Observers for Scene Reconstruction Using Light-Field Measurements

Sean G. P. O'Brien

A thesis submitted for the degree of
Doctor of Philosophy
The Australian National University

May 2021

Except where otherwise indicated, this thesis is my own original work.

Sean G. P. O'Brien
19 May 2021

Acknowledgments

I would firstly like to thank my supervisor Jochen, who has guided me through my candidature and has honed my skills as a researcher, and for our years of deep and insightful discussions. His efforts and lessons will not be forgotten, and I am infinitely grateful for and enriched by them. Thank you to Viorela, who has taught me the value of clear communication and knowing the literature, and has been a constant source of encouragement throughout my candidature. Thank you to Rob, who has shared with me his vast experience and knowledge and has consistently provided highly practical advice for conducting research and approaching problems. Thank you to Katrina for your expertise on wavelets and curvelets which have opened new avenues of research. To avoid the embarassment of missing anyone else in particular, I will name nobody else in particular, but I would like to thank generally all the other staff, students, and researchers in the ANU and ACRV who have provided more than just research. Lastly, I want to thank my friends and family, and in particular my parents Dan and Kath, to whom I dedicate this thesis, and to my partner Bridgette for giving me oppurtunities to not discuss my PhD.

This research was also supported in more material ways. The Australian Centre of Excellence for Robotic Vision has provided support for this research indirectly in the form of centre conferences, summer schools, and researchers, for which I am grateful. This research was supported by the Australian Research Council through the ARC Discovery Project DP160100783 "Sensing a complex world: Infinite dimensional observer theory for robots."

Abstract

This thesis investigates the task of visual scene reconstruction from a systems theory perspective. In this framework, the scene can be considered as the unknown state to be estimated, and the output of the system is a light-field. While measurements of a light-field can be obtained through more classical sensors such as monocular cameras, light-field cameras offer several advantages for scene reconstruction because the gradients of light-fields are known to be highly correlated with depth. Proving what conditions are necessary in order for depth estimation to be possible has remained a significant theoretical gap in the literature. In this thesis it is shown that for any mildly complex scene class, if depth can be extracted from light-field gradients for any scene in that class, then it is necessary and sufficient that each scene in the class is Lambertian and textured. The geometry of light-field cameras is explored in detail, resulting in a novel bijective point-projection model with clear applications to scene reconstruction that is later used for state-of-the-art camera calibration. The performance of scene reconstruction tasks depends crucially on the way in which the scene is represented. Observers for explicit and implicit scene representations are derived. In both cases, convergence is guaranteed and demonstrated experimentally, but in the latter case, finite-time convergence is derived and under milder conditions, even if the underlying state is infinite-dimensional.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis proposes the use of observers for the estimation of dense reconstructions of scenes using light-field measurements. In this setting, the state to be estimated is a scene, and the output of that scene is a light-field. Systems and control theoretical concepts are used in this thesis to derive estimation techniques that include mathematical guarantees of performance and accurate sensor modelling and calibration.

The primary contributions are divided into two main parts. The first part of this thesis focuses on contributions to the understanding of light-field geometry. This part proposes a novel projection model for light-field cameras that relates points in front of the camera to 3D features in the 4D light-field. This is followed by a method for estimating disparity, including the derivation of necessary and sufficient conditions for estimating depth from first-order properties of light-field data. In order to verify the correctness of the point-projection model, a novel calibration procedure is proposed that outperforms comparable methods on a variety of light-field cameras and test data.

The second part of this thesis proposes the use of observers for scene reconstruction. An integral part of any scene reconstruction method is the choice of representation of the scene. The two most common methods of representation are explicit and implicit scene representations. In the former case, the scene is represented as the image of a function, whereas in the latter case the scene is represented as a level or superlevel set of a function. This thesis proposes an observer for both the explicit and implicit cases, and in each case asymptotic convergence is proven analytically and demonstrated experimentally. In the explicit case, an observer is derived that updates scenes in a manner that minimises a photometric error derived on the light field data, and a theoretical proof of asymptotic convergence for this system is provided. For the implicit case, an observer is proposed that estimates scenes for a variety of implicit scene representations. It is proven that this observer exhibits convergence to the true scene in finite-time, even if the underlying scene is drawn from an infinite-dimensional class of scenes. It is shown that the way in which a state is

updated depends not only on the sensor pose and the data produced by the sensor, but also on the way in which the scene is represented, and that some choices of representation result in better noise reduction properties.

1.1 Motivation and Contributions

Scene reconstruction is one of the most studied topics in computer vision and robotics, with applications ranging from surveying to surgery. Studies of this problem vary in terms of approach to the estimation task, the sensors used, and the representations of the scene used.

The study of filtering and observer methods for depth estimation has progressed substantially in the last decade [50, 124, 60, 8, 6, 28]. However, the literature on observer methods for scene reconstruction focuses primarily on two main types of scene representations: point clouds, as in [50, 60, 28], and depth maps, as in [124, 6, 8]. The typical way observers are applied to point cloud representations is to exploit the inherent tracking property of observers to predict where an image feature will appear in a subsequent measurement, given an estimate of the camera motion. An error is then applied between the predicted image feature location and the measured feature location at a later time, and the world feature location is updated in such a way as to minimise that error. Point cloud representations are sparse, meaning that much of the structure of the scene being estimated is lost, and although there are techniques for generating dense reconstructions from sparse representations [48, 47], these methods are typically not performed in real-time. As already mentioned, the primary alternative to point cloud representations in the filtering literature are depth map or disparity map representations. Such representations of scenes are dense, but local, meaning that they only provide estimates of the portion of a scene that is visible at a given time. In such methods, tracking of individual features is usually not performed. Instead, an update of the entire depth map is predicted based on camera motion and corrected based on new data.

This thesis proposes alternative observer-based approaches to scene reconstruction. The first method proposed in this thesis, discussed in Chapter 4, uses a point cloud representation, but without the need for tracking individual points. Instead, a time-varying vector field is computed that depends on the camera pose and current measurement of the scene. This vector field determines the velocities of individual points in the scene estimate in a way that guarantees that those point estimates converge to the true scene. The derivation of this observer requires careful examination of the underlying sensor geometry, and the techniques used in the theoretical analysis may be applicable to other scenarios where convergence of a state estimate to a

limit set is desired rather than convergence to a single state.

The second observer design presented in this thesis, discussed in Chapter 5, uses an implicit representation of a scene rather than an explicit representation. An implicit representation of a scene consists of a test to determine whether a given point is contained within the scene or not, rather than a method for constructing samples of points in the scene as with an explicit method such as a point cloud. Implicit representations of scenes have been used for several decades [80]. In computer vision, a common technique is known as Poisson surface reconstruction [48], in which dense surfaces are reconstructed from points and surface normals by finding a function that has zero-crossings at each point with a gradient in the directions specified by the normals. As such, this method is a technique for converting a set of point-normal pairs into a dense surface, rather than taking image data and producing a dense surface. An alternative method commonly used in robotics is known as occupancy mapping [105]. Occupancy mapping methods usually represent scenes in terms of assignments of values to voxels. As the number of parameters used in such mappings grows by $O(n^3)$ where n is the side resolution of the voxel grid, techniques such as octrees are often necessary to store such representations in memory. The second observer derived in this thesis takes a more abstract approach by instead estimating the parameters of the implicit function used to represent the surface. In doing so, many different scene representations may be considered including voxels, curvelet representations, and neural network representations of the implicit function. Addtionally, theoretical guarantees of convergence are provided that show that this kind of observer may in principle exhibit finite-time convergence to the true scene, even if the underlying representation is infinite-dimensional.

The imaging sensor used throughout this thesis is known as a light-field camera. Recently, inexpensive and lightweight light-field cameras have been made available to consumers. Because such light-field cameras are new technologies that are experiencing wider usage and have potential in mobile devices, robotics, and automated inspection, their theoretical properties and geometric models deserve to be studied in their own right. Although there have been investigations that examine the theoretical properties of light-field cameras and their variants [74, 26, 83], there is still much work to be done on exploring the theoretical guarantees such cameras can offer and the relationships between the various camera models that are used in the literature.

Much of the literature on light-field photography is explored from a practitioner's perspective, particularly in the area of depth-estimation where much of the recent progress has been in the use of regularisation or machine-learning techniques [82, 96]. Such techniques achieve success by implicit assumptions on the scene classes that generate light-field data, rather than by exploring light-field geometry. In particular,

these techniques constitute assumptions on what kind of scenes can be estimated, for example by excluding scenes with highly varying geometries in the case of regularisation or by only estimating scenes for which the relation between the scene geometry and measured light-field is described by a neural network function. While the success of these techniques often justifies these implicit assumptions, and because these techniques are increasing in popularity, the pure theoretical questions concerning sensor geometry are becoming increasingly ignored. Such questions are adressed in this thesis.

