first step we need the depth map obtained from the method we discussed in Chapter 4. Estimating an accurate depth map is essential to having a complete 3D point cloud. Following this step, we will enhance the depth map by histogram stretching and edge detection and then by transforming the data we will acquire the 3D point cloud.

5.1 Histogram Equalization and Canny Edge Detection

After obtaining the 2D depth map image from the method of Chapter 4, $DM(u, v)$, in order to increase the distance between adjacent depth layers we will enhance the depth map by histogram equalization. In the first step we use histogram stretching to improve the separation between the depth planes.

$$HS_{uv} = \frac{pi(u,v) - DM_{min}}{DM_{max} - DM_{min}} * 255 \tag{28}$$

where $pi(u, v)$ is the pixel intensity at (u, v) in Image I, DM_{uv} is the 2D depth map image with a minimum value denoted as DM_{min} and a maximum value denoted as DM_{max} and HS_{uv} is the histogram stretched depth map.

Histogram stretching can increase the contrast in the 2D depth map image and gives us more detailed information to emphasize depth structure of the scene across different layers in the depth map. In the second step, Canny edge detection is applied to the input image $I(u, v)$. The main aim of the Canny operator is to utilize the first derivative of a Gaussian in different directions as a filter of noise and then on the filtered image the maximum value of the local gradient is calculated to determine image edges [95].

For a smoothing filter, the Canny filter uses a Gaussian filter as denoted below:

$$G(u, v) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{u^2+v^2}{2\sigma^2}\right) \tag{29}$$

which is applied to the input image $I(u, v)$ by convolution.

$$\acute{I}(u,v) = G(u,v) * I(u,v) \tag{30}$$

Following this, for the purpose of detecting the edges, the value of the local gradient and direction of image are calculated.

$$I_1(u,v) = (\acute{I}(u,v+1) - \acute{I}(u,v) + \acute{I}(u+1,v+1) - \acute{I}(u+1,v))/2$$

$$I_2(u,v) = (\acute{I}(u,v) - \acute{I}(u+1,v) + \acute{I}(u,v+1) - \acute{I}(u+1,v+1))/2 \tag{31}$$

$$CA_{uv} = \sqrt{I_1(u,v)^2 + I_2(u,v)^2}$$

$$\theta(u,v) = \arctan\,(I_1(u,v)/I_2(u,v))$$

where $\theta(u,v)$ is the direction of gradient and CA_{uv} is the edge image.

In the third step, the histogram stretched image and the edge image are combined linearly (image fusion), and the result is saved in T_{uv}.

$$T_{uv} = |HS_{uv} + CA_{uv}| \tag{32}$$

The operation in (32) is performed to provide additional high frequency information to the depth map to enhance the sharpness of edges.

5.2 Creating a 3D Point Cloud by Transforming the Point-Plane Correspondences

For estimating a 3D point cloud we need to estimate T_z (the *z* component of each point at position *u,v*) from which a point cloud can be created by transforming the point-plane correspondences.

For the points in the final point cloud we start by selecting *Tx* = *Tu* and *Ty* = *Tv*, then for computing T_z:

$$T_z = \frac{b * fl * fc}{T_{xy} * fc * b * \max{(T_{xy})}} \tag{33}$$

where b is amount of the baseline, fl is the focal length and fc is the focus distance. fl is an intrinsic parameter and depends on the captured image. b and fc are extrinsic parameters and depend on the type of camera. T_{xy} is the modified depth map. The *x* and *y* coordinates of the point are then given by:

$$\acute{\mathrm{T}}_x = \frac{\mathrm{T}_x * T_z}{Se}$$

$$\acute{T}_y = \frac{\mathrm{T}_y * T_z}{Se} \tag{34}$$

where Se is the sensor size (mm). We then denote the estimation of 3D point cloud by T_{xyz} :

$$T_{xyz} = (\acute{T}_x\,, \acute{T}_y, T_z) \tag{35}$$

5.3 Experimental Results

We evaluated our method with two different databases and compared the proposed method with two state of the art methods, as described in further detail below. For estimating the error of proposed method we used the number of points as shown in Section 5.3.3.

5.3.1. Result on Databases

We used two different databases; one synthetic database and one real-world light field image database. For the synthetic database we utilized a database popular within the research community, which was created by Wanner *et al*. [90]. For the dataset of real images, we used a Lytro Illum to capture various scenes. To improve evaluation, we captured images in different situations, including images with shading, low-texture, and challenging images such as very bright images and images with occluded pixels. Our method is tested in wide variety of images of which a sample of results on synthetic images is shown in Figure 20 and two other samples of real images captured ourselves are shown in Figure 21 and Figure 22.

5.3.2. Methods Compared

To illustrate the accuracy of our proposed method, we compared our result with those Perra *et al*. [10] and Dansereau *et al*. [96]. Perra *et al*. [10], produced 3D point clouds based on the depth map that is created by the software supplied with the Lytro camera, however, this software does not produce an accurate depth map in all situations. As shown in Figure 22, we compared the depth map output by the Lytro software with our depth map. We re-implemented Perra *et al*'s method in Matlab to compare its performance with our result.

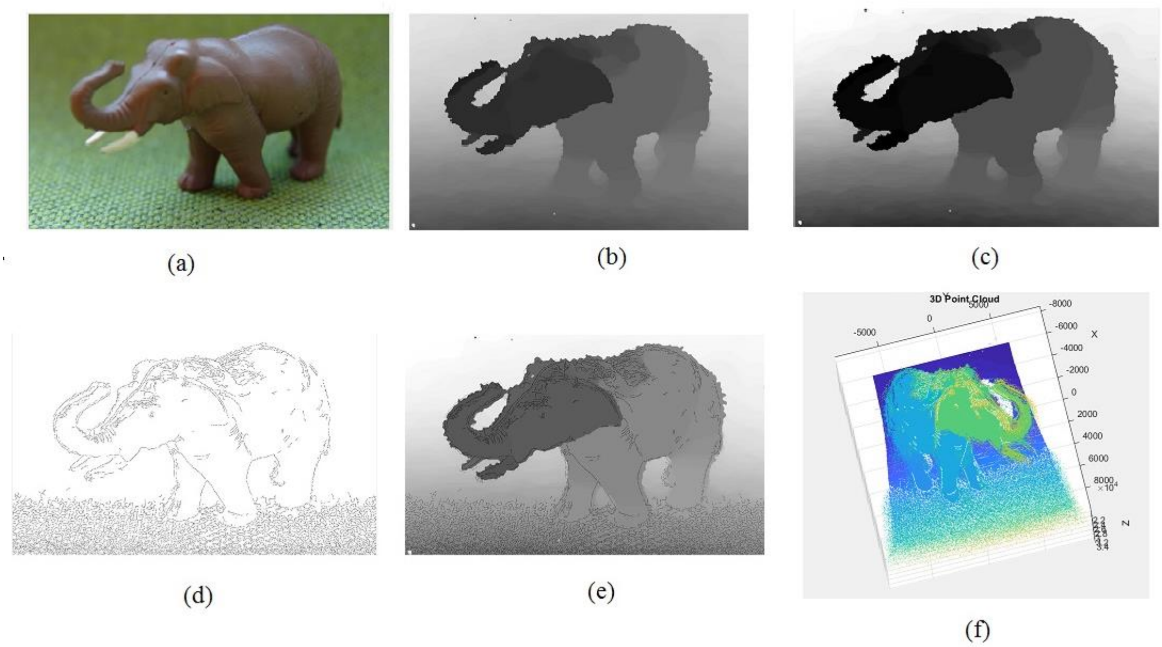


Figure 21. A step by step illustration of our methodology for creating 3D point clouds. (a) Input light field image. (b) Depth map based on our approach. (c) Result of histogram stretching applied to the depth map. (d) The results of applying Canny edge detection to the central sub-aperture image. (e) Image fusion (combining (c) and (d) linearly). (f) 3D point cloud.

For Dansereau *et al*. [96] we also re-implemented just the 3D reconstruction part of their method in Matlab and evaluated their result against ours. The comparison on a very challenging image is shown in Figure 22. The image is captured in low-light conditions and also has shadowed areas. However, it is clear that our result is more accurate compared to other methods.

5.3.3. Evaluation Methods

For evaluating the performance of the 3D point cloud reconstruction algorithms, one important factor is number of points [97]. We calculated the number of points in our point cloud for comparison against two other methods. Table 3 shows the result of this comparison.

Table 3. Comparing numbers of points in synthetic (Papillon) and real (Cat) images.

Light field image	**Perra *et al.*[10]**	**Dansereau *et al.*[96].**	**Our method**
Papillon	57860	43650	**69523**
Cat	86230	74253	**102365**

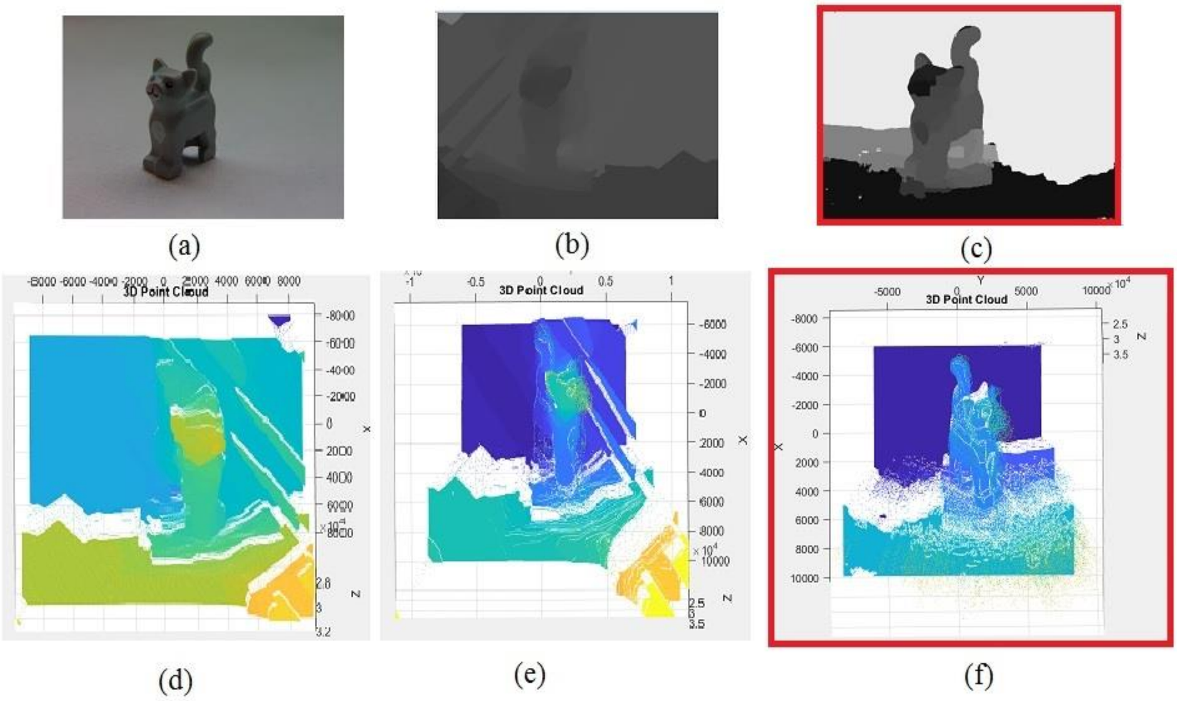


Figure 22. Comparing our result with other state-of-the-art methods. (a) Light field input image. (b) Depth map created by the Lytro software. (c) Depth map that is produced by our method. (d) 3D point cloud obtained by re-implementing the method of Dansereau et al. [95]. (e) 3D point cloud obtained by re-implementing the method of Perra et al. ***[10]****. (f) The result of our 3D point cloud reconstruction algorithm.*

Chapter 6: Proposed Enhanced 3D Model Method

In this work, we used a LF image and its depth map as an input image to generate an enhanced 3D point cloud. Our approach is based on transforming the point-plane correspondences on an enhanced depth map. Since having an accurate depth map is of paramount importance to generate a complete 3D point cloud, we improve the depth map in two different steps. In the first step we apply histogram stretching and equalization on the original depth map. In the second step of enhancement we intelligently add edge detection information to the equalized depth map, then we acquire the 3D point cloud by coordinate transformation.

6.1 Histogram Stretching and Equalization

In the first step of enhancement we apply histogram stretching and equalization on the original depth map. Our input is an 8-bit grayscale 2D depth map image, $DM(u, v)$. In order to increase the distance between adjacent depth layers, we apply histogram stretching and then on the result of histogram stretching, we apply histogram equalization. The equations below show the histogram stretching to improve the separation between the depth planes.

$$HS_{uv} = \frac{pi(u,v) - DM_{min}}{DM_{max} - DM_{min}} * 255 \tag{36}$$

where $pi(u, v)$ is the pixel intensity at (u, v) in Image I, DM_{uv} is the 2D depth map image with a minimum value denoted as DM_{min} and a maximum value denoted as DM_{max} and HS_{uv} is the histogram stretched depth map.

Then we apply histogram equalization on the result of the histogram stretching to spread the intensity values over the full range of the histogram image and for enhancing the contrast of the depth map [98].

Given the histogram stretched depth map HS_{uv}, if we consider r_k as the dynamic range of intensities in the depth map, then the probability based on the histogram $p(r_k)$ can be calculated as below:

$$p(r_k) = \frac{Total\ pixels\ with\ intensity\ r_k}{Total\ pixel\ in\ depth\ map\ HS_{uv}} \tag{37}$$

From this probability we can perform histogram equalization based on the below equation:

$$HQ_{uv} = \sum_{k=0}^{n} p(r_k) \tag{38}$$

where n is the number of pixels and HQ_{uv} is the result of histogram equalization.

6.2 Adding Edge Detection Information

The second step of enhancement is the most important part, where we intelligently add edge detection information. This is one of the novel aspects of this work. We developed a new strategy in this area where we acquire fuzzy logic based edge information from the result of histogram stretching on the depth map and feature matching from sub-aperture images and the original depth map, then combining these two results to obtain an intelligent edge detection. As a result, this kind of edge detection is more reliable compared to ordinary edge detection because noise is reduced compared to prior art edge detection methods such as Canny and Sobel. The details of this development is shown in Figure 23 and Figure 24 which also show the steps required for improvement of the depth map.

6.2.1. Fuzzy Logic

We found that a fuzzy logic approach can help us with detecting edges by comparing the intensity of neighbouring pixels and based on the gradient of the image we can find which pixels belong to an edge. This kind of information is very helpful for depth map images because the structure of levels in a depth map is based on the gradients.

We first obtain the image gradients based on the convolution of the image to acquire a matrix containing the u-axis and v-axis gradients of the depth map image.

For this purpose, we convolve the depth map HS_{uv} with gradient operator, G, using the convolution method. The gradient values are in the [-1 1] range.

$$Gx = [-1\ 1], Gy = Gx^{T} \tag{39}$$

$$G_{uv} = \sum_u \sum_v HS(u, v)G(x - u + 1, y - v + 1) \qquad (40)$$

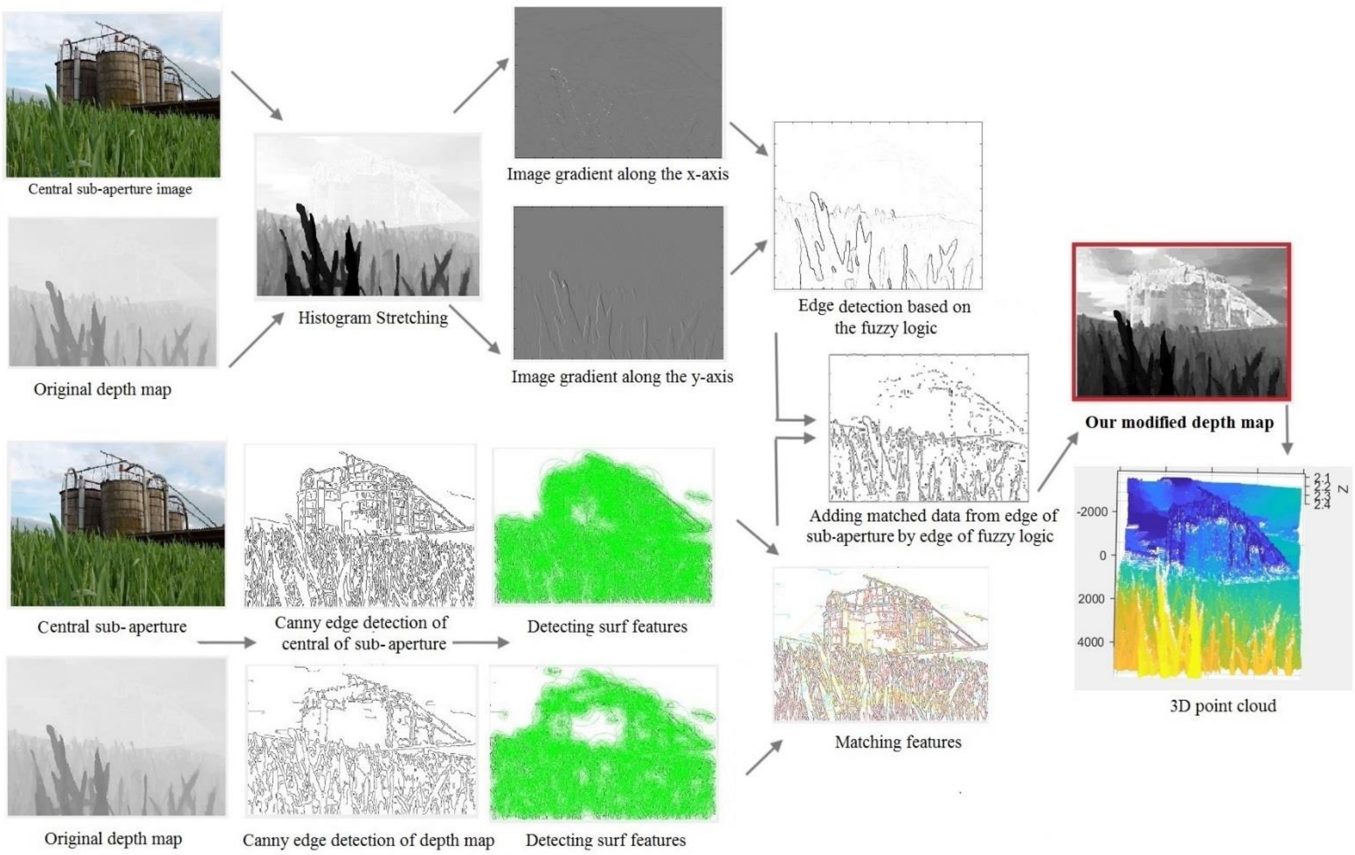
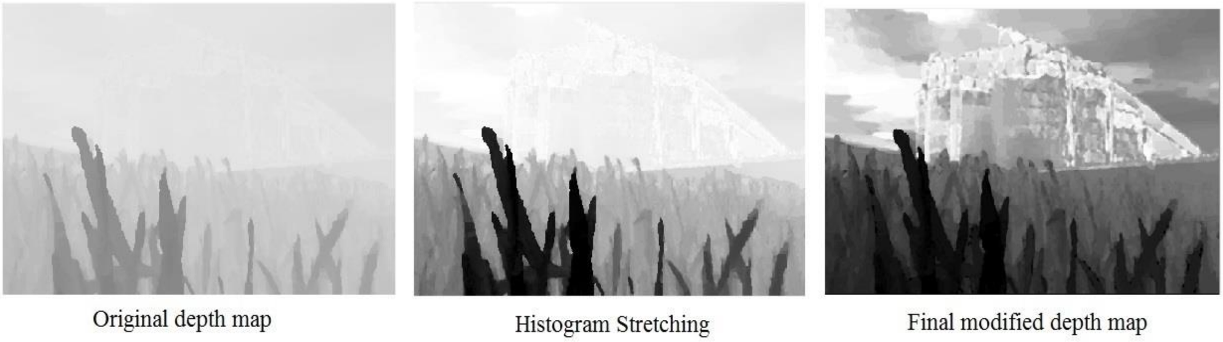

Central sub-aperture image
Original depth map
Histogram Stretching
Image gradient along the x-axis
Image gradient along the y-axis
Edge detection based on the fuzzy logic
Adding matched data from edge of sub-aperture by edge of fuzzy logic
Our modified depth map
-2000
0
2000
4000
2.1
2.2
2.3
2.4
Z
3D point cloud
Central sub-aperture
Canny edge detection of central of sub-aperture
Detecting surf features
Original depth map
Canny edge detection of depth map
Detecting surf features
Matching features

Figure 23.Details of our proposed 3D point cloud estimation method.



Original depth map

Histogram Stretching

Final modified depth map

Figure 24. Depth map modification steps.

Considering gradients of the depth map as an input, we will create a Fuzzy Inference System (FIS) for edge detection.

An FIS makes a decision based on whether a pixel belongs to an edge or not. Membership functions are needed to define a fuzzy system. We defined a Gaussian membership function for each input:

$$\mu_{uv} = Gaussian(G_{uv}) = e^{[-(G_{max}-G_{\mathrm{uv}})^2/2\sigma^2]} \tag{41}$$

where μ_{uv} is a Gaussian function, G_{max} , G_{uv} are the maximum and (uv)$^{\mathrm{th}}$ grey values respectively and σ is the standard deviation associated with the input variable.

F_{uv} defines the final pixel classification as edge or non-edge.

$$F_{uv} = \frac{\sum_{u,v} \partial_c \ \mu_{uv}(\partial_{uv}\ )}{\sum_{u,v} \mu_{uv}(\partial_{uv}\ )} \tag{42}$$

where ∂_{uv} are the fuzzy sets as a part of a fuzzy rule, similar to [99] and ∂_c is the output class centre of fuzzy rule. As a result, a final fuzzy edge is defined by F_{uv} where 0 indicates that the pixel is almost certainly not part of an edge and 1 indicates that the pixel is almost certainly part of an edge.

6.2.2. Feature Matching

In the second step of enhancement we also, in parallel, extract the SURF features of intermediate results CSA_{uv} (Canny edge detection for the central sub-aperture image) and CDE_{uv} (Canny edge detection of original depth map). Then we match the features and extract features with higher amplitude to add to the result of the edge detection using fuzzy logic.

For this purpose, we used the SURF detector for detecting features. The SURF detector extracts features based on the Hessian matrix, which is determined at any point $po = (u, v)$ and scale *σ=1.2* as the second order derivative of a Gaussian filter.

$$H_{approx}(po, \sigma) = \begin{bmatrix} D_{uu}(po, \sigma) & D_{uv}(po, \sigma) \\ D_{uv}(po, \sigma) & D_{vv}(po, \sigma) \end{bmatrix} \tag{43}$$

where $D_{uu}(po, \sigma), D_{uv}(po, \sigma), D_{vv}(po, \sigma)$ are the convolution of the Gaussian second order at the point $po = (u, v)$. This can be executed methodically if utilizing an integral image, as a result we calculate the integral image for those two input images:

$$CSA(u, v) = \sum_{0 \le i \le u} \sum_{0 \le j \le v} CSA(i, j)$$

(44)

$$CDE(u,v) = \sum_{0 \le i \le u} \sum_{0 \le j \le v} CDE(i,j)$$

Where i and j are defined by $0 \le i \le u$ and $0 \le j \le v$. The determinant of the Hessian matrix can be presented as follows

$$det(H_{approx}) = D_{uu}D_{vv} - (0.9D_{uv})^2 \quad (45)$$

Therefore, the interest points, which includes their locations and scales, will be detected in an approximate Gaussian scale space [100].

For matching features of those two images, we used the nearest neighbour method similar to [100]. In this way, image CSA_{uv} has n_1 directed line segments and image CDE_{uv} has n_2 directed line segments, the nearest neighbour pair can be obtained by defining matrix K as below:

$$K(i,j) = \begin{cases} 1 & CSA\ is\ the\ nearest\ neighbor\ of\ CDE, \\ 0 & otherwise \end{cases} \quad (46)$$

Then we determined the feature points with high amplitude from $K(i,j)$.

At the end of this step, we add the result of feature matching to the edge determined by fuzzy logic. As a result, we will have an intelligent edge detection that we denote as Int_{uv}.

$$Int_{uv} = |F_{uv}| + |K_{uv}| \tag{47}$$

The merging of the two sources of information (fuzzy logic result F_{uv} and nearest neighbor edge detection result K_{uv}) allows a more accurate edge detection. In the next step, we add this edge detection result to the equalized depth map, followed by the application of a median filter (3*3) on the result of adding, and the result is saved in T_{uv}.

$$T_{uv} = MedianFilter[\, Int_{uv} + \, HQ_{uv}] \tag{48}$$

6.3 Creating 3D point cloud by transforming the point-plane correspondences

For estimation of the 3D point cloud we need to estimate T_z (the *z* component of each point at position *u,v*) from which a point cloud can be created by transforming the point-plane (a similar process to Section 5.2).

For the points in the final point cloud we start by selecting $T_x = T_u$ and $T_y = T_v$, then for computing T_z:

$$T_z = \frac{b \, * fl * fc}{T_{xy} * fc * b * \max{(T_{xy})}} \tag{49}$$

where b is amount of the baseline, fl is the focal length and fc is the focus distance. fl is an intrinsic parameter and depends on the captured image. b and fc are extrinsic parameters and depend on the type of camera. T_{xy} is the modified depth map. The *x* and *y* coordinates of the point are then given by:

$$\acute{\mathrm{T}}_x = \frac{\mathrm{T}_x * T_z}{Se}$$

$$\acute{T}_y = \frac{\mathrm{T}_y * T_z}{Se} \tag{50}$$

where Se is the sensor size (mm). We then denote the estimation of 3D point cloud by T_{xyz} :

$$T_{xyz} = (\acute{T}_x\ , \acute{T}_y, T_z) \tag{51}$$

6.3 Experimental Results

We evaluated our method with three different databases and compared the proposed method with two state-of-the-art methods, as described in further detail below. For assessing the accuracy of our proposed method numerically, we used two different metrics: Histogram Analysis and LoD (Level of Details), as described in Section 6.3.3.