This thesis contains numerous theoretical contributions to the study of light-field geometry. The first is the development of a projection model that is of particular use in scene reconstruction tasks. Typically, light-field cameras are modelled by mapping rays of light to a 4-dimensional coordinate representing a lenslet and a pixel [56]. It is expected, then, that the set of all rays that pass through a point in space projects to a set of lenslets and pixels. Analysis of the geometry of light-field cameras reveals that this set takes the form of a disc-like object that in this thesis is called a 'plenoptic disc'. As a disc, it relies on 3 parameters: a centre and a radius. It is shown that the projection that maps points to plenoptic disc parameters is bijective, and so identification of plenoptic disc features within a light-field image allows for the precise estimation of the location of the point corresponding to that disc in front of the camera. This is very unlike the situation for a conventional camera where 3D points project to 2D objects, and it is this property that makes depth estimation from a single light-field image possible.

This thesis applies the extraction of these plenoptic disc features to camera calibration. The calibration of light-field cameras pertains to the identification of model parameters which are used in subsequent tasks such as scene reconstruction, pose estimation, and SLAM. There have been many different approaches to light-field camera calibration in recent years [22, 9, 75]. Point projection models offer some advantages over ray projection models due to massive redundancies in the data used to estimate plenoptic discs. Because of this, calibration techniques based on point projections are more resistant to noise and systematic errors in the underlying data, meaning that plenoptic disc features may be identified with more accuracy than ray-based features. This insight was used to develop a light-field camera calibration technique that was then compared against several competing ray projection calibration techniques and found to have state-of-the-art performance when compared with these methods.

Related to the concept of plenoptic discs, is the concept of disparity, which is a quantity proportional to the radius of these discs and is related to depth through the calibration parameters of the camera. One of the primary advantages of light-field

cameras over stereo cameras is the ability to estimate disparity with simple image processing operations such as image convolutions rather than through more complex feature-matching techniques. As such, there exist extremely computationally efficient techniques for estimating high-resolution dense depth maps with a light-field camera. Disparity estimation is a popular subject of study amongst the light-field literature, with several new papers on the topic published in recent years [45]. While these techniques have steadily increased in accuracy when applied to synthetically-generated benchmark data [37], the theoretical questions surrounding the limits of depth estimation have remained open. It has been observed in the light-field literature that non-Lambertian scenes and poorly textured scenes pose difficulties to depth estimation techniques [45]. While it is well-known in the computer vision literature that Lambertian and textured scenes are important, previous work has not proven this importance. The final contribution of this thesis is a theorem that states roughly that if a depth estimation technique that uses first-order light-field data is capable of estimating the depth of the simplest class of scenes – planar scenes that are fronto-parallel to the camera – then, under certain other mild conditions, that depth estimation technique will only ever correctly estimate depth for scenes that are Lambertian and textured.

In summary, the contributions of this thesis are as follows:

- A novel projection model that relates points imaged by the camera to *plenoptic disc features* present in the raw camera data.
- A mathematical proof of the necessary and sufficient conditions that a scene colouring must satisfy in order for depth to be estimated from first-order light-field data.
- Derivation of a feature estimator for plenoptic disc features of checkerboards.
- A calibration method based on this geometry that outperforms existing state-of-the-art techniques in terms of accuracy and robustness.
- Derivation of a geometric observer for point cloud representations of scenes with theoretical guarantees of convergence.
- Derivation of an observer for infinite-dimensional implicit scene representations, such as curvelets or neural networks, that provably converges in finite-time.

Many of these contributions were previously reported in the following individual papers:

1. S. G. P. O'Brien, J. Trumpf, V. Ila, and R. Mahony, *Calibrating Light-Field Cameras Using Plenoptic Disc Features*, 2018, International Conference on 3D Vision (3DV), 286-294. [79]

2. S. G. P. O'Brien, J. Trumpf, V. Ila, and R. Mahony, *A Geometric Observer for Scene Reconstruction Using Plenoptic Cameras*, 2018, IEEE Conference on Decision and Control (CDC), 557-564. [77]

3. S. G. P. O'Brien, J. Trumpf, V. Ila, and R. Mahony, *Estimation and Geometry of Disparity-Fields from Raw Light-Fields*, 2019, *(Preprint)* [78]

4. S. G. P. O'Brien, K. Ashton, J. Trumpf, *An Observer for Infinite Dimensional 3D Surface Reconstruction that Converges in Finite Time*, 2020, 21st International Federation of Automatic Control World Congress (IFAC), 4947-4954 [76].

1.2 Literature Review

Scene reconstruction is a topic with three major components. The first component concerns how a scene may be effectively represented as a mathematical object, the second involves what information about a scene can be gathered from sensor measurements, and the third involves effective ways to combine sensor measurements taken over time into a single coherent scene estimate in the chosen representation. As such, this section is divided into three subsections: one that reviews different standard sensor geometries that have been used in past decades to estimate scenes, one that reviews different techniques for representing a scene estimate, and one that reviews techniques that produces estimates from sensor data. The terms '3D reconstruction', 'simultaneous localisation and mapping' (SLAM), and 'structure-from-motion' (SfM), all share similar meanings and have been studied extensively by many different communities over several decades, and numerous surveys and books of these topics and their variants have been published [93, 81, 11, 25, 92, 18, 103]. Because it is so expansive, it is beyond the scope of this thesis to review all of this literature, and so the interested reader is referred to one of the available surveys and texts on this topic. The following sections review recent and historical work.

1.2.1 Scene Reconstruction

Scene reconstruction is a broad area of research that has been studied throughout history by many different disciplines. Historically, the task of reconstructing scenes from visual measurements was known as *photogrammetry*, a term that was first used in 1867 by the architect Albrecht Meydenbauer, who used the then novel technology

of photography for surveying purposes [71]. Although this topic has developed dramatically since then, all of the following methods are at least one of the following:

1. *local*, meaning that they only represent the portion of the scene that is currently being measured,
2. *sparse*, meaning that the scene is represented as a discrete set of points,
3. *offline*, meaning that they only operate once all data is collected,
4. *resource-intensive*, meaning that they quickly encounter memory or processing limits as the scene becomes larger, or the computational resources available become smaller
5. *situational*, meaning that they only work in very controlled scenarios outside of which it is difficult to guarantee that they work.

Closed-Form Methods

Closed-form solutions are the earliest types of scene reconstruction methods discussed in the literature, with classical work on computer-aided scene reconstruction developed by the neurologists Marr and Poggio in 1976 [66] in a paper that also contains an early reference to the concept of disparity. This work was followed in 1981, when the psychologist Longuet-Higgins published a paper detailing what is now known as the 8-point algorithm [58].

Closed-form solutions typically separate scene reconstruction into two main tasks: the *correspondence problem*, whereby a one-to-one correspondence between points in two or more different perspectives must be constructed, and the *triangulation problem*, whereby depth must be obtained from these two perspectives of the same point. Typically, this problem takes the form of a system of linear equations for which there is an exact solution given perfect data, and which can still be solved in the least-squares sense in the case of imperfect data. Even though it has been observed that such solutions minimise the *algebraic error* and not a *geometric error* that respects the camera geometry, these techniques can often achieve good results [34]. Since closed-form methods tend to rely on the relation between feature points visible in at least two views, as with a stereo camera, these methods tend to be sparse and local.

Optimisation Methods

Optimisation methods are often used to refine the results obtained using closed-form methods. With an optimisation method, scene reconstruction becomes a task of

cost-minimisation, where some *cost function* is constructed that it is assumed that the true scene will minimise given the measurement data, and an optimisation routine is performed in order to find the parameters of the scene that minimise this cost function.

A classical optimisation-based approach to scene reconstruction is *bundle adjustment*, which has been widely studied [107]. In bundle adjustment, the *reprojection error* which is an error that has as data the obtained 2D feature point projection locations in each image, or some other geometric error is minimised, and has as parameters the 3D locations of feature points on the scene, as well as the intrinsic and extrinsic camera parameters. These parameters are varied until the predicted location of the image features are close to the measured location of the image features.

Many of the following methods could be considered subtypes of optimisation methods, or use explicit optimisation techniques to refine their initial results. Since solving high-dimensional optimisation problems is often computationally expensive, optimisation methods tend to be resource-intensive and offline, and most implementations of these methods also tend to be sparse.

Model-Fitting Methods

Model-fitting techniques are methods that obtain scene reconstructions when the problem is under-specified, such as from reconstructions from single monocular images. While optimisation is often used in such methods, a distinction should be drawn between the methods that depend on strong assumptions on the scene model as in this subsection, and methods that only depend on assumptions on the camera model as in the previous subsection.

Examples of such methods include depth from *monocular cues.* A monocular cue is some feature within an image that contains depth information if an assumption is made of the texture of the scene or the lighting conditions present in the environment. One such method is *shape-from-shading* [38], whereby the depth of a point in a scene can be estimated from a single image under Lambertian texture conditions by solving the *brightness equation*, a partial differential equation describing the apparent brightness of a point on the scene. It is now understood that this PDE does not have a unique solution, but that with further constraints applied to the location of the lighting source, the reflectance model, and the camera parameters the problem can be solved [88].

Another technique known as *photometric stereo* uses a stationary camera and stationary object but a moving lighting source [120]. Under a Lambertian reflectance model, the brightness of a point on the scene will depend on the angle between the

normal vector of that point and the vector pointing from that point to the lighting source. Because of this, it is possible to obtain the overall shape of a scene from a single perspective under different illumination conditions.

One futher method that is used is *shape-from-texture* [118], whereby scene information is reconstructed from knowledge of the colouring applied to that scene, and its apparent distortion through a perspective projection.

Model-fitting methods tend to be situational and require a careful experimental setup or specific lighting conditions, offline, and local – as these methods are typically used to estimate depth maps.

Voxel-Based Methods

Voxel-based methods are methods whereby the scene estimate is discretised into a finite-resolution voxel grid, and the task is to determine which set of voxels are contained within the scene and which are not.

Voxel-based methods include *space-carving* which is a technique used for stationary objects [53]. In space carving, it is initially assumed that every voxel in the total voxel grid is occupied. Then, given an image of the object to be estimated, voxels from the overall grid are removed if that voxel appears to lie in the background of the image. Of course, it can be seen that such an approach will only ever estimate the visual hull [55] of an object unless further model constraints are applied.

Another voxel-based method is *occupancy grid mapping*, whereby each voxel in the scene is assigned a probability of being occupied or not [105]. This probability is updated based on an *inverse measurement model*, which is essentially a pose and measurement dependent conditional probability distribution which is integrated into the current occupancy mapping estimate at every timestep.

Space carving methods tend to be offline and situational and require a careful experimental setup to ensure the foreground can be separated from the background in images. Because voxel methods also divide the scene into a discrete voxel grid, they are also sparse.

Learning-Based Methods

There has been an explosion of research into learning-based methods for 3D reconstruction in the last decade due to the success of neural networks for solving computer vision problems. A recent paper surveying learning-based techniques for scene reconstruction states that 149 such methods have been published between the years 2015 and 2019 [30], a recent example of which is given by Mescheder *et al.*[70]. Learning-based methods are essentially a type of model-fitting, where the model

used is constructed implicitly from previously obtained experimental data rather than explicitly based on a theoretical model. These techniques have been studied extensively in recent years by the machine learning community. In that community, methods consist of learning a function f that takes as input an image $\mathbf{I}$ and produces an implicit representation $f(\mathbf{I}, \cdot) : \mathbb{R}^3 \to \mathbb{R}$, and as such the function learned is not an implicit representation of a scene but a parametrised implicit representation of a scene class where the parameter is an image.

While practitioners may argue otherwise, it could be argued that these methods are situational as it is very difficult to analytically prove that they work far beyond the data they are trained on. These methods are typically resource-intensive, requiring the use of powerful GPUs for training. Usually these methods are used to estimate depth-maps or the structure of objects they currently see, making them local.

1.2.2 Filtering and Observer Methods

A final class of techniques that have been applied in the literature involve control theoretical techniques. In contrast to most of the other methods listed, filtering and observer methods are designed to be online, non resource-intensive, and typically come with analytical proof of their convergence behaviour based on rigorous foundations [54, 3, 10]. However, all methods in the literature for the purpose of scene reconstruction use either sparse or local representations of the scene.

Of the sparse, feature-based methods, earlier works were based on the theory of perspective systems, which model the way that feature points appear to change due to motions of a pinhole camera. Seminal work in this area includes that of Chen and Kano [15, 16]. The observers in those papers are based on an internal model principle, where the innovation is analysed using Lyapunov methods. This work was followed by Dahl *et al.*[19] which continued the work in perspective systems, performing an observability analysis for the system, and exploits the algebraic structure of the system dynamics that can be expressed in block triangular observer form allowing for the relevant design techniques to be applied [111]. The study of perspective vision systems is still ongoing, with a recent work using contraction analysis to design these systems [28]. Other recent work includes that of Keshavan *et al.*[50] which uses Lyanpunov techniques for the design and analysis of the resulting observer when the underlying motion parameters of the system are unknown. Other work involves the design of Kalman filtering methods [60], however these techniques involve the transformation of the nonlinear system state into a linear time-varying state. More recent techniques are based on the internal-model principle and propose gradient-descent based techniques [130]. Recent developments adopt a more geometric and unified

approach to the SLAM problem, involving the use of equivariant observer theory which instead represent the state of the system as an element of an underlying symmetry group of the combined pose and point-cloud in order to design an observer on a single geometric object [62, 109]. However, the resulting representations of the scene are still finite-dimensional and sparse in these methods.

The only alternative in the control theory literature to point-cloud representations are local depth-map estimation techniques. Techniques based on Kalman filtering methods date back to the late eighties [68]. These techniques continue to be studied, and in the last decade a number of new techniques have been published. A recent observer design exploiting the invariance of spherically-defined depth maps to rotation were proposed by Zarrouati *et al.*[124]. More recent work includes that of Becker *et al.*[6] that provides a second-order filtering approach to the problem combined with spatial regularisation of the estimated depth maps. Since then, second-order minimum energy filtering methods have been proposed for this problem by Berger [8] that have built upon the theoretical work of Saccon *et al.*[91].

While observer and filtering techniques for the purpose of scene reconstruction continue to be studied, none of these techniques have so far estimated dense and global representations of scenes. Additionally, the adaption of sparse or local observer techniques to novel sensor technologies is still relatively unexplored.

1.2.3 Sensing Modalities

There are many different types of sensor that produce data from which scene information can be extracted. These sensors can be divided into two main categories: passive sensors, and active sensors. As there are many different types of sensors that can be used for scene reconstruction, we will limit this discussion only to those sensors that have been widely researched by the computer vision and robotics communities, and avoid sensors that are more typically used in medical imaging, surveying, and geology communities, such as MRI, CT, ultrasonic, and radar. All of the following sensor technologies are regularly used in both computer vision and robotics communities, and several surveys with details about these sensors are available [106, 11].

Monocular

A monocular camera is a standard visual imaging sensor that is now found on many consumer devices. Monocular cameras are the most widely studied and used sensor in both computer vision and robotics, with most computer vision texts devoting the majority of their contents to the data produced by these sensors. The standard model of a monocular camera is that it records the colour and direction of the rays of light

that pass through the optical centre of its aperture. This model is known as the *pinhole camera model* and defines a perspective projection, which maps 3D points to 2D points [33]. Because of this, depth information is not present within a single image produced by a monocular camera unless model-fitting techniques are applied. However, a sequence of images of the same object from many perspectives often does contain depth information.

All of the techniques of the previous section have been used to produce scene reconstructions from monocular image data. There are two main ways of obtaining scene reconstructions from sets of monocular data: methods which use ordered sequences of images and methods that use unordered sets of images. When ordered sequences of images are used, typically the motion of the camera from one frame to the next is either estimated using optic flow tracking methods, or is estimated using inertial measurements. When the data consists of an unordered set, a global optimisation on a reprojection error on all of the images is typically used as in bundle adjustment methods.

Stereo

A stereo camera consists of two cameras spaced a known distance apart from one another. The most standard configuration for the relative orientations and intrinsic calibration parameters for stereo systems is that these parameters are equal, however there has been research on cameras that can change their orientation and focal settings to focus on particular objects, as is the case with human eyesight.

Classical work on binocular vision systems typically involves solving a correspondence problem, then computing a fundamental matrix for the system using the 8-point algorithm which provides a constraint on where a feature appearing in one image can appear in a second image [58]. This is usually followed by a refinement of the feature-matching step taking into account the fundamental matrix estimate, and estimation of the essential matrix which provides a constraint on the normalised image coordinates. With the essential matrix and image feature coordinate data, a system of linear equations can be constructed which can be solved for camera pose and 3D feature coordinates. These estimates are often refined using an optimisation on the reprojection error.

Trifocal

A trifocal camera is a camera that images a scene from three different perspectives rather than two as with a stereo camera. The primary advantage of such a camera is the redundancy of the sensor data that can be used to improve scene reconstruction

estimates, as well as, for the case of non-colinear arrangements of the three cameras, the ability to better handle certain degenerate scenarios where disparity cannot be effectively estimated due to the aperture problem. The relationship between three cameras that are modelled by perspective projections is given by the trifocal tensor, which plays a similar role to the fundamental matrix in a stereo camera [31].

Multi-view

Multi-view reconstruction methods typically either refer to reconstruction of a scene due to a moving camera (see Section 1.2.3) or to the reconstruction of a scene using a camera rig or pushbroom camera. A pushbroom camera is a sensor that consists of more than three optically similar co-linear cameras with the same orientation [32]. A camera array is a sensor consisting of a large number of fixed cameras with known poses imaging a scene from multiple perspectives, and scene reconstruction using this technology has been extensively studied [90, 115, 56, 122, 127, 116]. Camera arrays serve as a precursor to modern lenslet-based light-field cameras.

Sonar

Although passive sonar exists, and is used in military applications, active sonar is the more common variant in computer vision and robotics. Sonar operates on the same time-of-flight principle as lidar, but uses sound as a medium. Scene reconstruction using sonar has been achieved by a variety of communities including robotics, geology, surveying, and medical imaging. In robotics, sonar is often used in conjunction with occupancy mapping methods, and occupancy grid mapping was classically tested with this sensor [105, 104]. Sonar is an actively researched topic in marine engineering as light rapidly attenuates in underwater conditions.

Structured Light

Structured-light techniques are methods in which a projector is used to illuminate a scene in a way that allows the correspondence problem to be solved with a high degree of accuracy [72]. This technology relies on the fact that a projector has the same geometry as a monocular camera, but that the projection is applied in reverse. Because of this, standard stereo techniques can be applied even though the 'stereo camera' consists of a projector and a single monocular camera. Most techniques project a sequence of images onto the scene in order to uniquely encode each point on the scene with a binary sequence of colour values. Structured light is the method used by the Kinect sensor, and because of the low-cost and ease of use of this sensor, this technique is widely used amongst robotics communities.