6.3.1. Result on Databases

We utilized three different databases: one synthetic database and two real-world light field image databases. We tested our method with several images. For the synthetic database, we used a database popularized by the research community, which was created by Honauer *et al.* [93]. For the real-world image databases, we used the real-world LF database from JPEG (JPEG Pleno Database) [91], which includes the result of their depth map estimation. We also used a third custom database of images acquired using a Lytro

Illum. For a more comprehensive evaluation, we captured images in different situations, including images with shading, low-texture, and challenging images such as very bright images and images with occluded pixels. Our method was tested on a wide variety of images, of which, a sample of results on the synthetic light field images is shown in Figure 2. Figure 24 and Figure 25 are based on the JPEG database. Figure 26 shows the result for images we captured using a Lytro Illum camera. Figure 27 is another sample from the JPEG database.

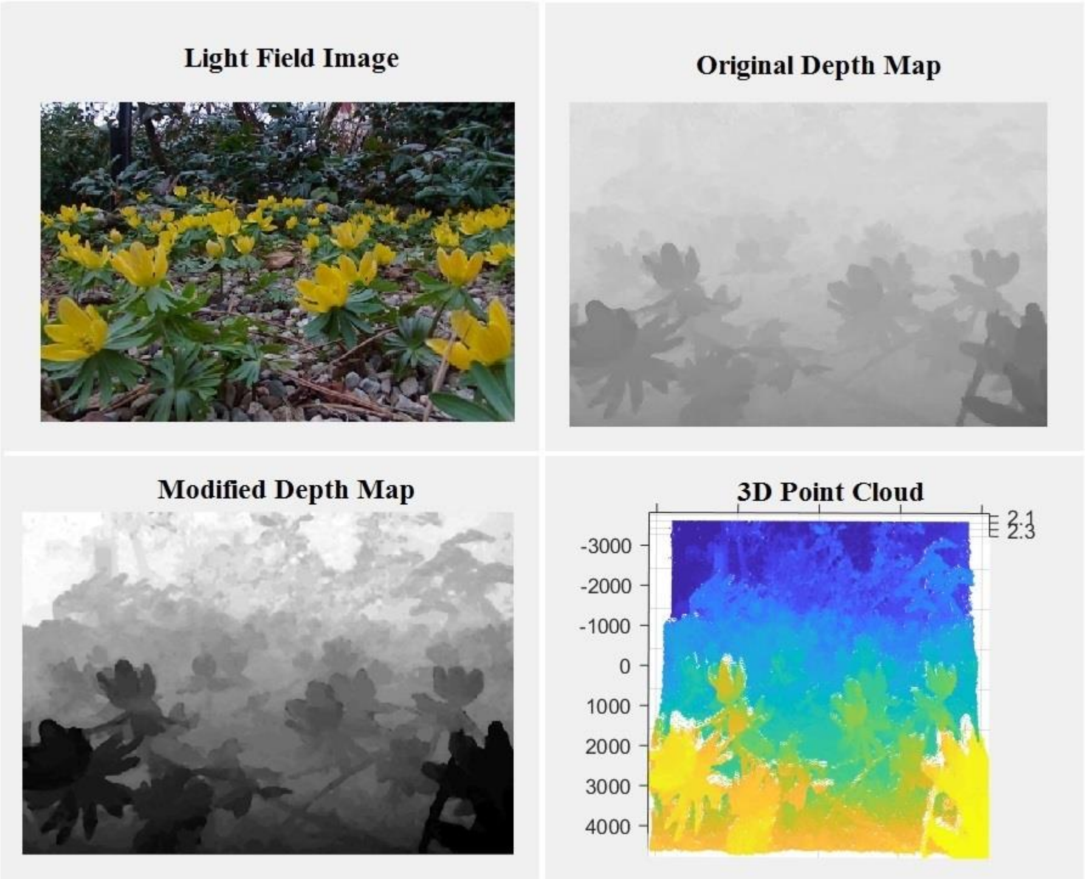
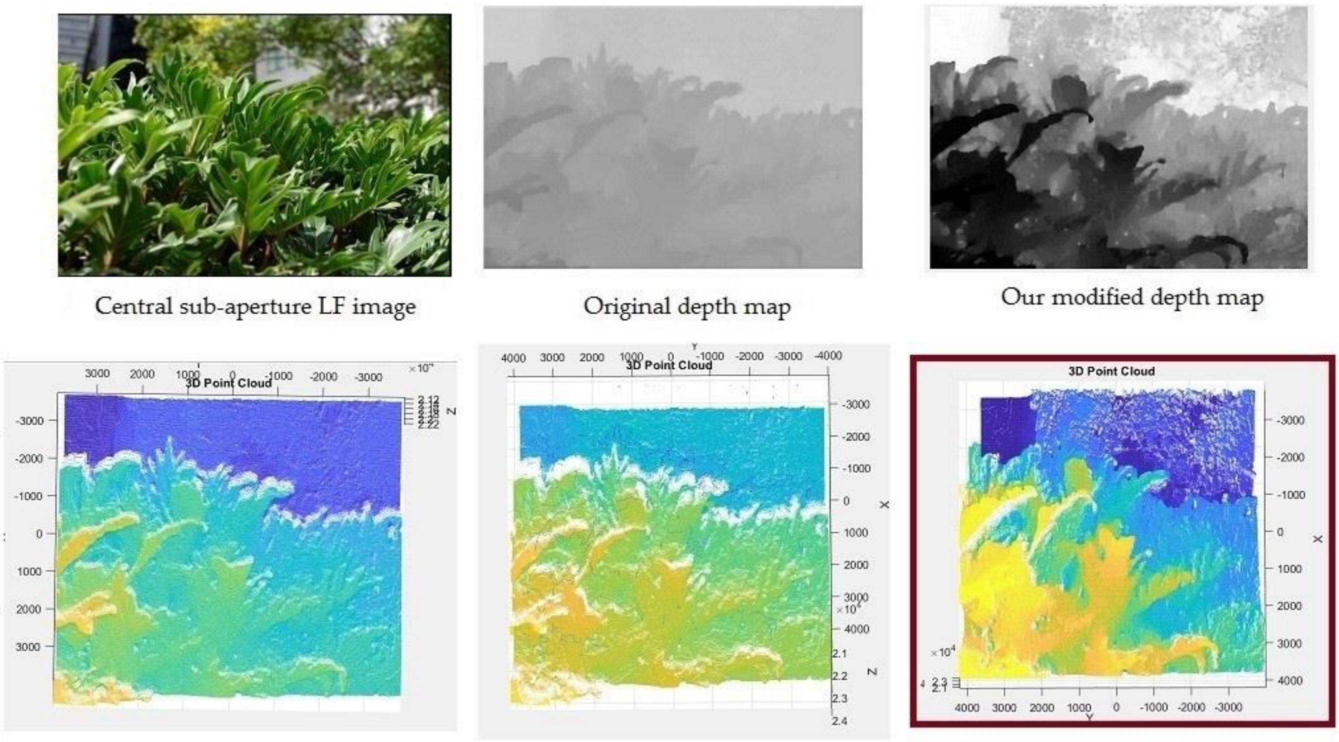


Figure 25. Obtaining a 3D point cloud based on a real-world light field image (Nature-Flowers) from the JPEG Pleno database.

6.3.2. Methods Compared

To illustrate the accuracy of our proposed method, we compared our result with the methods of Perra *et al*. [10] and Dansereau *et al*. [96] for creating 3D point clouds based on depth maps. As shown in Figure 26 and Figure 27, we compared the output of our method with their data. We re-implemented the method of Perra *et al*. in MATLAB to compare its performance with our result. As seen in Figure 26, because Perra *et al*. used Sobel edge detection for modifying their depth map, this kind of edge detection will cause noise that appears as blue pixels in the image. For Dansereau *et al*. [96], we also re-implemented just the 3D reconstruction part of their method in Matlab and evaluated their result against ours. Another comparison on a challenging image is shown in Figure 27, where we captured this image with a Lytro Illum camera.



Central sub-aperture LF image

Original depth map

Our modified depth map

3D point cloud of the method of Dansereau et al. [96]

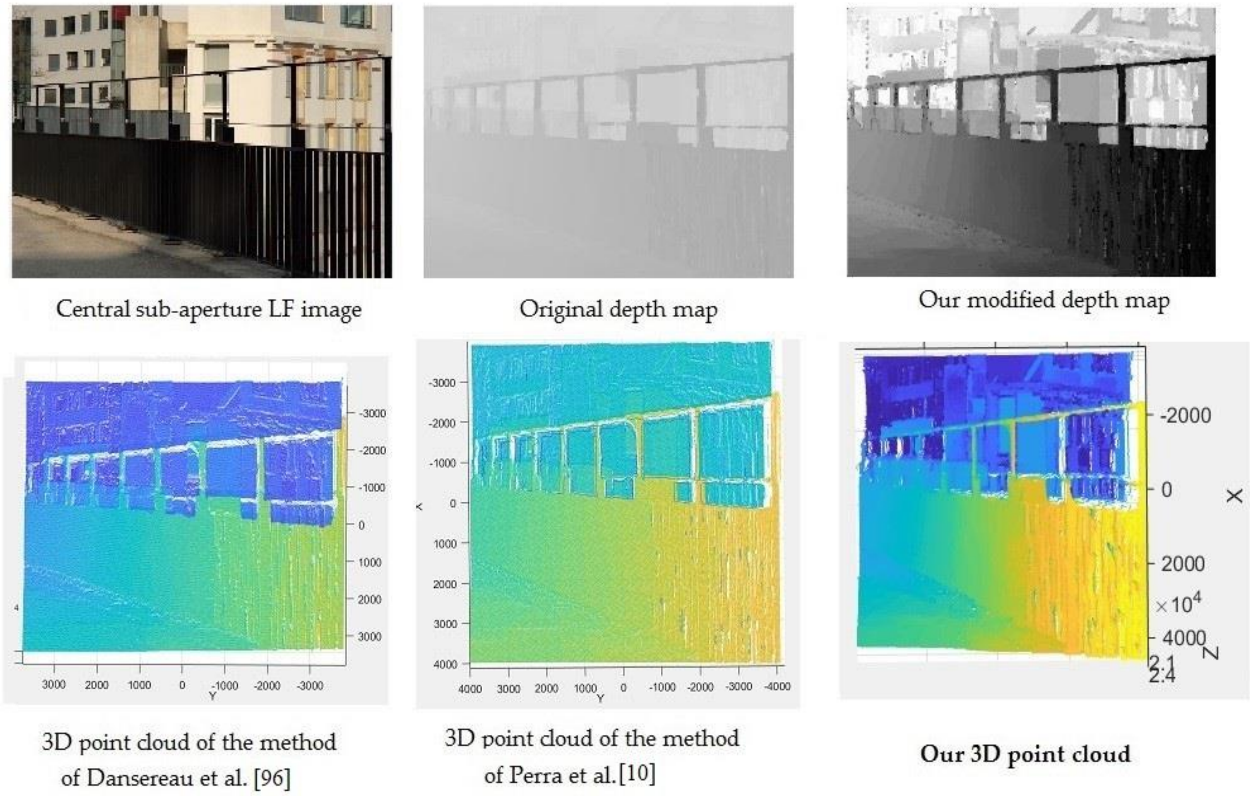
3D point cloud of the method of Perra et al. [10]

Our 3D point cloud

Figure 26. Comparing our result with other state-of-the-art methods. This input light field image is captured by ourselves with a Lytro Illum camera.

6.3.3. Evaluation Methods

For evaluating the performance of 3D point cloud algorithms, we used two different metrics: Histogram Analysis and Level of Details (LoD). These metrics assess the accuracy of a 3D point cloud by considering two important factors—density and distribution [101]. In this way, by analysing the histogram we can assess the range of distance values of the 3D point cloud distribution and by measuring LoD, we can evaluate the range of densities. We describe each part below:



Central sub-aperture LF image

Original depth map

Our modified depth map

3D point cloud of the method of Dansereau et al. [96]

3D point cloud of the method of Perra et al.[10]

Our 3D point cloud

Figure 27. Comparing our result with other state-of-the-art methods. This input light field image is a real-world LF image (Buildings-Black fence) from the JPEG Pleno database.

Histogram Analysis: One significant factor for assessing the accuracy of the 3D point cloud is evaluating the distribution of positions of pixels in the 3D point cloud by analysing the histogram [101, 102]. A histogram of an image is a plot that indicates the distribution of intensities in an image. For a point cloud, this concept can be extended to indicate the distribution of positions of points in the point cloud. Ideally, for a dense, natural scene, the histogram of positions should have an even, flat distribution, indicating details evenly distributed across all depths and directions. To compute histogram statistics, we calculate the histogram of our 3D point cloud as well as histograms of two other state-of-the-art methods. To increase the accuracy of the evaluation, we measured the histogram of the 3D point cloud based on each dimension (*X, Y* and *Z*) along with the mean and standard deviation. Figure 28 shows the histogram of 3D point clouds for the three different methods for a light field image that is captured by a Lytro camera (Green. Figure 26). These histograms indicate the number of pixels in the image at each different value of position relative to the centre of the viewpoint. In Figure 28 , the first part (a) corresponds to the histograms that are obtained from our 3D point cloud, and it is clear that the range of positions of our 3D point cloud is higher and more evenly distributed than the other methods compared, especially in the *Z* dimension.

In Figure 28 b, the histograms from Perra *et al*. [10] are shown, which have a more even distribution of positions compare to Dansereau *et al*. [96]. As a result, based on the histogram analysis, we can confirm that in terms of distribution, our 3D point cloud provides a more favourable distribution of positions compared to the two other methods.