Lidar

Lidar systems are systems that provide estimates of the distance of a point on a scene in a range of directions by emitting a laser pulse and measuring the time it takes for the pulse to be reflected back to a reciever. The primary advantage of lidar is that the sensor provides a direct measurement of distance that does not depend on the texture of the scene, rather than a measurement from which distance may be inferred and disrupted due to poorly-textured regions. Lidar is a well-understood sensing technology [69], with wide applications in robotics tasks such as SLAM [11].

1.2.4 Light-Field Cameras

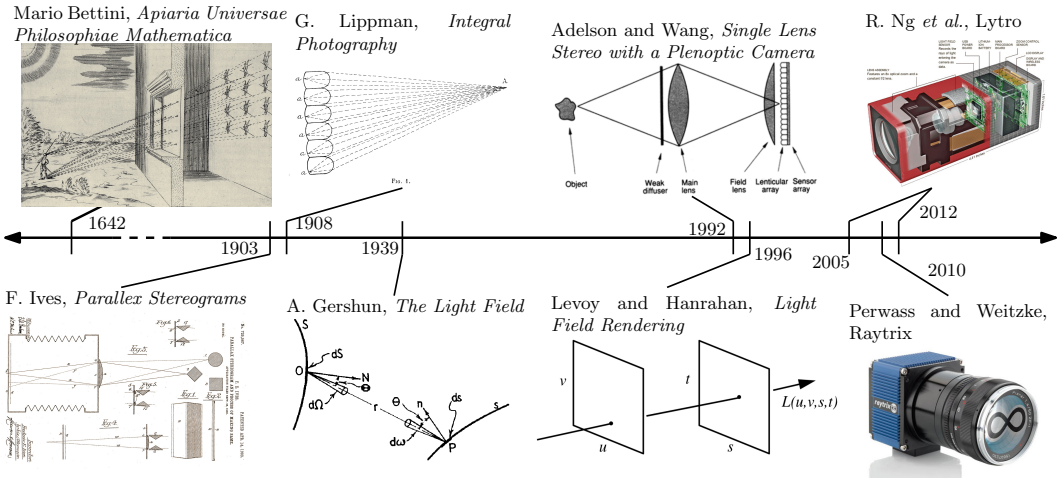


Figure 1.1: Timeline of Light-Field Research, citing [40, 57, 27, 2, 56, 61, 74].

The primary difference between a light-field camera and a multi-view camera system is that light-field cameras image light from much more densely sampled perspectives, and because of this, are capable of estimating the differential properties of a light-field. Not only are light-field cameras capable of estimating this differential information, but they are capable of doing so using simple image processing operations. Since it is known that light-field gradients are highly correlated with depth, this means that depth can be estimated from light-field data very efficiently, without necessitating the use of more costly reconstruction methods such as bundle adjustment. Typically, these sensors are modelled as cameras that record not only the direction from which a ray of light originated, as a monocular camera does, but also record the point on the aperture of the camera through which the ray passes. As these sensors are the primary sensors that are used in this research, more recent literature is covered in Chapter 2. However, a brief history of these sensors will be

provided here.

Arguably one of the earliest depictions of a light-field camera was produced in 1642 by the mathematician Mario Bettini in his work *Apiaria Universae Philosophiae Mathematicae*[1], see Fig. 1.9. Optics, and the ray theory of light, has an older history than this, dating back into antiquity with studies by Euclid, Ptolemy, and al-Haytham. Investigations into perspective geometry continued all throughout the Renaissance, primarily motivated by art [5]. Many centuries passed before the physicist Frederic E. Ives, in 1903, patented a device for recording parallax stereograms using a single focus lens by placing a screen in front of a photosensitive film in order to record two separate perspectives of a scene [40]. This work was followed by that of Gabriel Lippman who in 1908 published a paper describing a similar method, which he termed *'integral photography'*, whereby images could be formed by an array of lenslets each producing their own image of a scene from slightly different perspectives [57]. It was decades before the term *'light-field'* was introduced by Andrey Gershun in 1939 in a text that provided the first rigorous treatment on the subject [27]. Much later, in 1991, the term *'plenoptic'* was coined by Adelson and Bergen [1], and in 1992 the first lenslet-based light-field camera was constructed in the seminal work of Adelson and Wang [2]. Investigations into rendering light-fields from gantry setups followed in 1996 by Levoy and Hanrahan in work that also simplified the 7-dimensional plenoptic function of Adelson and Wang into the 4-dimensional two-plane parametrisation which is now standard [56].

While the concept of a light-field camera has emerged in various forms throughout history, it was not until 2005 that light-field cameras could be made small enough to be used by hand, in work by Ng *et al.*[74], the lead author of which went on to found Lytro, formerly[2] a manufacturer of lenslet-based light-field cameras. Before Lytro, however, Raytrix became the first manufacturer of lenslet-based light-field cameras with its founding in 2010, whereas Lytro was founded in 2012. Raytrix produces cameras whose designs are reminiscent of the work of Georgiev and Lumsdaine [26]. The camera design in that paper sacrifices the angular resolution of a light-field camera in order to obtain better spatial resolution by moving the microlens array further away from the imaging sensor than the focal length of the lenslets.

[1]The translation of this work is: *Beehives of Universal Mathematical Philosophy*, and was a textbook on assorted mathematical topics [5]. The text surrounding the Bettini's illustration reproduced within Fig 1.1 in that work describes its use as an optical illusion multiplying the apparent number of people present in front of it. While this is a remarkable depiction of what we would now call a light-field camera, it is unlikely that more modern insights into this construction were known to Bettini.

[2]Lytro ceased operations in 2018.

1.2.5 Scene Representations

As with the previous literature sections, it is beyond the scope of this thesis to survey all methods that have been used to represent scenes. An additional difficulty arises with reviewing scene representation methods, however, in that these methods are often not studied in isolation. Research describing new techniques for representing scenes are rarely published unless they also provide a new technique for solving the scene estimation problem using that representation. Nonetheless, numerous surveys and books exist that detail many of the more common scene representation methods [119, 95, 4, 12, 51].

Explicit Methods

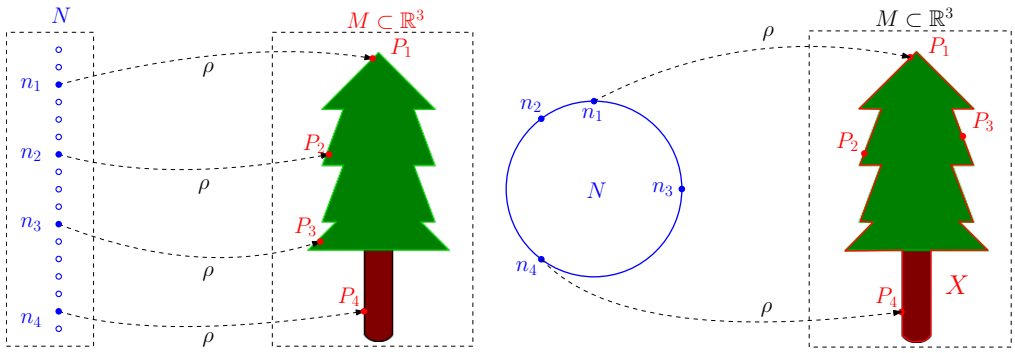


Figure 1.2: An explicit representation of a scene is a 'mapping to' the scene. In this figure a base space N is given, and the representation ρ maps each point n_i in N to a point P_i on the scene $X \subset M$. On the left, a sparse explicit representation of a scene is depicted as a point cloud, whereas on the right a dense representation of the scene is given as a parametrised surface.

An explicit scene representation is a method that is defined by a map from some parameter space to the scene. For an explicit method, given some points in the scene or some set of parameters of the scene, there is a simple way of constructing a sample of a point that lies in the scene. However, given a particular point in space, it may be difficult to determine whether that point is in the scene or not. An explicit representation is called *sparse* if its domain is discrete, and *dense* if its domain is continuous.

Point-clouds are typically seen as an example of a sparse explicit scene representation, as they represent the scene explicitly as a set of points. A point cloud simply consists of a list of points $X = (P_i)_{i=1}^N$. Another example of a scene representation is a triangle mesh. A triangle mesh is a tuple consisting of a point cloud X and

a list of faces $F = ((j_1, j_2, j_3)_i)_{i=1}^M$ where the entries j_1, j_2, and j_3 are required to be distinct integers between 1 and N. If (j_1, j_2, j_3) is an entry in F then the triangle with vertices P_{j_1}, P_{j_2}, and P_{j_3} is in the scene. However, it is more difficult to test whether an arbitrary point $P \in \mathbb{R}^3$ is in the triangle mesh or not. To do this we need to determine whether it lies on any of the triangles in the triangle mesh, and the only way to do this is by iterating over each of the faces. Both point clouds and triangle meshes have been used in computer graphics for many decades, and are usually discussed in textbooks on computer vision [102]. Often rendering of other surface representations first involves converting those representations into a triangle mesh through a process such as the marching cubes algorithm [59].