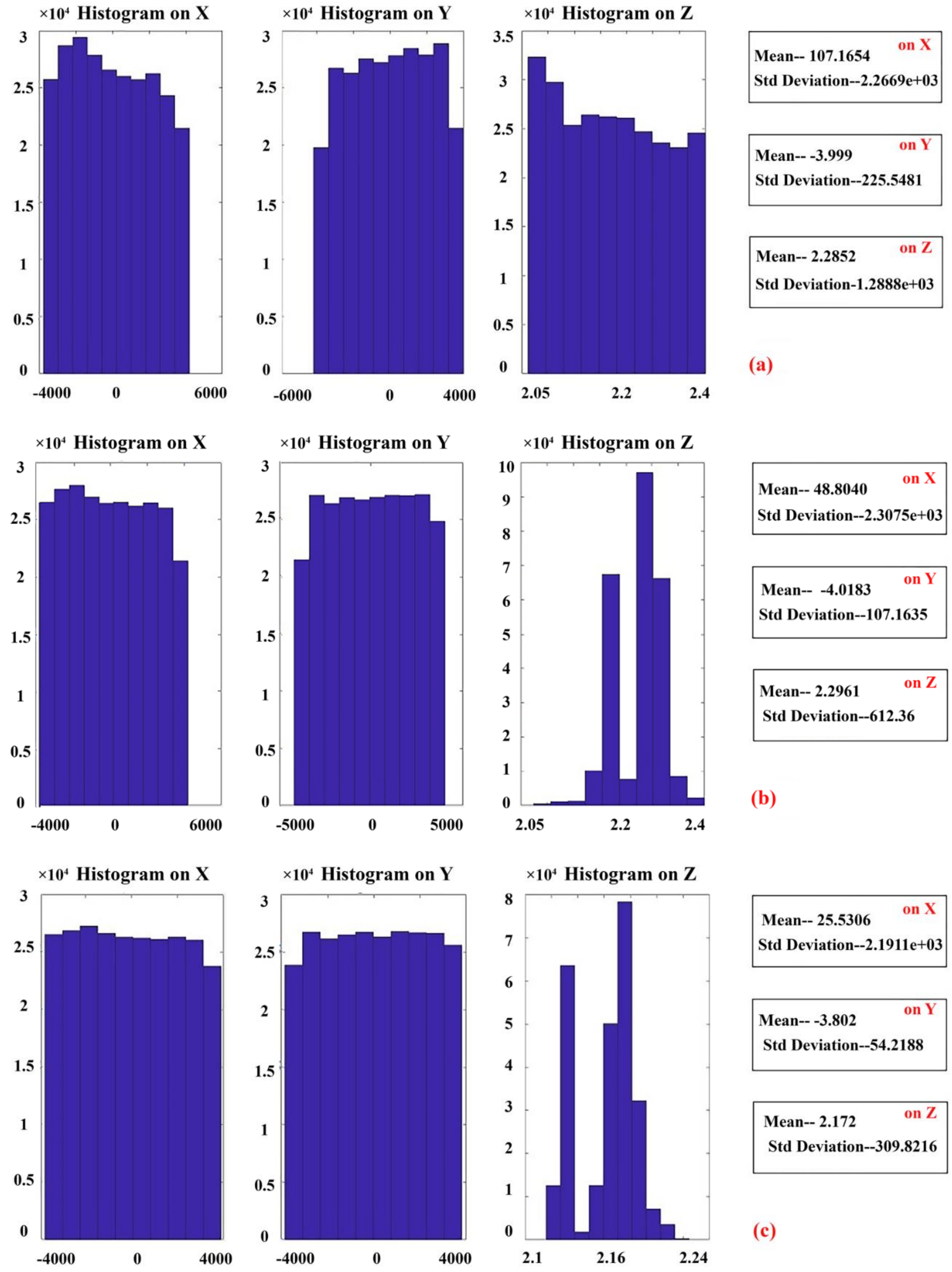


Figure 28. Comparing the histogram of our result with other state-of-the-art methods for image Green (Figure 27). (a) Shows the histograms obtained from our 3D point cloud method (ideal uniform distribution. (b) Indicates the histograms of Perra et al. [10] method for creating 3D point cloud and (c) shows the result of Dansereau et al. [96]. From the figure it is obvious that the histograms of our method (a) have a more favourable distribution of positions, especially in Z dimension, compared to the two other methods.

Level of Details (LoD): One of the significant factors for evaluating the density of a 3D model is measuring the level of detail [103]. In computer graphics, the level of detail is defined as the number of vertices (or faces) that generate an object. The level of detail influences the density of the 3D model [103]. Therefore, having a higher number of vertices or surfaces means having higher density, which indicates a more complex and potentially informative surface [10]. Moreover, sometimes level of detail is utilized to indicate the number of needed polygons for describing an object. The information about the LoD will be obtained from the 3D mesh. For this purpose, we write the 3D point cloud to PLY (Polygon File Format) format for the calculation of mesh data. Then, we calculated the number of vertices and surfaces for each object. For comparing the LoD of our 3D point cloud with the two other state-of-the-art methods, we have chosen some light field images as an input and after converting the 3D point cloud to PLY format, we obtained the LoD properties. Table 4 shows the result of LoD information for three different 3D models of light field images. It can be seen that the result of our method produced a higher number of vertices and faces compared to the two other methods. As a result, our method can create a denser 3D point cloud. We have done this experiment for several light field images and in all cases, the density of our method was higher than the other two methods. It should be noted that LoD can be increased by the addition of noise, however the results of the histogram analysis in the previous section show that our approach produces a more even distribution of point cloud positions, which is not what one would expect if our approach was simply noisier than other approaches.

Table 4. Numerical evaluation of 3D point cloud based on LoD ([number of vertices], [number of faces]).

Light Field Image	**Perra *et al.*[10]**	**Dansereau *et al.*[96]**	**Our Method**
Buildings-Black-fence. Figure 27	[65,264]^a , [12,6970]^b	[62,215]^a , [123,056]^b	[70,568]^a , [139,452]^b
Nature-Flowers. Figure 25	[84,729]^a , [165,584]^b	[72,980]^a , [141,069]^b	[98,245]^a , [192,874]^b
Green. Figure 26	[72,548]^a , [142,309]^b	[73,415]^a , [145,977]^b	[82,579]^a , [161,231]^b

^{a} number of vertices, ^{b} number of faces.

6.3.4. Discussion

The experimental results were evaluated visually and statistically, as explained in Sections 6.3.1-3 above. We evaluated our method with several light field images and from the results we can show that our 3D point cloud is more accurate and has less noise due to our modification of the depth map in two different steps. We solved the problem of pixel distribution in the first step of the modification by using histogram stretching and equalization. Furthermore, we solved the problem of density by using a special edge detection technique compared to using any current state-of-the-art methods for adding edges. We showed that by using fuzzy logic we could choose some special edges in a multi-modal fashion by comparing the intensity of neighbouring pixels. Our proposed approach is also novel because it employs parallel processing to improve what is conventionally achieved in generating a raw depth map from the Lytro camera. Unlike previous methods, the method involves a dual-enhancement approach that first computes the fuzzy logic orientation field computed from the histogram equalized sum of depth map and central sub-aperture images. The alternative pathway of our approach is to determine the Canny edge response to the central sub-aperture image and the LF depth map, which are then computed for SURF features and combined together through feature matching and merged with the result of the first orientation field computation phase. This newly proposed approach was found here to generate 3D point clouds for the purpose of remote sensing which were superior in detail and clarity, compared with the conventional approaches for generating a 3D point cloud from the depth map of a LF alone.

Chapter 7: Reflectance Recovery

An important aspect of the reconstruction of 3D objects is the accurate representation of the surface properties of an object. In this Chapter, we present a novel method using light field data to overcome the problem of reflectance estimation able to cover more general reflectance models associated with more complex material appearance. In this way, we analyse the light field image in two different spaces: colour and binarized space, to identify specular pixels and eliminate them from the image and address this problem. The advantages of using a light field camera in this approach is that we can access the wealth of useful information about the behaviour of pixels from different angles in a single capture with minimal effort. Extracting this information with a traditional camera in a single capture is impossible because we need information on different viewpoints of each specific patch of an image which is provided naturally by light field images as sub-aperture images.

Therefore our algorithm consists of two significant parts: one part is based on the RGB light field image analysis and the second part will analyse pixels in the binarized image.

7.1 The Challenge of Material Appearance

The challenge of accurate representation of material appearance in different objects with complex geometric layout and complicated spatially-varying indirect lighting such as, glossy surface reflectance, transparent or semitransparent surfaces have drawn considerable attention from the research community. Some aspects of this complex problem is shown in Figure 29 where a diagram shows various ways in which light can interact with a surface. The reconstruction of transparent, specular or refractive surfaces from images has become a key goal for active researchers in the field of computer vision and computer graphics. One of the solutions to overcome the problems of these complicated surfaces is to modify the reflectance or transmission characteristics of surfaces to be scanned to make scanning more simplified. This goal also can be obtained through the evaluation of refractive distortions from the diffuse background or analysis of illumination pattern which standard photography systems are not able to obtain this [104, 105]. Most approaches generally require multiple cameras, or multiple images however, recent research aims for a single camera and single image approach.

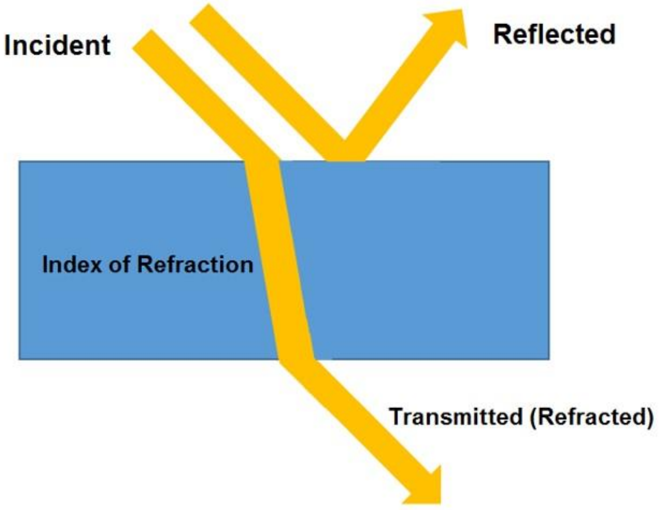
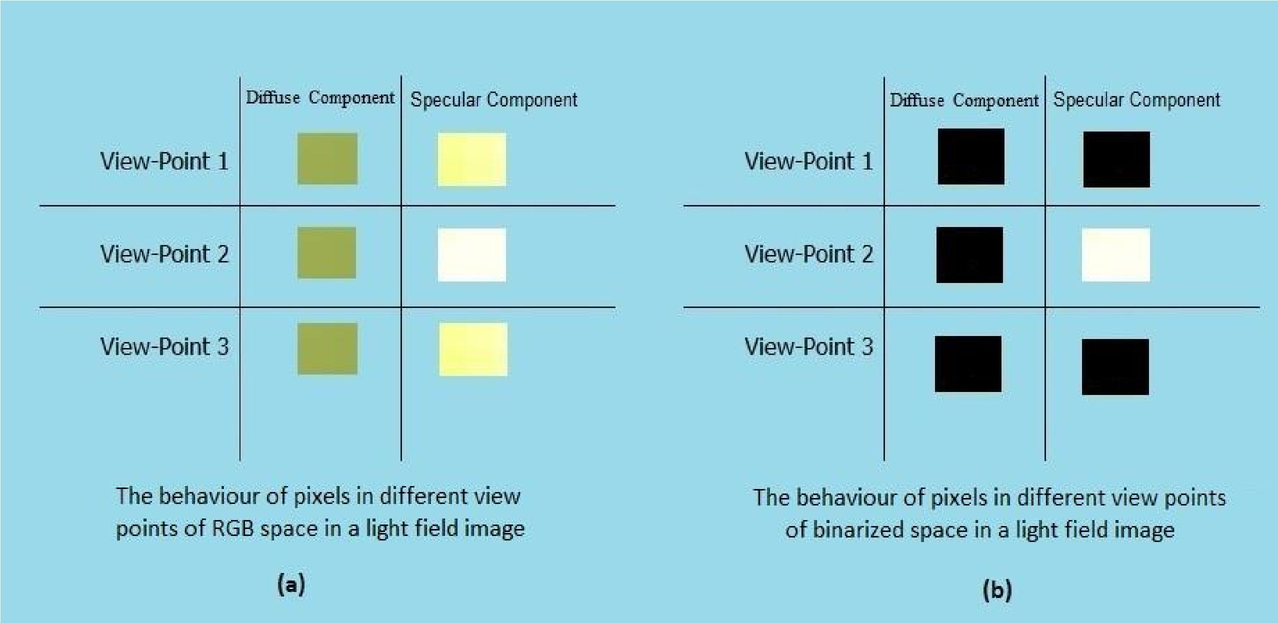


Figure 29. Complicated material appearance: The index of refraction of a material control whether incident light is reflected from the surface, enters the surface and is refracted or some combination of both effects.

The bidirectional reflectance distribution function (BRDF) describes how light is reflected at the surface to describe the surface appearance. BRDF is a four-dimensional function that describes how a surface reflects light based on view angles, light source angles, and the normal of surface. Therefore, identifying the properties of surface BRDF is one of the important components for several applications in computer vision. However, such tasks generally need extensive image captures from different viewing angles (up to hundreds of images). A light field image can be used to estimate the type of BRDF, which is not easy for traditional methods with a standard image. Next we will describe some works focusing on measuring BRDF based on a single light field image with the goal of remote BRDF type identification [106].