An example of a dense explicit scene representation is a parametrised surface. A parametrised surface is a mapping from some other parameter space into $\mathbb{R}^3$. Such surfaces are constrained by the topology of the domain. Nonetheless, such surfaces, depending on very few parameters, are among the most widely used methods in computer graphics and industrial design. A foremost example of a class of parametrised surfaces are those generated by B-splines, and in particular non-rational uniform basis splines (NURBS) [85].

A related but distinct type of explicit representation method is a surfel-based method. With a surfel-based method, a set of pre-generated surfaces, called surfels, are treated as primitives and are embedded into $\mathbb{R}^3$ to represent the scene [84]. The parameters of a surfel-based representation usually include the type of surfel used, and the pose of the surfel, but may also include parameters that involve scaling of the patch along various axes or rigid constraints between patches. A common type of surfel-based method represents a surface in terms of a set of discs.

Implicit Methods

An implicit scene representation is a method that consists of a test to determine whether a particular point is in the scene or not. For an implicit method, given a point in space it is simple to determine whether that point is in the scene or not, however it may be difficult to construct a sample of a point that lies in the scene. Just as is the case with explicit scene representations, an implicit scene representation is called *sparse* if its domain is discrete and *dense* if its domain is continuous.

For every point-cloud X there is a polynomial[3] in $\mathbb{R}[x, y, z]$ such that the roots of that polynomial are precisely X. This implies that every point-cloud representation can be represented implicitly as the roots of a polynomial, however in practice it is more efficient to simply iterate over every point in the point cloud to determine

[3]For example, one such polynomial is $f_X(Q) := \prod_{P \in X} ||P - Q||^2$.

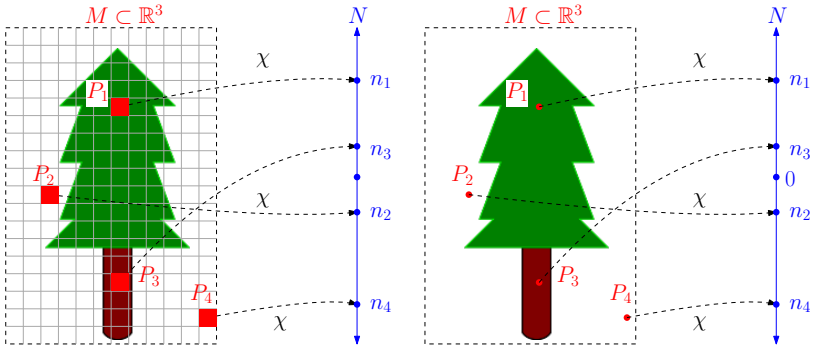


Figure 1.3: An implicit representation of a scene is a 'mapping from' the scene. In this figure a base space M is given, and the representation χ maps each point P_i in M to a point n_i in some image space N. The scene is then the set of points P such that $\chi(P)$ satisfies some criterion, such as $\chi(P) \geq 0$. On the left, a sparse explicit representation of a scene is depicted as an occupancy grid map, whereas on the right a dense representation of the scene is given as an extended characteristic function.

whether a given point is in it.

However, this example illustrates the usual way in which an implicit scene representation is defined. Implicit scene representations are typically given as level, sublevel, or superlevel sets of a specified function $\chi : \mathbb{R}^3 \to \mathbb{R}$. We call such a function χ an extended characteristic function. When χ is precisely 0 outside of the scene and precisely 1 on the scene, it is known as the characteristic function of the scene. Techniques for using level-set methods has an extensive literature [80], however the role of the function space and representation in scene reconstruction has not been completely explored.

Voxel-based methods are examples of sparse implicit methods because their domains are discrete. Typically, what is defined by a voxel-based method is either a characteristic function defined on a finite $3D$ grid of points, as in space-carving methods [53], or a conditional probability distribution defined on a grid of points. In the latter scenario, this approach is known as occupancy grid mapping [105], and has been used in robotics for decades.

The various approaches to implicitly defined surfaces involve different choices of parametrisations of the class of extended characteristic functions χ used. Common types of functions include distance functions, which map a point to the distance of the closest point on the scene, signed distance functions [64], which do the same except there is a notion of whether a point is 'inside' in which case the value is negative or 'outside' the scene in which case it is positive.

1.3 Problem Formulation

In this section, a theoretical framework for using light-field measurements to solve the scene estimation problem is introduced. Although the subsequent chapters are self-contained, this section provides a broader perspective on the work contained in those chapters.

This section begins with introducing the relationship between light-fields and scenes. Measurements of light-fields are called images, and their formation is given by the assignment of sensor elements to rays.

1.3.1 Scenes and Light-Fields

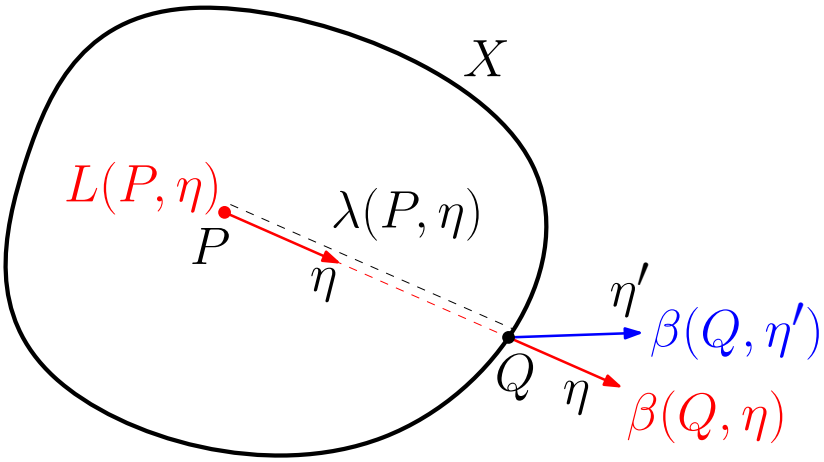


Figure 1.4: A light-field generated from a scene X. The distance $\lambda(P,\eta)$ of the scene X from point P in direction η is shown, and the point that lies at this distance in this direction is shown as Q. Under the ray constancy assumption, $L(P,\eta) = \beta(Q,\eta)$. Under the Lambertian assumption $\beta(Q,\eta) = \beta(Q,\eta')$ for all $\eta' \in S^2$.

A *coloured scene* is a pair (X,β) where X is a closed subset of $\mathbb{R}^3$ and β is a map called the *colouring*[4] that accepts as input a point $Q \in X$ and a direction $\eta \in S^2$ and returns as output a colour $c \in \mathcal{C}$, where $\mathcal{C}$ is a *colour space* (typically taken to be the unit cube $[0,1]^3$ representing RGB values). A set of coloured scenes is denoted $\mathbb{X}$ and is called a *coloured scene class*. The set X is called a *scene* and the set of all possible scenes is denoted $\mathbf{X}$.

A coloured scene determines a function $L_{(X,\beta)}$ the *light-field* generated by the coloured scene (X,β), whose domain consists of a subset of the set of all rays (when it is clear from context, the subscripts (X,β) are omitted), and whose codomain is $\mathcal{C}$. A *ray* is a pair $(P,\eta) \in \mathbb{R}^3 \times S^2$. The set of rays in $\mathbb{R}^3 \times S^2$ that forms the domain

[4] In some communities, this map is known as a *surface light-field*.

of the light-field $L_{(X,\beta)}$ is precisely those rays (P, η) for which there exists a positive real number α such that $P + \alpha\eta \in X$, and is called the *environment* (determined by the scene X), denoted by Σ_X (when clear from context, the subscript X is omitted). The intuition behind these defintions is that they capture the idea of a scene X being viewed from a point P in direction η. Because X is a closed set, the smallest such α for which $P + \alpha\eta \in X$ is a function of the pair (P, η), that is $\lambda_X(P, \eta) = \alpha$ for a map $\lambda_X : \Sigma \to \mathbb{R}^+$. Intuitively, the value $\lambda_X(P, \eta)$ is the shortest distance that must be travelled from point P in direction η before the scene is intersected. The map λ_X is canonically known as a *range map* (again, X is omitted when clear from context).

While a light-field L is simply a map from the set of rays that intersect the scene X to a colour space, the way in which a coloured scene determines a light-field depends on the sensor used and the context in which it is used. The most common constraint is that for all rays (P, η) in the environment Σ, we have that

$$L(P, \eta) = \beta(P + \lambda(P, \eta)\eta, \eta),$$

that is the colour seen from point P in direction η is determined by the colour of the closest point on the scene as seen from direction η. As a consequence of this constraint, we have that

$$L(P, \eta) = L(P + \alpha\eta, \eta)$$

for all $\alpha \leq \lambda(P, \eta)$. This captures the intuitive idea that the colour of a light ray viewed from P in direction η is given by the colour emitted by the closest point on the scene along this ray. This assumption, which is rarely explicitly stated[5] we will call the *ray-constancy assumption*. Another classical assumption is the *Lambertian assumption* that $\beta(P, \eta) = \beta(P, \eta')$ for all $P \in X$ and $\eta, \eta' \in S^2$, *i.e.* that the apparent colour of a point does not depend on which direction it is viewed from. Together, the ray-constancy assumption and the Lambertian assumption are the two most classical assumptions in computer vision.