One significant method for handling the problem of reflectance for complicated surfaces is to estimate the light source colour by considering the variance of different views then separating specularity from glossy objects [17]. However, this method cannot cover several types of material appearances such as metals. For this reason, in this work in addition to using the coloured image, we also analysed the image in binarized space to recognise specular highlights and eliminate them from the image.

Lighting is one of the major contributors to the perception of 3D shape of objects. For illuminating an object, the incident light can be reflected, transmitted, scattered, absorbed and refracted [107]. Therefore, some part of the light will be reflected after it strikes the surface. The amount of reflectance is dependent on the type of material appearance and surfaces can represent various ways of reflecting light. The complexity of reflected light can be approximated by two general types of reflection: specular and diffuse reflection. Diffuse reflection reflects the light equally in all directions scattering the light like the surface of a paper. In contrast, in specular reflection, the light is reflected mostly at the same angle as the incoming light on the opposite side of the surface normal, like a mirror surface [108]. For analysing the reflection in light field images, we have studied the behaviour of pixels in different viewpoints. As a result, we found that specular and diffuse reflections can behave completely differently in different viewpoints of light field images. In diffuse surfaces we saw only a small change of colour at different viewpoint angles however, specular pixels can cause dramatic intensity and colour changes in different viewpoints. Furthermore, for finding the specular highlights we used an image binarization technique. Highlights can also demonstrate various behaviours under different illumination or under different viewpoints in binary images. In a binarized sub-aperture of a light field image, specular surfaces can be white in one sub-aperture and black in other sub apertures. However, for diffuse surfaces, there is comparatively minimal change across sub-apertures with different angular viewpoints (as shown in Figure 30).



The behaviour of pixels in different view points of RGB space in a light field image

(a)

The behaviour of pixels in different view points of binarized space in a light field image

(b)

Figure 30. An illustration of the different behaviours of diffuse and specular components in a light field image. (a) Analysis of the behaviour of pixels in coloured (RGB) space. It can be seen that the diffuse surface has no colour change with viewpoint (sub-aperture) changes, but the specular surface has colour intensity changes in different viewpoints. (b) In binarized space, the diffuse component stays black, but the specular pixels can change from black to white in different viewpoints.

7.2 Finding Specular Pixels by Colour Estimation

In this step our goal is to estimate the light source colour for every sub-aperture (15*15) in different viewpoints of light field image. Therefore, our input image is light field data and we need to detect substantial differences of colour in each patch of each sub-aperture image. In this way, we keep the central sub-aperture image as a reference and compare it with the multi-perspective viewpoint (other sub-apertures) to find pixels with large variance and changes. We denote these pixels as specular pixels and later we will separate these pixels from the central sub-aperture image. This approach relies on the colour values of diffuse pixels remaining constant in different viewpoints but the colour values of specular pixels being changed dramatically by changing angles of light in different views. However, using only this approach for finding specular pixels is not sufficiently discriminatory and we also find specular highlights using another method (Section 7.3) and then combine selected specular pixels by the two different techniques together and remove them from the original image.

Extracting multiple viewpoints by remapping light field data: Initially, for analysing colour variance, we need to extract multiple viewpoints from the light field data. In this work we use the refocusing ability of the light-field for this extraction and rearrangement of light field data can help us in this step. We have the light-field input image $I\alpha(s,t,u,v)$ where (u,v) the position of a pixel within sub-aperture image (s,t). Therefore, a pixel (u,v) in the re-focused sub-aperture image (s,t) is obtained by:

$$I\alpha(s,t,u,v) = I(s + u(1 - (1/\alpha)\ ), t + v(1 - (1/\alpha)), u, v) \tag{52}$$

The coordinates (*s,t*) represent the spatial domains and (*u,v*) represents the angular domains as shown in Figure 8. Angular information will be obtained from the lens plane and the spatial data will acquired from the sensor plane. The variable α is defined as the ratio of the refocused focal length to the current focal length (in our method, we used α = 0.1, which works for most of our examples). So, this kind of remapping can enable refocusing to extract multiple views.

Computing the colour intensity by K-means clustering: After extracting multiple views, for each patch of the sub-aperture image we compute the pixel value in RGB colour space at different angles to identify the component of BRDF. In particular, if we had high variability in the intensity of pixels across the R, G and B channels, then they will be recognized as a specular pixels and if we had little change we select them as diffuse pixels. This is due to the fact that pixels have different behaviour in different viewpoints.

This is achieved by using a K-means cluster method similar to [85]. We calculate the colour intensity changes within *u; v* of each *s; t*. Within *u; v*, we cluster them into two groups of specular and diffuse pixels.

Considering the difference in centroids between the clusters we can present two different groups of pixels: pixels *b* shows both diffuse and specular components together and *r* represent specular-free pixels. So, the total pixels number of angular position *u,v* in each *s,t* can be equal to *b*+*r*. We used a k-means cluster through the *u; v* pixels of each *s; t* to estimate the type of BRDF. In brief, the two centroids of clusters will be represented as ($\langle . \rangle$ defines the expected value):

$$C_b(s,t) = \langle I\alpha \rangle(s,t,b)$$

$$C_r(s,t) = \langle I\alpha \rangle(s,t,r) \tag{53}$$

To increase accuracy and amplify the difference between specular pixels and noise we define a confidence value and assign higher confidence to patches where the distance between the two centroids of clutters are maximum. We define the confidence value for each *I*(*s, t*) as follows,

$$F_I(s,t) = e^{-\beta_0/|C_b(s,t)|-\beta_1/|C_b(s,t)-C_r(s,t)|+\beta_2/\rho} \tag{54}$$

where ρ is the average distance between clusters and , β_0 related to the brightness term, β1 is for the centroid distance term, and β2 related to the robustness of the clustering. These are constant parameters (β_0 *and* β_1=0.5) and $\beta_2 = 1$ [85].

Separating specular pixels: For obtaining specular pixels we need to separate specular from diffuse pixels as below:

$$N(s,t) = C_b(s,t) - C_r(s,t)$$

$$SP1(s,t) = \langle F_I(s,t)N(s,t) \rangle \tag{55}$$

where $SP1(s,t)$ shows the specular pixels in this step.

7.3 Finding Specular Pixels by Binarized Image Analysis (Intensity based Thresholding)

In the second part we will obtain specular pixels based on brightness analysis in binarized space. After light strikes a surface, it may be reflected in either a diffuse or specular fashion. When reflected diffusely, the energy of the light is radiated in all directions equally, whereas in specular reflection the energy of the reflected light is radiated in one direction preferentially. When the image is viewed from the direction of specular reflection, this will lead to increase of intensity of pixels corresponding to the light source regions and create bright regions in an image which we denote as glare or a specular highlight. Our aim in this part is to remove these bright points which may not have been detected precisely by the previous part. The input of our system in this part is a sub-view of light field image $I_{u,v}(s,t)$ which can be acquired by gathering the samples with fixed *u,v*.

In this way we convert light field data to grayscale and then to binary space for detecting glare areas. This means binarizing the light field data by thresholding and gives us a set of specular hot-spots of light in the image because the specular pixels in an image have the highest intensity compared to the whole image. For finding the specular areas we need to detect components connected to the light source. With the image binarization technique we can segment an image into a connected group of pixels [109]. For detecting the reflected area in the binary image, having an accurate threshold is very important. For this reason we used Bernsen’s local thresholding technique taking into account higher intensities to obtain specular highlights in the image. In the first step of this part we convert RGB data to grayscale by establishing a weighted sum of the R, G, and B components as below:

$$I_g(s,t) = 0.2989 \ast R_{I(s,t)} + 0.5870 \ast G_{I(s,t)} + 0.1140 \ast B_{I(s,t)} \quad (56)$$

Then we used a local thresholding method to convert the grey scale image to binary. A popular method for finding the threshold locally is Bernsen's thresholding method. This method calculates the local contrast value (the minimum, maximum intensity of pixel) and the local mid-grey value (the mean of the pixels in the neighbourhood of each pixel) in the local window. This method uses circular local windows instead of rectangular ones. The binary image created using Bernsen's thresholding method $I_b(s,t)$ is obtained as below:

$$I_{lcv}(s,t) = (\, I_{gmax}(s,t) + \,(\, I_{gmax}(s,t))/2$$

$$I_{lmv}(s,t) = I_{gmax}(s,t) - \, I_{gmin}(s,t) \quad (57)$$

$$I_b(s,t) = \begin{cases} 1 & if\ (\mathrm{I}(s,t) < I_{lcv}(s,t)\ and\ I_{lmv}(s,t) \geq \gamma\,) \\ & or\ (\mathrm{I}(s,t) < \gamma^*\ and\ I_{lmv}(s,t) < \gamma\,) \\ 0, & otherwise \end{cases}$$

where $I_{gmax}(s,t)$ the maximum is grey value and $I_{gmax}(s,t)$ is the minimum gray value. $I_{lcv}(s,t)$ is defined as the local mid-grey value and $I_{lmv}(s,t)$ defined as the local contrast value. γ and γ^* are grey value threshold and contrast threshold respectively.

To recognize the specular highlights regions more precisely and compare with Bernsen's thresholding method, we also binarize the input grayscale image by finding the highest grey value point of image ($I_{gmax}(s,t)$). This means we binarize the grayscale image with a thresholding factor 0.9 of ($I_{gmax}(s,t)$) to obtain the highlight spots of light source as specular pixels which we denote as $I_h(s,t)$:

$$I_h(s,t) = \begin{cases} 1 & if\ (\mathrm{I}(s,t)\ >\ (\ I_{gmax}(s,t) * 0.9\ )) \\ 0, & \quad otherwise \end{cases} \tag{58}$$

Now we compare the two obtained binarized images from the two above mentioned methods and we remove false detections by using a connected component technique to discover the maximum connected component present in two corresponding regions of the two binarized images for finding the intersection of two regions. The maximum connected components of $I_b(s,t)$ and $I_h(s,t)$ is given as $Sp2(s,t)$.

$$Sp2(s,t) = (\ I_b(s,t) \oplus\ I_h(s,t)\ ) \ominus\ I_h(s,t) \tag{59}$$

The connected component obtain from morphological operations and ⊕ shows morphological dilation, which this operation thickens or grows objects in a binary image and ⊖ means morphological erosion that thins or shrinks objects in a binary image [110]. Therefore, $Sp2(s,t)$ shows specular pixels from second method (binarized image analysis.

7.4 Combination of Obtained Specular Pixels and Separation from the Original Image

Finally, we combine the specular pixels from the two different methods described in Sections 7.2 and 7.3, and remove the specular pixels from the original image to obtain the specular-free image as below:

$$IR(s,t) = I(s,t) - (Sp1(s,t) + Sp2(s,t)) \tag{60}$$

where $IR(s,t)$ shows specular free image.

Then we normalize specular free image as below:

$$N_{IR(s,t)} = \frac{IR(s,t) - \min\,(IR(s,t))}{\max(IR(s,t)) - \min\,(IR(s,t))} \tag{61}$$

where, $N_{IR(s,t)}$ is the normalized specular free image and Abs is absolute value of the image data. Then, we apply Wiener filtering – a kind of adaptive noise removal filter – as below:

$$IRF(s,t) = \mu + \frac{\sigma^2 - v^2}{\sigma^2}\,(N_{IR(s,t)}\,(N * M) - \mu) \tag{62}$$

$IRF(s,t)$ is the final specular free image and μ, σ^2, v^2 are mean, variance and local variance respectively and *N*M* is the local neighbourhood of each pixel in the image. *M* and *N* determines the size of the neighbourhood used to compute the local image mean and standard deviation. Neighbourhood size, defined as a two element vector [*M N*] where *M* is the rows number and *N* is the columns number.

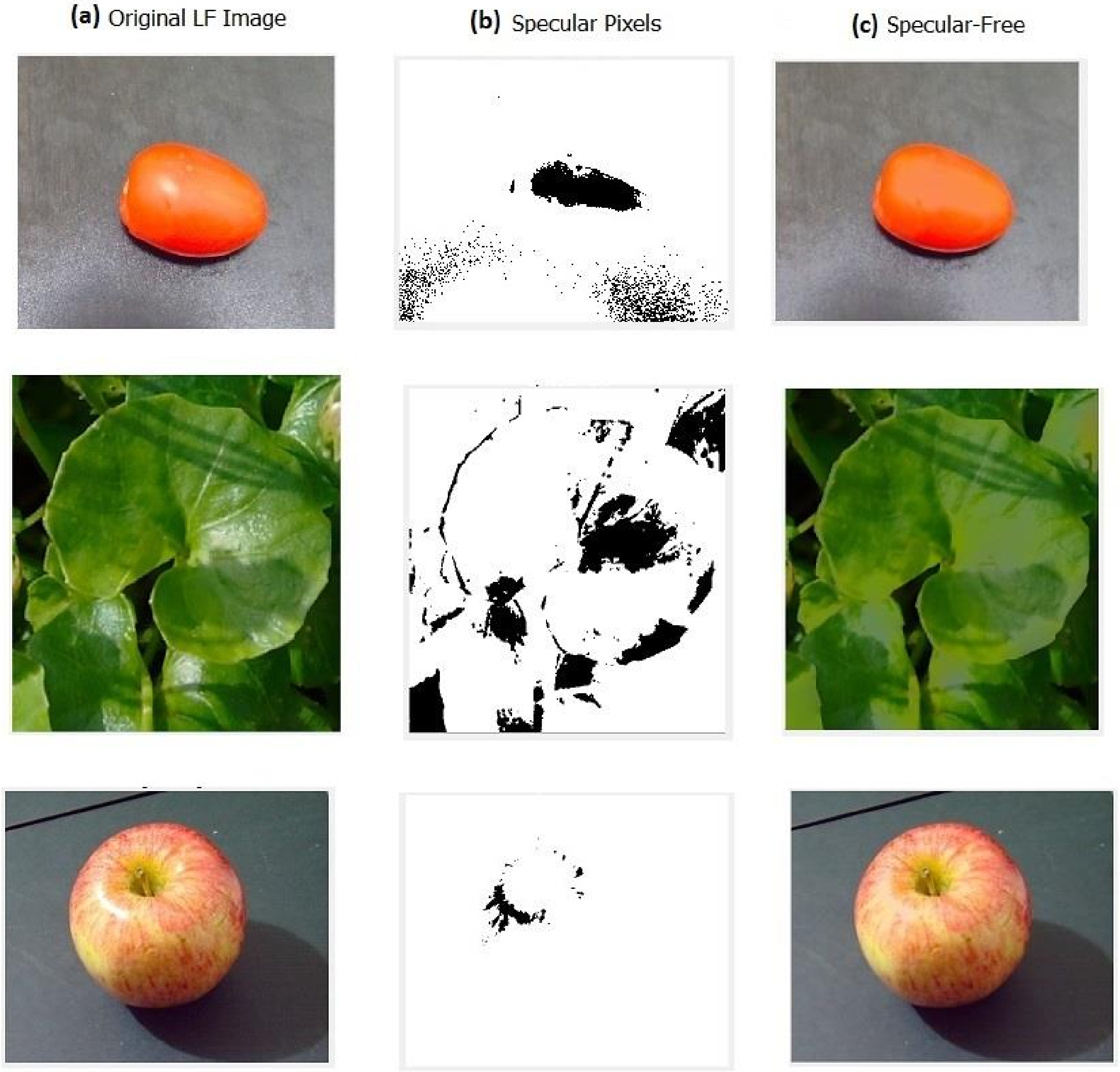


Figure 31. Our experimental results on the surface of fruit and leaves. The first column (a) shows real-world light field images captured by a Lytro camera. The second column (b) demonstrates the combination of obtained specular component from two different strategies of proposed method and the third column (c) shows specular free images.

7.5 Experimental Results

We tested our method with three different databases and compared the proposed method against two state of the art methods, as described in further detail below. For more extensive evaluation, in addition to visual comparison, we calculated the MSE of the proposed method and show the numerical evaluation in Section 7.5.3.

7.5.1. Result on Databases

We used three different databases; one synthetic database and two real-world light field image datasets. We evaluated our method with various types of objects. For the synthetic database we used a database popularized by the research community, created by Wanner [90]. For the real-world image databases, we used the real-world LF database from Tao *et al*. [17] which was attached with their released source code.

To demonstrate the capability of the proposed method on real-world images we also utilized a third custom database of images captured ourselves using a Lytro Illum camera. For a more comprehensive evaluation we captured images in different situations and different lighting conditions, including night time, day time and outdoor and indoor images. Our proposed method was tested on wide variety of images with different material appearance such as fruit, leaves, jewellery and glossy objects some of which are shown in Figure 31 and Figure 32. The results show that the proposed method can overcome the problem of reflectance estimation in wide variety of complicated material appearance situations while other methods are limited in the type of objects they can handle. These images are part of the results captured using a Lytro Illum camera.

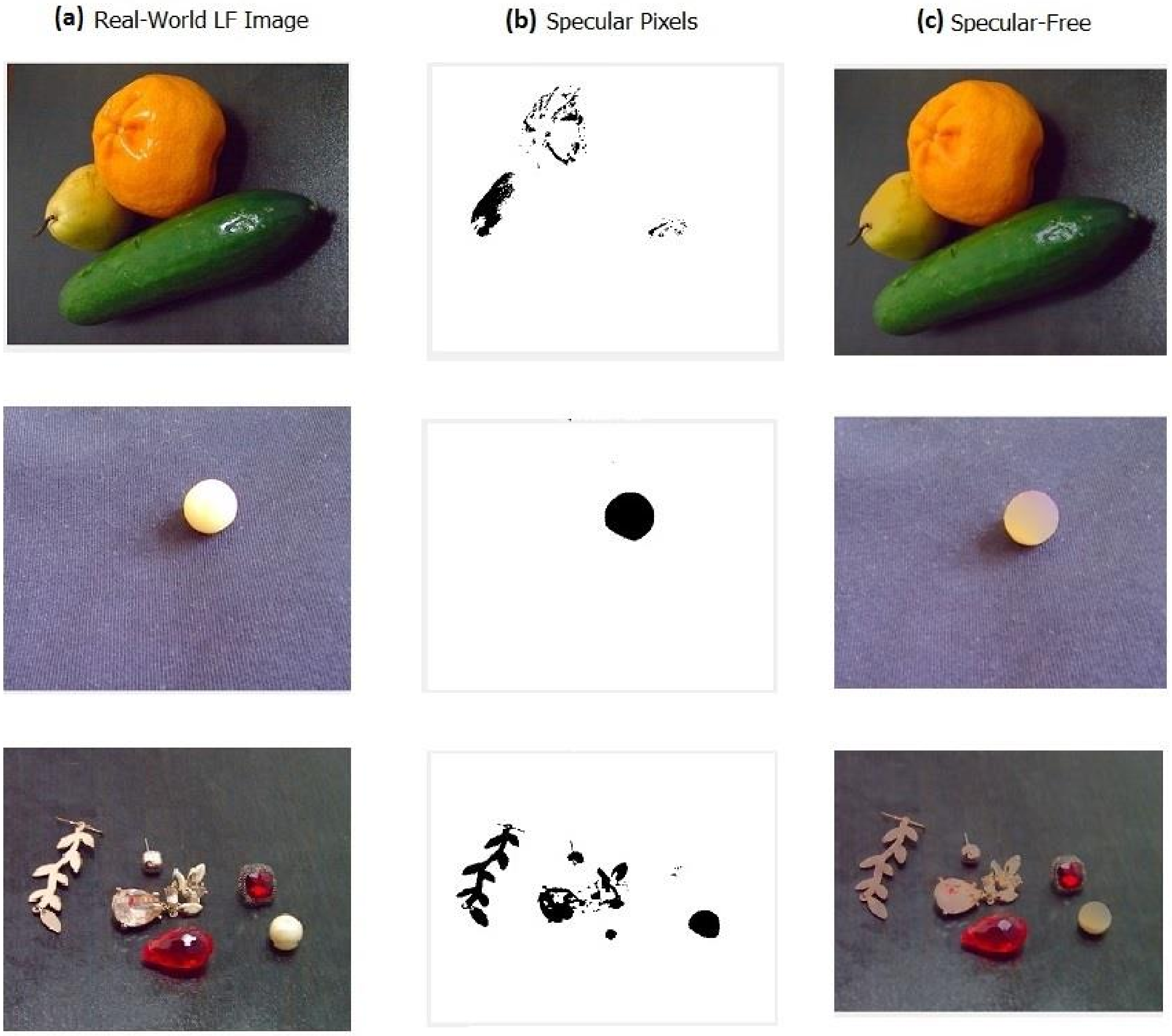


Figure 32. Our experimental results on surfaces of fruit, pearl, glass and metal. The first column (a) shows the real-world light field images captured by a Lytro Illum camera. The second column (b) demonstrates the combination of obtained specular component from two different strategies of proposed method and the third column (c) shows specular-free images. The first and second images were captured during daytime and last image was captured during night time.

7.5.2. Methods Compared

To demonstrate the accuracy of our proposed method, we compared our result against of two other methods: Tao *et al.* [17] and Shen *et al.*[38]. As shown in Figure 33, we compared the output of our method with their data. We re-implemented the method of Shen *et al.* [38] in MATLAB to compare its performance with our result. Shen *et al.* used a single image to separate the diffuse and specular pixels. Pixels are categorized into different clusters in a pseudo-chromaticity space and specular reflection of each pixel is acquired based on the ratio of corresponding intensity. However, this method has some limitations in dark and achromatic situations. As shown in Figure 33 , the Shen *et al.* method did not provide an accurate result. Tao *et al.* [17] released their source code and we ran their source code in MATLAB with their attached databases and compared their result with our results. As it is shown in Figure 33, Tao *et al.* used only source colour analysis for finding specular pixels, and fails to detect all highlights. In the contrast, as we combined the light source analysis by brightness analysis, and we can detect specular pixels more accurately.

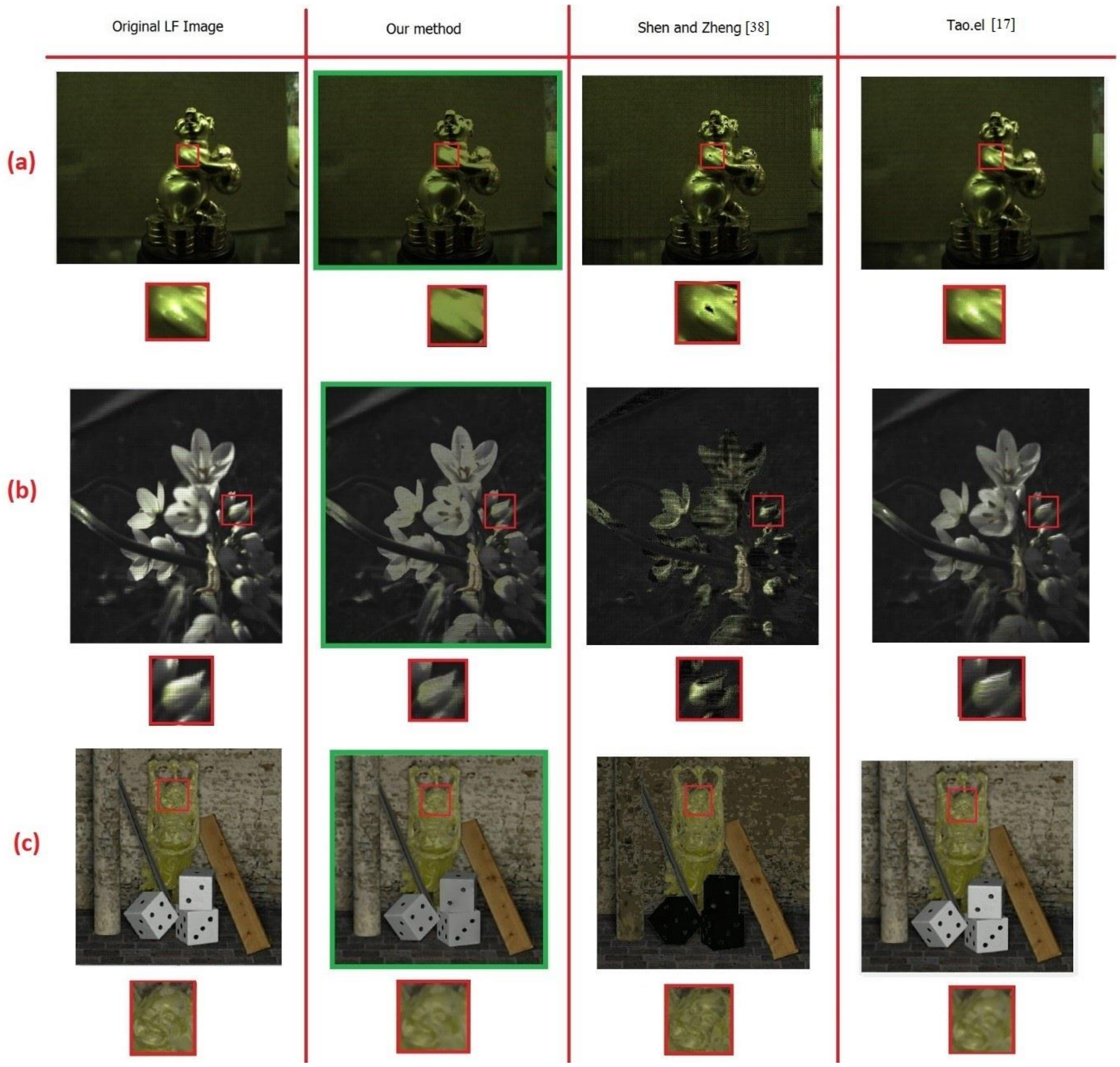


Figure 33. A comparison of proposed method with two other methods. The first and second images (a,b) are part of Tao et al. dataset and the last light field image (c) is from the Wanner synthetic database (Buddha).

7.5.3. Quantitative Evaluation

For evaluating the performance of our algorithms numerically, the MSE is calculated for the specular-free images compared to diffuse images provided by the Shekhar synthetic database [110]. As part of their database, they provided diffuse images and specular images separately which is convenient for us to compare our result with the extracted diffuse component by Shekhar *et al.* [110].