1.3.2 Images

The problem of visual scene reconstruction is that of estimating the scene X from samples of the light-field L. Samples of a light-field typically take the form of an image. An *image* is a function $I : S \to \mathcal{C}$ whose domain S is called the *sensor plane*. For example, in a monocular camera, the sensor plane S consists of a finite grid of points in $\mathbb{R}^2$ and elements of this grid are called pixels. However, note that for ana-

[5]This assumption is usually only made explicit in its absence. In underwater settings, it is usually not assumed that the colour of light is constant along rays as it is here, as light-attenuation is a much more significant effect in that environment than in air.

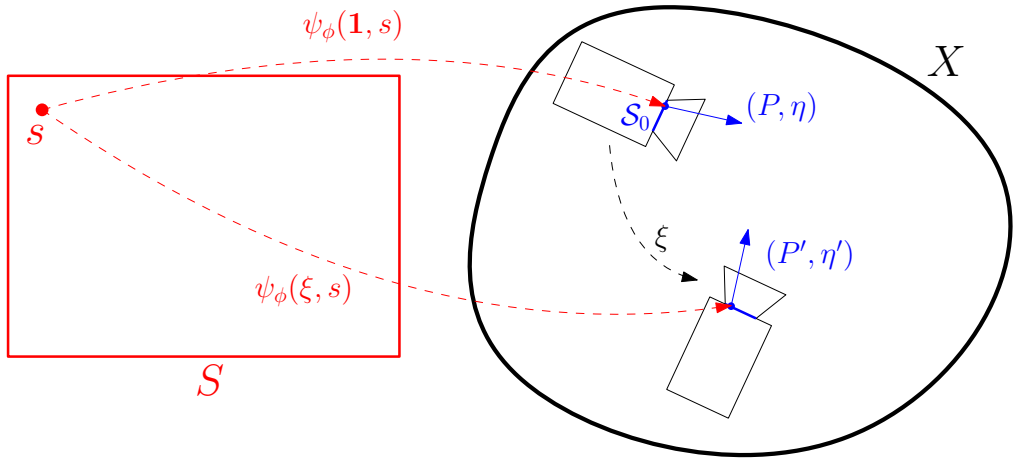


Figure 1.5: An example of a backprojection model. A scene X is shown together with the spatial part of the sensor bundle $\mathcal{S}_0$. Independent of these objects is a sensor plane S. The ray (P', η') assigned to the backprojection $\psi_\phi(\xi, s)$ is a rigid body transformation of the ray (P, η) assigned to the backprojection $\psi(\mathbf{1}, s)$ of the sensor element $s \in S$.

lytical purposes it is sometimes advantageous to assume a continuous sensor model rather than a discretised model, so that sensor elements are points in a rectangle in $\mathbb{R}^2$. The set of all possible images obtainable by that camera model is known as the *image space* and is a subset of the set of functions $\mathcal{C}^S$. In application, often constraints on the image space are imposed because a randomly selected image from $\mathcal{C}^S$ is overwhelmingly likely to be of random noise. It is the fact that real images have additional structure, and may be represented with far fewer parameters than is necessary to parametrise all of $\mathcal{C}^S$, that is exploited by image compression algorithms. These constraints on an image space often take the form of regularity assumptions on the image space. Points on a sensor plane are related to rays through a camera model. A camera model describes the relationship between images obtained by a camera and a light-field.

A *backprojection model* is a function $\psi_\phi : \Xi \times S \to \Sigma$ that depends on *intrinsic parameters* ϕ, which are parameters of the projection model that are constant for a particular camera, and takes as arguments an *extrinsic parameter* $\xi \in \Xi$ which is typically a pose, that is $\Xi \subset SE(3)$, and a sensor element $s \in S$. In this work, ξ will always be a pose, although other extrinsics are possible. However, to be mathematically correct, the set of possible poses will need to be restricted to the cases where $\psi_\phi(\xi, s) \in \Sigma$, and as such the domain Ξ may depend on the scene X. The backprojection model is defined by taking a fixed subset $\mathcal{S}_0 \subset \Sigma$, called the *sensor bundle*, that depends on the intrinsic parameters ϕ, and letting $\psi_\phi(\mathbf{1}, \cdot) : S \to \mathcal{S}_0$ be a bijective map, where

1 is the identity of $SE(3)$. A backprojection model satisfies $\psi_\phi(\xi, s) = \xi \cdot \psi_\phi(I, s)$, where the action of a pose ξ with translational part τ and rotational part R on a ray (P, η) is given by $\xi \cdot (P, \eta) = (RP + \tau, R\eta)$. An *image-formation model*[6] is the function $\mu := L \circ \psi : \Xi \times S \to \mathcal{C}$. For a fixed pose $\xi \in \Xi$, the function $I = \mu(\xi, \cdot)$ is the image of the scene as produced by the given image-formation model at the specified pose.

1.3.3 Projections

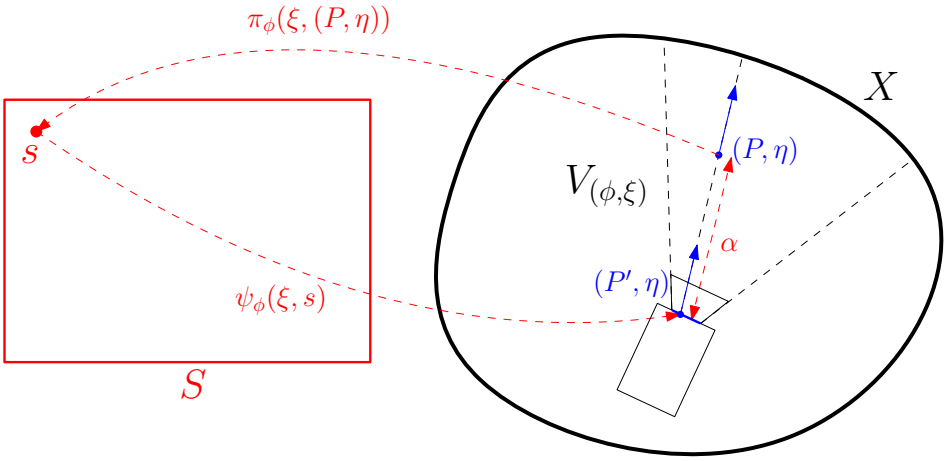


Figure 1.6: An example of a ray-projection model. A scene X is shown together with the set of rays $V_{(\phi,\xi)}$ that are visible to the camera. Independent of these objects is a sensor plane S. The ray (P, η) is forward-projected to the sensor element $s = \pi_\phi(\xi, (P, \eta))$. This sensor element is back-projected to the ray $(P', \eta) = \psi_\phi(\xi, s)$ on the sensor bundle. The original ray is given by $(P, \eta) = (P' + \alpha\eta, \eta)$, for some non-negative α, or alternatively $(P', \eta) = (P - \alpha\eta, \eta)$.

Several errors defined in the next sections cannot be defined properly without some mechanism for mapping rays and features to sensor elements, *i.e.* a forward-projection model that is compatible with the back-projection model. Given calibration parameters ξ and ϕ, and given (P, η), there exists an $s \in S$ and an $\alpha \geq 0$ such that $\psi_\phi(\xi, s) = (P - \alpha\eta, \eta)$ then s is said to be *imaging* (P, η). The set $V_{(\phi,\xi)}$ of all rays $(P, \eta) \in \Sigma$ for which there exists an $s \in S$ that is imaging (P, η) is called the *visible set*. A *ray-projection model* is a map $\pi_\phi : \Xi \times V_{(\phi,\xi)} \to S$ such that $s = \pi_\phi(\xi, (P, \eta))$ is always a sensor element imaging (P, η), *i.e.* such that if $s = \pi_\phi(\xi, (P, \eta))$ and $(P', \eta) = \psi_\phi(\xi, s)$ then there exists an $\alpha \geq 0$ such that $(P', \eta) = (P - \alpha\eta, \eta)$ (see Fig. 1.7).

[6] It should be noted that there is another hidden assumption here: that the camera actually captures the true colour of the scene. When it is not safe to assume this, *colour-calibration* of the camera must be performed to relate measured colours of rays to known colours of rays.

More generally, a feature-projection model is a map that sends a 'feature' to the collection of sensor elements corresponding to that feature. A *feature* is a set of rays that is given by a finite number of parameters. For example, a point is a feature and corresponds to the set of rays that pass through that point. Other examples include lines, rectangles, and planes. Typically, features are described by sets of points. Explicitly, a *feature-projection model* is a map $\Pi_\phi : \Xi \times F_\Sigma \to F_S$ where F_Σ is a finite-dimensional 'feature space' parametrising some specific type of *world-features* in the environment Σ, and F_S is a finite-dimensional feature space parametrising *image-features* that are sets of sensor elements corresponding to the world-features.

An important type of feature-projection model is a point-projection model. A point-projection model is a map that sends a point to all of the sensor elements that map to a ray that passes through that point. Specifically, let $F_{(\phi,\xi)}$ be the projection of the set $V_{(\phi,\xi)}$ into $\mathbb{R}^3$ under the projection $(P, \eta) \mapsto P$. Then a *point-projection model* is a map $\Pi_\phi : SE(3) \times F_{(\phi,\xi)} \to F_S$ such that for every sensor element s in the image-feature parametrised by $\Pi_\phi(\xi, P)$, if $(P', \eta) = \psi_\phi(\xi, s)$ then there exists an $\alpha \geq 0$ and an $\eta \in S^2$ such that $(P', \eta) = (P - \alpha\eta, \eta)$.