In this way, we have chosen some light field images from Shekhar *et al.* [110] and calculated MSE as shown in Table 5.

$$MSE_N = (\sum_{i\in N}(IRF(i) - difD(i))^2)/N \qquad (63)$$

where $difD$ is the diffuse image which provided by database [110], and IRF is the specular-free image which is estimated. N is the number of pixels in the specular-free image.

Table 5. MSE error of the specular-free images compared with diffuse images from the Shekhar et al. database [110].

Image	Tao *et al.* [17]	Shen *et al.*[38]	Our method
Living_Room	26.7	33.4	**18.2**
BMW_R80_Monster	21.6	24.2	**12.6**
Street_Guitar	3.2	5.6	**2.1**

In addition to the MSE metric, we also used the BadPix (0.07) metric, which is formulated based on the percentage of pixels where the absolute difference of the depth map and ground truth is greater than 0.07 (the specified threshold) [52]. Once again, the database of Shekhar et al. [109] were used. The results of this metric is shown in Table 6.

$$BadPix_N(0.07) = |(i \in N: |difD(i) - IRF(i)| > 0.07)|/|N| \qquad (64)$$

where $difD$ is the diffuse image which provided by database [110], and IRF is the specular-free image that obtained from previous step. N is the number of pixels in the specular-free image.

Table 6. The BadPix (0.07) error of the specular-free images compared with diffuse images from the Shekhar et al. database [110].

Image	Tao *et al.* [17]	Shen *et al.*[38]	Our method
Living_Room	58.1	86.2	**54.7**
BMW_R80_Monster	62.5	59.1	**35.1**
Street_Guitar	9.4	16.4	**6.3**

Chapter 8: Conclusions and Future Work

In this work, we have presented a new strategy to address major challenges in 3D reconstruction by using light field images. 3D reconstruction based on a single image is a challenging topic, and using a light field camera for this purpose is relatively new in this field. However, most current methods can only support limited material appearance such as Lambertian scenes, making them unreliable for complicated material appearance. Therefore, to have more general techniques to cover different material appearance we have proposed a 4-step approach. First, we have developed a novel strategy to estimate an accurate depth map, and then based on this depth map we have created a technique to generate a 3D point cloud. In the third step, we have enhanced the 3D point cloud and in the final step, we have addressed the problem of reflectance in light field images by utilizing a combination of two different methods.

In the first step, for depth map estimation, we have presented a strategy to overcome some major problems in depth map estimation such as occlusions, shadows and low-texture areas. Due to recent advancements in light field technology and the use of light field images in several applications, we have applied light field images to depth map estimation. Moreover, we have observed that by using the 4D light field format (instead of the lenslet format) the problem of depth map estimation for low texture areas is improved. We have showed that by using occlusion-aware images and combining sub-aperture image matching and defocussing, the problem of occlusion to some extent is addressed. We also have demonstrated that by using a fast weighted median filter, the amount of noise in the output will be reduced, improving the quality of the reconstruction in shadowed regions. The result also showed that our method is faster than other state-of-the-art methods. The effectiveness of our approach was confirmed by both synthetic and real-world datasets, and the improvements detailed in this work can be helpful for various applications such as 3D reconstruction, virtual, and augmented reality.

In the second step, for creation of a 3D point cloud, we have developed a solution for creating 3D point clouds based on a single image capture. We have used a light field image as an input to our system. The unique features of light field cameras lead to accurate depth map estimates. In particular, rich information about the scene can be obtained from the one image capture, including light intensity and the direction of light at a range of angles incident to the sensor. In our method, the 3D point cloud has been produced by transformation of the point-plane correspondences. We first have estimated the depth map based on the previous step, and then we create a 3D point cloud by transformation of the point-plane based on the enhanced depth map. The results confirmed that our method can create point clouds with improved accuracy compared to other state-of-the-art methods, and that our depth map is more accurate than that estimated by the Lytro software.

In the third step, to create an enhanced 3D point cloud, we have generated a 3D point cloud based on one light field image. For this purpose, we have proposed a modified two-step depth map approach to increase the accuracy of the depth map estimation for the generation of a 3D point cloud by transformation of the point–plane correspondences. Firstly, we have used histogram stretching and equalization which can improve the separation between the depth planes, and in the second step, we have developed a new strategy for adding multi-modal edge detection information to the previous step using fuzzy logic and feature matching. In this work, we have utilized a light field camera which can be useful for remote sensing applications such as generating a 3D point cloud for agriculture monitoring and monitoring of plant growth. We have chosen our images of buildings and plants to show some applications of our work in the area of remote sensing and environmental research. The results confirm that our method can generate 3D point clouds with improved accuracy compared to other state-of-the-art methods, and our modified depth map ensures an improved result.

Finally, in the last step, for overcoming the problem of reflectance estimation for complicated material appearance, we have presented a novel reflectance estimation approach by combination of two different pixel value analysis methods in colour and binary spaces. Using a single light field image provides us with rich information about the intensity and light of an image in a single snapshot which helps us to find specular pixels with minimum effort. On the one hand, we have used light source colour analysis techniques to find specular pixels and identify the specular and diffuse components of the BRDF. On the other hand, we have utilized an image binarization technique to detect specular highlights based on the combination of Bernsen's local thresholding technique with a higher intensities detection technique. Finally, we have combined the detected specular pixels and removed them from the original image to create a specular-free image. Experimental results confirm that the proposed approach can handle a wide variety of cases with different material appearance for which previous methods fail.

The future directions of this research would include:

1- Working on converting the 3D point cloud to recover the surface geometry of objects for further manipulation.

2- Considering the performance of the proposed reflectance recovery model using objects with more complicated material appearance, which are more challenging to scan into 3D point clouds.

REFERENCES

[1] D. Hendricks, "3D printing is already changing health care," *Accessed October,* vol. 18, p. 2018, 2016.

[2] A. S. Malik and T.-S. Choi, "A novel algorithm for estimation of depth map using image focus for 3D shape recovery in the presence of noise," *Pattern Recognition,* vol. 41, no. 7, pp. 2200-2225, 2008.

[3] S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE transactions on pattern analysis and machine intelligence,* vol. 36, no. 3, pp. 606-619, 2014.

[4] C. Chen, H. Lin, Z. Yu, S. Bing Kang, and J. Yu, "Light field stereo matching using bilateral statistics of surface cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1518-1525.

[5] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 673-680.

[6] H. Sheng, P. Zhao, S. Zhang, J. Zhang, and D. Yang, "Occlusion-aware depth estimation for light field using multi-orientation EPIs," *Pattern Recognition,* vol. 74, pp. 587-599, 2018.

[7] I. Viola, M. Řeřábek, and T. Ebrahimi, "Comparison and evaluation of light field image coding approaches," *IEEE Journal of selected topics in signal processing,* vol. 11, no. 7, pp. 1092-1106, 2017.

[8] C.-K. Liang and R. Ramamoorthi, "A light transport framework for lenslet light field cameras," *ACM Transactions on Graphics (TOG),* vol. 34, no. 2, p. 16, 2015.

[9] B. Yang, S. Rosa, A. Markham, N. Trigoni, and H. Wen, "Dense 3D object reconstruction from a single depth view," *IEEE transactions on pattern analysis and machine intelligence,* 2018.

[10] C. Perra, F. Murgia, and D. Giusto, "An analysis of 3D point cloud reconstruction from light field images," in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2016, pp. 1-6: IEEE.

[11] K. Li, T. Pham, H. Zhan, and I. Reid, "Efficient dense point cloud object reconstruction using deformation vector fields," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 497-513.

[12] R. Ng, M. Levoy, M. Brédif, 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.

[13] T.-T. Ngo, H. Nagahara, K. Nishino, R.-i. Taniguchi, and Y. Yagi, "Reflectance and shape estimation with a light field camera under natural illumination," *International Journal of Computer Vision,* vol. 127, no. 11-12, pp. 1707-1722, 2019.

[14] W.-C. Ma, S.-H. Chao, B.-Y. Chen, C.-F. Chang, M. Ouhyoung, and T. Nishita, "An efficient representation of complex materials for real-time rendering," in

Proceedings of the ACM symposium on Virtual reality software and technology, 2004, pp. 150-153.

[15] T.-C. Wang, M. Chandraker, A. A. Efros, and R. Ramamoorthi, "SVBRDF-invariant shape and reflectance estimation from light-field cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5451-5459.

[16] S. Jiddi, P. Robert, and E. Marchand, "Detecting specular reflections and cast shadows to estimate reflectance and illumination of dynamic indoor scenes," *IEEE transactions on visualization and computer graphics,* 2020.

[17] M. W. Tao, J.-C. Su, T.-C. Wang, J. Malik, and R. Ramamoorthi, "Depth estimation and specular removal for glossy surfaces using point and line consistency with light-field cameras," *IEEE transactions on pattern analysis and machine intelligence,* vol. 38, no. 6, pp. 1155-1169, 2015.

[18] M. Hog, "Light field editing and rendering," Rennes 1, 2018.

[19] R. Ng, *Digital light field photography.* stanford university Stanford, 2006.

[20] B. Wilkinson and P. Calder, "Augmented reality for the real world," in *Computer Graphics, Imaging and Visualisation, 2006 International Conference on*, 2006, pp. 452-457: IEEE.

[21] D. Cho, M. Lee, S. Kim, and Y.-W. Tai, "Modeling the calibration pipeline of the lytro camera for high quality light-field image reconstruction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3280-3287.

[22] G. Wu *et al.*, "Light field image processing: An overview," *IEEE Journal of Selected Topics in Signal Processing,* vol. 11, no. 7, pp. 926-954, 2017.

[23] R. J. Monteiro, N. M. Rodrigues, S. M. Faria, and P. J. Nunes, "Light field image coding: objective performance assessment of Lenslet and 4D LF data representations," in *Applications of Digital Image Processing XLI*, 2018, vol. 10752, p. 107520D: International Society for Optics and Photonics.

[24] M. Hog, "Light field editing and rendering: Édition et rendu de champs de lumière," Rennes 1; Rennes 1, 2018.

[25] R. Schima *et al.*, "Imagine all the plants: evaluation of a light-field camera for on-site crop growth monitoring," *Remote Sensing,* vol. 8, no. 10, p. 823, 2016.

[26] A. Isaksen, L. McMillan, and S. J. Gortler, "Dynamically reparameterized light fields," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 297-306.

[27] M. Levoy, B. Chen, V. Vaish, M. Horowitz, I. McDowall, and M. Bolas, "Synthetic aperture confocal imaging," *ACM Transactions on Graphics (ToG),* vol. 23, no. 3, pp. 825-834, 2004.

[28] Y. Xu, H. Nagahara, A. Shimada, and R.-i. Taniguchi, "Transcut: Transparent object segmentation from a light-field image," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3442-3450.

[29] K. Maeno, H. Nagahara, A. Shimada, and R.-i. Taniguchi, "Light field distortion feature for transparent object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2786-2793.

[30] Y. Furukawa and C. Hernández, "Multi-view stereo: A tutorial," *Foundations and Trends® in Computer Graphics and Vision,* vol. 9, no. 1-2, pp. 1-148, 2015.

[31] Y. Qu, J. Huang, and X. Zhang, "Rapid 3D Reconstruction for Image Sequence Acquired from UAV Camera," *Sensors,* vol. 18, no. 1, p. 225, 2018.

[32] R. Mohr, L. Quan, and F. Veillon, "Relative 3D reconstruction using multiple uncalibrated images," *The International Journal of Robotics Research,* vol. 14, no. 6, pp. 619-632, 1995.

[33] P. Beardsley, P. Torr, and A. Zisserman, "3D model acquisition from extended image sequences," in *European conference on computer vision*, 1996, pp. 683-695: Springer.

[34] N. Snavely, I. Simon, M. Goesele, R. Szeliski, and S. M. Seitz, "Scene reconstruction and visualization from community photo collections," *Proceedings of the IEEE,* vol. 98, no. 8, pp. 1370-1390, 2010.

[35] C. Sweeney, T. Sattler, T. Hollerer, M. Turk, and M. Pollefeys, "Optimizing the viewing graph for structure-from-motion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 801-809.

[36] D. J. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher, "SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion," *IEEE transactions on pattern analysis and machine intelligence,* vol. 35, no. 12, pp. 2841-2853, 2012.

[37] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *2010 IEEE computer society conference on computer vision and pattern recognition*, 2010, pp. 1434-1441: IEEE.

[38] S. Shen, "Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes," *IEEE transactions on image processing,* vol. 22, no. 5, pp. 1901-1914, 2013.

[39] M. Liu, S. Huang, G. Dissanayake, and H. Wang, "A convex optimization based approach for pose SLAM problems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 1898-1903: IEEE.

[40] T. Chen, Z. Zhu, S.-M. Hu, D. Cohen-Or, and A. Shamir, "Extracting 3D objects from photographs using 3-sweep," *Communications of the ACM,* vol. 59, no. 12, pp. 121-129, 2016.

[41] D. Jelinek and C. J. Taylor, "Reconstruction of linearly parameterized models from single images with a camera of unknown focal length," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, no. 7, pp. 767-773, 2001.

[42] P. Sturm and S. Maybank, "A method for interactive 3d reconstruction of piecewise planar objects from single images," in *The 10th British machine vision conference (BMVC'99)*, 1999, pp. 265-274: The British Machine Vision Association (BMVA).

[43] T. Xue, J. Liu, and X. Tang, "3-d modeling from a single view of a symmetric object," *IEEE Transactions on Image Processing,* vol. 21, no. 9, pp. 4180-4189, 2012.

[44] L. Zhang, G. Dugas-Phocion, J. S. Samson, and S. M. Seitz, "Single-view modelling of free-form scenes," *Computer Animation and Virtual Worlds,* vol. 13, no. 4, pp. 225-235, 2002.

[45] C. Zou, X. Peng, H. Lv, S. Chen, H. Fu, and J. Liu, "Sketch-based 3-D modeling for piecewise planar objects in single images," *Computers & Graphics,* vol. 46, pp. 130-137, 2015.

[46] N. Jiang, P. Tan, and L.-F. Cheong, "Symmetric architecture modeling with a single image," in *ACM SIGGRAPH Asia 2009 papers*, 2009, pp. 1-8.

[47] E. Toppe, C. Nieuwenhuis, and D. Cremers, "Relative volume constraints for single view 3D reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 177-184.

[48] F. Yan, M. Gong, D. Cohen-Or, O. Deussen, and B. Chen, "Flower reconstruction from a single photo," in *Computer Graphics Forum*, 2014, vol. 33, no. 2, pp. 439-447: Wiley Online Library.

[49] T. Georgiev and A. Lumsdaine, "Superresolution with plenoptic camera 2.0," *Adobe Systems Incorporated, Tech. Rep,* 2009.

[50] H.-G. Jeon *et al.*, "Accurate depth map estimation from a lenslet light field camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1547-1555.

[51] K. Inoue and M. Cho, "Enhanced depth estimation of integral imaging using pixel blink rate," *Optics and Lasers in Engineering,* vol. 115, pp. 1-6, 2019.

[52] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras," in *Computer Vision (ICCV), 2015 IEEE International Conference on*, 2015, pp. 3487-3495: IEEE.

[53] Y. Anisimov, O. Wasenmüller, and D. Stricker, "Rapid Light Field Depth Estimation with Semi-Global Matching," *arXiv preprint arXiv:1907.13449,* 2019.

[54] Y. Wang, X. Zhang, H. Li, and A. Ming, "Real-Time Light Field Depth Estimation via GPU-Accelerated Muti-View Semi-Global Matching," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 1054-1058: IEEE.

[55] G. HOUBEN, S. FUJITA, K. TAKAHASHI, and T. FUJII, "Fast and Robust Disparity Estimation from Noisy Light Fields Using 1-D Slanted Filters," *IEICE Transactions on Information and Systems,* vol. 102, no. 11, pp. 2101-2109, 2019.

[56] W. Chantara, J.-H. Mun, and Y.-S. Ho, "Efficient Depth Estimation for Light Field Images," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2018, pp. 1499-1502: IEEE.

[57] R. Ziegler, S. Bucheli, L. Ahrenberg, M. Magnor, and M. Gross, "A Bidirectional Light Field-Hologram Transform," in *Computer Graphics Forum*, 2007, vol. 26, no. 3, pp. 435-446: Wiley Online Library.

[58] M. Zhang, Y. Piao, C. Wei, and Z. Si, "Occlusion removal based on epipolar plane images in integral imaging system," *Optics & Laser Technology,* vol. 120, p. 105680, 2019.

[59] I. Tosic and K. Berkner, "Light field scale-depth space transform for dense depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 435-442.

[60] O. Johannsen, A. Sulc, and B. Goldluecke, "Occlusion-aware depth estimation using sparse light field coding," in *German Conference on Pattern Recognition*, 2016, pp. 207-218: Springer.

[61] W. Zhou, X. Wei, Y. Yan, W. Wang, and L. Lin, "A hybrid learning of multimodal cues for light field depth estimation," *Digital Signal Processing,* vol. 95, p. 102585, 2019.

[62] P. o. V. Raytracer. (2019, October 15). *POV.* Available: http://www.povray.org

[63] A. Pezzuolo, D. Giora, L. Sartori, and S. Guercini, "Automated 3D reconstruction of rural buildings from structure-from-motion (SfM) photogrammetry approach," in *proceedings of the international scientific conference.[Latvijas Lauksaimniec i¯ bas universit a¯ te]*, 2018.

[64] M. Weiss and F. Baret, "Using 3D point clouds derived from UAV RGB imagery to describe vineyard 3D macro-structure," *Remote Sensing,* vol. 9, no. 2, p. 111, 2017.

[65] H. Bae, J. White, M. Golparvar-Fard, Y. Pan, and Y. Sun, "Fast and scalable 3D cyber-physical modeling for high-precision mobile augmented reality systems," *Personal and Ubiquitous Computing,* vol. 19, no. 8, pp. 1275-1294, 2015.

[66] A. Dimitrov and M. Golparvar-Fard, "Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections," *Advanced Engineering Informatics,* vol. 28, no. 1, pp. 37-49, 2014.

[67] C. Kim, B. Kim, and H. Kim, "4D CAD model updating using image processing-based construction progress monitoring," *Automation in Construction,* vol. 35, pp. 44-52, 2013.

[68] M. Jarząbek-Rychard, D. Lin, and H.-G. Maas, "Supervised Detection of Façade Openings in 3D Point Clouds with Thermal Attributes," *Remote Sensing,* vol. 12, no. 3, p. 543, 2020.

[69] P. Kim, J. Chen, and Y. K. Cho, "SLAM-driven robotic mapping and registration of 3D point clouds," *Automation in Construction,* vol. 89, pp. 38-48, 2018.

[70] Y. Xia *et al.*, "RealPoint3D: Generating 3D Point Clouds from a Single Image of Complex Scenarios," *Remote Sensing,* vol. 11, no. 22, p. 2644, 2019.

[71] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan, "Pointflow: 3d point cloud generation with continuous normalizing flows," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4541-4550.

[72] L. Wang *et al.*, "3D Point Cloud Analysis and Classification in Large-Scale Scene Based on Deep Learning," *IEEE Access,* vol. 7, pp. 55649-55658, 2019.

[73] A. Vetrivel, M. Gerke, N. Kerle, F. Nex, and G. Vosselman, "Disaster damage detection through synergistic use of deep learning and 3D point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning," *ISPRS journal of photogrammetry and remote sensing,* vol. 140, pp. 45-59, 2018.

[74] W. Wang, R. Yu, Q. Huang, and U. Neumann, "Sgpn: Similarity group proposal network for 3d point cloud instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2569-2578.

[75] P. Mandikal, N. Murthy, M. Agarwal, and R. V. Babu, "3D-LMNet: Latent Embedding Matching for Accurate and Diverse 3D Point Cloud Reconstruction from a Single Image," *arXiv preprint arXiv:1807.07796,* 2018.

[76] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "Cospace: Common subspace learning from hyperspectral-multispectral correspondences," *IEEE Transactions on Geoscience and Remote Sensing,* 2019.

[77] Q. T. Pham and N.-S. Liou, "Hyperspectral Imaging System with Rotation Platform for Investigation of Jujube Skin Defects," *Applied Sciences,* vol. 10, no. 8, p. 2851, 2020.

[78] S. Rahman and A. Robles-Kelly, "An optimisation approach to the recovery of reflection parameters from a single hyperspectral image," *Computer vision and image understanding,* vol. 117, no. 12, pp. 1672-1688, 2013.

[79] X. Chen, M. S. Drew, and Z.-N. Li, "Illumination and reflectance spectra separation of hyperspectral image data under multiple illumination conditions," *Electronic Imaging,* vol. 2017, no. 18, pp. 194-199, 2017.

[80] G. Oxholm and K. Nishino, "Multiview shape and reflectance from natural illumination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2155-2162.

[81] J. Kim and A. Ghosh, "Polarized Light Field Imaging for Single-Shot Reflectance Separation," *Sensors,* vol. 18, no. 11, p. 3803, 2018.

[82] Y. Jeong, S. Moon, J. Jeong, G. Li, J. Cho, and B. Lee, "One shot 360-degree light field capture and reconstruction with depth extraction based on optical flow for light field camera," *Applied Sciences,* vol. 8, no. 6, p. 890, 2018.

[83] H. Farhood, S. Perry, E. Cheng, and J. Kim, "Enhanced 3D point cloud from a light field image," *Remote Sensing,* vol. 12, no. 7, p. 1125, 2020.

[84] H. Farhood, S. Perry, E. Cheng, and J. Kim, "3D point cloud reconstruction from a single 4D light field image," in *Optics, Photonics and Digital Technologies for Imaging Applications VI*, 2020, vol. 11353, p. 1135313: International Society for Optics and Photonics.

[85] M. W. Tao, T.-C. Wang, J. Malik, and R. Ramamoorthi, "Depth estimation for glossy surfaces with light-field cameras," in *European Conference on Computer Vision*, 2014, pp. 533-547: Springer.

[86] M. Works. (2019, October 15). *Find edges in intensity image*. Available: https://au.mathworks.com/help/images/ref/edge.html

[87] E. Y. Lam, "Computational photography with plenoptic camera and light field capture: tutorial," *JOSA A,* vol. 32, no. 11, pp. 2021-2032, 2015.

[88] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision,* vol. 60, no. 2, pp. 91-110, 2004.

[89] Q. Zhang, L. Xu, and J. Jia, "100+ times faster weighted median filter (WMF)," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2830-2837.

[90] S. Wanner, S. Meister, and B. Goldluecke, "Datasets and benchmarks for densely sampled 4D light fields," in *VMV*, 2013, vol. 13, pp. 225-226: Citeseer.

[91] M. Rerabek and T. Ebrahimi, "New light field image dataset," in *8th International Conference on Quality of Multimedia Experience (QoMEX)*, 2016, no. CONF.

[92] S. Wanner and B. Goldluecke, "Globally consistent depth labeling of 4D light fields," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 41-48: IEEE.

[93] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4D light fields," in *Asian Conference on Computer Vision*, 2016, pp. 19-34: Springer.

[94] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong, "Robust depth estimation for light field via spinning parallelogram operator," *Computer Vision and Image Understanding,* vol. 145, pp. 148-159, 2016.

[95] C.-X. Deng, G.-B. Wang, and X.-R. Yang, "Image edge detection algorithm based on improved canny operator," in *2013 International Conference on Wavelet Analysis and Pattern Recognition*, 2013, pp. 168-172: IEEE.

[96] D. G. Dansereau, I. Mahon, O. Pizarro, and S. B. Williams, "Plenoptic flow: Closed-form visual odometry for light field cameras," in *2011 IEEE/RSJ*

International Conference on Intelligent Robots and Systems, 2011, pp. 4455-4462: IEEE.

[97] J. Chen, Y. Fang, and Y. K. Cho, "Performance evaluation of 3D descriptors for object recognition in construction applications," *Automation in Construction,* vol. 86, pp. 44-52, 2018.

[98] F. M. Abubakar, "Image enhancement using histogram equalization and spatial filtering," *International Journal of Science and Research (IJSR),* vol. 1, no. 3, pp. 105-107, 2012.

[99] D. Aborisade, "Fuzzy logic based digital image edge detection," *Global Journal of Computer Science and Technology,* 2010.

[100] Z. Yang, D. Shen, and P.-T. Yap, "Image mosaicking using SURF features of line segments," *PloS one,* vol. 12, no. 3, 2017.

[101] S. Harwin and A. Lucieer, "Assessing the accuracy of georeferenced point clouds produced via multi-view stereopsis from unmanned aerial vehicle (UAV) imagery," *Remote Sensing,* vol. 4, no. 6, pp. 1573-1599, 2012.

[102] J. Chen, O. Mora, and K. Clarke, "ASSESSING THE ACCURACY AND PRECISION OF IMPERFECT POINT CLOUDS FOR 3D INDOOR MAPPING AND MODELING," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences,* vol. 4, 2018.

[103] A. Koutsoudis, B. Vidmar, G. Ioannakis, F. Arnaoutoglou, G. Pavlidis, and C. Chamzas, "Multi-image 3D reconstruction data evaluation," *Journal of Cultural Heritage,* vol. 15, no. 1, pp. 73-79, 2014.

[104] G. Wetzstein, D. Roodnick, W. Heidrich, and R. Raskar, "Refractive shape from light field distortion," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 1180-1186: IEEE.

[105] G. Wetzstein, R. Raskar, and W. Heidrich, "Hand-held schlieren photography with light field probes," in *Computational Photography (ICCP), 2011 IEEE International Conference on*, 2011, pp. 1-8: IEEE.

[106] F. Lu, L. He, S. You, X. Chen, and Z. Hao, "Identifying surface BRDF from a single 4-D light field image via deep neural network," *IEEE Journal of Selected Topics in Signal Processing,* vol. 11, no. 7, pp. 1047-1057, 2017.

[107] F. Leloup, S. Forment, J. Versluys, and P. Hanselaer, "Characterization of printed textile fabrics," in *Oxford V Conference on Spectrometry*, 2006.

[108] L. P. G. Christiansen, "Implementing and Combining Light, Shadows and Reflections in 3D Engines."

[109] M. Singh, R. K. Tiwari, K. Swami, and A. Vijayvargiya, "Detection of glare in night photography," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2016, pp. 865-870: IEEE.

[110] S. Shekhar *et al.*, "Light-field intrinsic dataset," in *British Machine Vision Conference 2018 (BMVC)*, 2018: British Machine Vision Association.

[111] https://cseweb.ucsd.edu//~viscomp/projects/LF/papers/ICCV15/occCode/