The development of accurate sensor models is essential for scene reconstruction tasks. Sometimes, as with light-field cameras, it is possible to extract scene geometry information from the measurements of the light-field produced by that scene, but only under specific conditions on the class of scenes to be estimated. These questions will be examined in more detail in Chapter 2, and the following section introduces the precise meaning of 'scene class'.

1.3.4 Representations

In order for a scene reconstruction to be implementable on a computer, the scene must be representable by a finite number of parameters, even though the potential number of parameters that may be used in a specific representation is unbounded. This is a problem however, because the set of closed subsets of $\mathbb{R}^3$ is not countable, as the set of scenes must be. For this reason, every implemented scene estimation method entails assumptions about the scene on top of those used for the sensor and light propagation models. These assumptions take the form of a *scene class* $\mathbf{X}$, that the scene estimates are constrained to lie within. This assumption has two components to it: the inclusion-relation of the estimate to the true scene, and whether the true scene is a member of the scene class. Firstly, it may be the case that only a subset of the scene is estimated, as with a point-cloud, or that a superset of the scene is estimated, or that precisely the scene is estimated. Secondly, it may be the case that the scene estimate does not belong to the same scene class as the true scene, but that

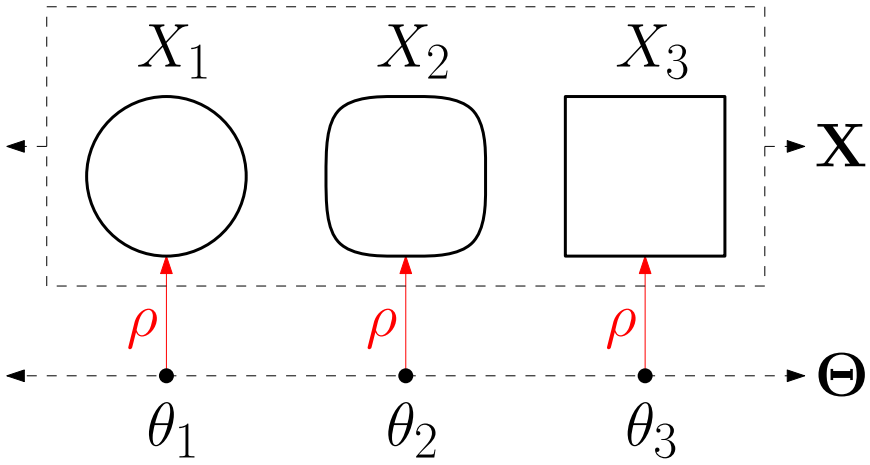


Figure 1.7: An example of a scene representation. A scene class $\mathbf{X}$ is shown together with a parameter space $\mathbf{\Theta}$. A parameter $\theta_i \in \mathbf{\Theta}$ is mapped by the representation ρ to a scene X_i in the scene class.

for any desired degree of accuracy, there is a scene estimate $\hat{X} \in \mathbf{X}$ that approximates the true scene X to that degree of accuracy (the notion of accuracy will be made formal later).

Distinct from the choice of scene class $\mathbf{X}$ is how to parametrise this scene class. A *scene class representation* is a surjective mapping ρ from some set of admissible sequences $(\theta_i)_{i=0}^{\infty}$ of real or complex numbers to the scene class $\mathbf{X}$. The set of admissible sequences $\mathbf{\Theta}$ is known as the *parameter space*. If the representation is injective, it is called *faithful*. If there exists some $n \in \mathbb{N}$ such that for every $i > n$, the map ρ does not depend on θ_i, the representation is finite-dimensional, otherwise it is infinite-dimensional. The (*n-th*) *truncation* of a scene representation is a map which takes a sequence $(\theta_i)_{i=0}^{\infty}$ and sets all but the first n parameters to 0, leaving the first n entries unchanged. The choice of representation does not in itself constitute an assumption on the true scene X, as it is already assumed to reside in the scene class $\mathbf{X}$ that the representation maps to. However, if the representation used in an implementation is always truncated, as must be done in any real-world application when the representation is infinite-dimensional, a different choice of representation will result in a different scene class.

Ideally, a representation should be a continuous map. However, in order to define continuity of this map, a topology on both the scene class $\mathbf{X}$ and parameter space $\mathbf{\Theta}$ must first be specified. Typically, the parameter space is taken to be either finite, in which case it inherits a Euclidean topology, or consists of ℓ^P sequences for some $P \in \mathbb{N}$, in which case it is a metric space. Defining a topology for the scene class $\mathbf{X}$ without reference to the representation involves more technicalities, but may be

done. However in some situations, it suffices to have $\mathbf{X}$ coinduce its topology from the representation ρ, and the open sets of $\mathbf{X}$ consist of the sets $O \subset \mathbf{X}$ such that $\rho^{-1}(O)$ is open in $\mathbf{\Theta}$.

1.3.5 Distances and Errors

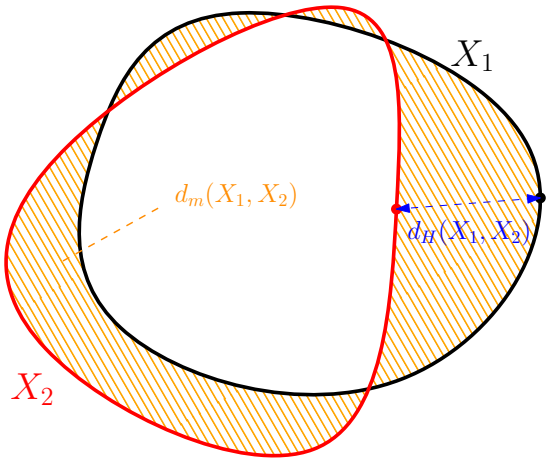


Figure 1.8: An illustration of two measurable compact scenes of nonzero measure, X_1 whose boundary is shown in black and X_2 whose boundary is shown in red. The largest minimum distance between any two points in the two scenes is known as the Hausdorff distance $d_H(X_1, X_2)$ and is shown in blue in this example. The total measure of the symmetric difference between the two sets, illustrated in orange, is $d_m(X_1, X_2)$.

It is also useful in an optimisation setting to define some metric, or if that is not possible, some notion of 'distance' on the scene class $\mathbf{X}$. At this point it should be briefly noted that any set of non-empty bounded subsets of a metric space can be made into a metric space by using the *Hausdorff distance* as a metric [39]. Intuitively, the Hausdorff distance between two sets is the largest distance between any two points in either set. Explicitly, if M is a space with metric d, and X_1 and X_2 are two bounded subsets of M, the Hausdorff distance d_H is given by:

$$d_H(X_1, X_2) := \max\Big(\sup_{P_1 \in X_1} \inf_{P_2 \in X_2} d(P_1, P_2), \sup_{P_2 \in X_2} \inf_{P_1 \in X_1} d(P_1, P_2)\Big).$$

The first argument of the 'max' function is the largest distance between a point in X_1 and the set X_2, and the second argument is the largest distance between a point in X_2 and the set X_1. Note that if the first argument is 0, then $X_1 \subseteq X_2$ and if the second argument is 0 then $X_2 \subseteq X_1$. Alternative classical notions of convergence of sets include Wijsman convergence [114], which extends the notion to unbounded

sets, and Kuratowski convergence [49].

An arguably more important measure of distance between sets – and one that is used more intensively in this work – comes from their measures. The reason that we state that this notion of distance is more important in this work than Hausdorff distance comes from an optimisation perspective. Attempting to minimise Hausdorff distance between two sets will only ever focus on the outlier points in the sets at each optimisation step, whereas these outliers are ignored in a measure-derived distance.

If $(M, \mathcal{A}, m)$ is a measure space and $\mathbf{X} \subset \mathcal{A}$ is a scene class consisting of measurable sets of finite and non-zero measure (under measure m), and if $X_1 \in \mathbf{X}$ and $X_2 \in \mathbf{X}$, then we can define the distance $d_m(X_1, X_2)$ between X_1 and X_2 as the measure of the symmetric difference of these sets:

$$d_m(X_1, X_2) := m((X_1 \setminus X_2) \cup (X_2 \setminus X_1)).$$

Note that this distance requires the sets to be of *non-zero* measure. Further, this suggests that this distance is only affected by the portions of the scene that have volume. Denoting $\tilde{X}$ as the closure of the interior of X, it can be observed that

$$d_m(X_1, X_2) = d_m(\tilde{X}_1, \tilde{X}_2).$$

We will call $\tilde{X}$ the *erosion*[7] of X. If $X = \tilde{X}$, the set X is called *eroded*. Eroded sets are closed sets that do not contain portions that could be considered surfaces, curves, or points in space unless those portions are contained in a volume. When the scene class $\mathbf{X}$ consists only of eroded (or open[8]) and bounded sets, the pair $(\mathbf{X}, d_m)$ is a metric space. Because all point clouds erode into the empty set, this notion of distance is not applicable in that case. Note that d_m may also be expressed as an integral. If χ_1 and χ_2 are the indicator functions of the sets X_1 and X_2, then we have that

$$d_m(X_1, X_2) = \int_M |\chi_1(x) - \chi_2(x)| \, dx.$$

The two notions of distance introduced are the only definitions used in this work that do not rely on the extrinsic structure of scenes. An additional notion of distance comes from the representation of the scene. If θ_1 are the parameters of X_1 and θ_2 are the parameters of X_2 under a faithful representation ρ, and if $\mathbf{\Theta}$ is a normed space, then the distance between X_1 and X_2 is given by $||\theta_1 - \theta_2||$.

These distance measures are useful when investigating the theoretical properties of a scene reconstruction. However, these distances rely on complete knowledge of

[7] This term is inspired by a morphological operation of the same name used in image processing.
[8] Because a scene is a closed set, eroded is used in this work.

both of the scenes being compared. In scene reconstruction tasks, the true scene is of course unknown, and so any attempt to minimise the distance between an estimate and a scene directly cannot be implemented. Instead, we rely on minimising *errors.* An error is a measure of the difference between the outputs of a coloured scene. There are two main types of errors: photometric errors, and reprojection errors. An error that is fundamentally defined between objects that inhabit the sensor plane S of a sensor is known as a *reprojection error*. When the error consists of a difference between objects in the image space I, the error is known as a *photometric error*.

1.3.6 Calibration

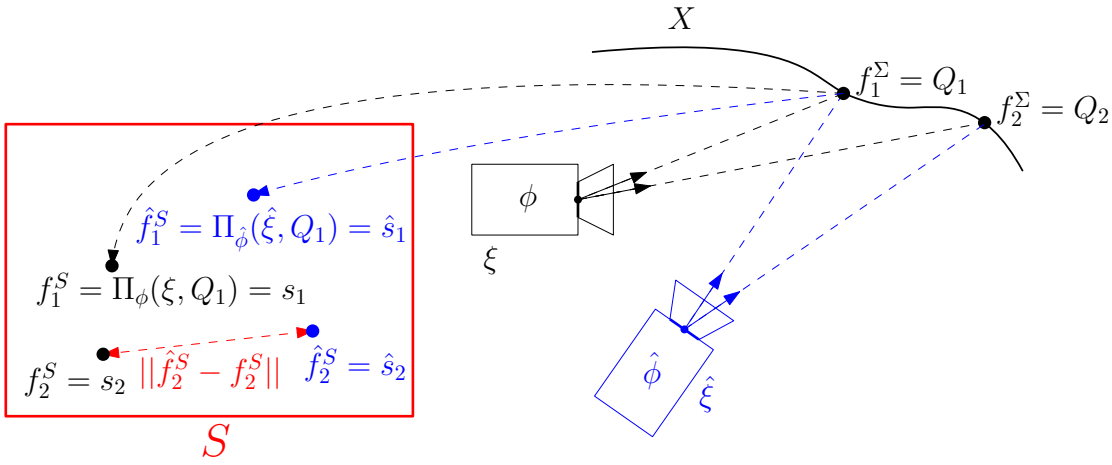


Figure 1.9: An illustration of a standard process for calibration of a monocular camera. Both the world-feature points Q_1 and Q_2 and the image-feature points s_1 and s_2 corresponding to them are known. The intrinsics ϕ and extrinsics ξ of the camera need to be estimated. Given the known locations of the points Q_1 and Q_2, we can, given an estimate $(\hat{\phi}, \hat{\xi})$ of the calibration parameters, produce the image-feature point estimates $\hat{s}_1$ and $\hat{s}_2$. The difference between the estimated image features and known image features determines an error function that may be minimised with respect to the calibration parameter estimate.

Calibration is the task of estimating from a sequence of images $\{I_t : t \in T\}$ the intrinsic parameters Φ and extrinsic parameters ξ_t of the camera associated with each image. The pair $(\Phi, \{\xi_t : t \in T\})$ is known as the calibration parameters associated with the images. Usually, the focus of calibration is on obtaining the intrinsic parameters rather than the extrinsic parameters because the intrinsic parameters are constant across all images obtained with a given camera model as long as the optical settings of the camera are held constant. However, unless the extrinsics of the camera are carefully controlled and measured independently, these parameters will also need to be estimated during calibration.

When the calibration task involves estimating the extrinsics, there are two main approaches: calibration from a structured scene, and calibration from an unstructured scene. Calibration from a structured scene is a procedure whereby parameters of the scene corresponding to the images I_t are known beforehand and are used in the calibration procedure. Calibration from an unstructured scene is also known as *autocalibration*, and is a procedure whereby no information about the scene being imaged is known beforehand and treated as a prior in the calibration task.

A typical method of camera calibration involves the use of a structured scene consisting of a finite number of feature points. The most common type of structured scene used in camera calibration is a checkerboard, and the identified features in this setting are usually the corners of the checkerboard. These scenes are used because there are dedicated *feature-extraction* algorithms that are able to accurately identify the corners of a checkerboard. Thus, from a structured scene, we are able to estimate from each image I_t a set of image-features $\{f^S_{(i,t)}\}_{i=1}^n$ that correspond with known coordinates of n world-features $\{f^\Sigma_i\}_{i=1}^n$. These estimates of the image-features are usually treated as data because the feature-extraction techniques are often highly reliable when lighting and other factors are controlled.

There is often a closed-form solution for most of the calibration parameters from the pairing of image-features and world-features when some of the other parameters are held constant at known values. This closed-form solution is refined and the other parameters of the backprojection model are estimated by performing an optimisation procedure that minimises reprojection error. This task is as follows: given an estimate $\hat{\Phi}$ of the intrinsics and an estimate $\hat{\xi}_t$ of the extrinsics for each image, we compute the expected image-features $\hat{f}^S_{(i,t)}$ corresponding to each of the world-features $\hat{f}^\Sigma_i$ by using a feature-projection model $\hat{f}^S_{(i,t)} = \Pi_{\hat{\Phi}}(\hat{\xi}_t, f^\Sigma_i)$. The optimisation is performed by varying the parameters $\hat{\Phi}$ and $\{\hat{\xi}_t : t \in T\}$ so that the reprojection error $\sum_{t\in T} \left|\left|\hat{f}^S_{(i,t)} - f^S_{(i,t)}\right|\right|^2$ is minimised. This error is an ℓ^2 reprojection error, although other types of reprojection errors have been used to minimise the influence of outliers in the image-feature data.

This general framework is explored further and applied to light-field cameras in Chapter 3.

1.3.7 Scene Reconstruction

By *scene reconstruction* we mean a method of producing an estimate $\hat{X}$ of the true scene X from a set of measured input-output pairs $\{(\xi_t, I_t) : t \in T\}$, where $I_t := \mu(\xi_t, \cdot) : S \to \mathcal{C}$, and $T \subset \mathbb{R}$ is some index set that may either be continuous or discrete. If the scene X is estimated perfectly, so that $\hat{X} = X$, and if the colouring β

is also estimated perfectly, new images I of the scene can be generated from novel perspectives that were not in the original dataset that would be equal to real images captured from those perspectives. It should be emphasised that generation of arbitrary images of a scene is possible only if the *total state* (X, β) is estimated, rather than only the *partial state* X.

When the poses ξ_t from which the image data $\{I_t : t \in [0, T]\}$ were generated is unknown, the problem of simultaneously estimating ξ_t and X is known as *Visual Simultaneous Localisation and Mapping (VSLAM)*. More generally, the problem of estimating both the poses ξ_t and the scene X from some sequence of sensor measurements $\{\mu_t : t \in [0, T]\}$ is known as *Simultaneous Localisation and Mapping (SLAM)*[9]. Solving a VSLAM problem always requires stronger constraints than solving a (visual) scene reconstruction problem[10]. Most SLAM implementations represent scenes with a discrete point cloud. At this level, there is no mathematical difference between the definition of VSLAM and structure-from-motion (SfM) other than that a VSLAM method typically requires the set of input images to be ordered and SfM does not. In practice, VSLAM algorithms may have more importance assigned to computational efficiency.

This thesis focuses on scene reconstruction, where the poses of the camera are estimated using techniques separate from those used to estimate the scene. In particular, Chapters 4 and 5 develop different techniques for estimating scenes using light-field measurements using both explicit and implicit representation methods, respectively. The methods developed in those chapters are both instances of observers, which are introduced in the following section.

1.3.8 Observers

A *dynamical system* is a triple $\mathbb{S} = (\mathbb{T}, \mathbb{W}, \mathcal{B})$, where $\mathbb{T}$ is called the *time axis*, $\mathbb{W}$ is called the *signal space*, and $\mathcal{B}$ is a subset of $\mathbb{W}^{\mathbb{T}}$ called the *behaviour* [86]. The behaviour consists of all the possible ways in which the system may evolve over time. For this work we will assume that $\mathbb{W} = \mathbb{W}_1 \times \mathbb{W}_2$ and that both $\mathbb{W}_1$ and $\mathbb{W}_2$ are metric spaces. We will also use the following notation: if $\mathbb{W}$ is a finite product of some other signals so that $\mathbb{W} = \prod_{i=1}^{N} \mathbb{W}_i$, then for $w = (w_1, \cdots, w_i, \cdots, w_N) \in \mathbb{W}$, $\pi_i(w) = w_i$. A (global and asymptotic) observer for $\omega_2 \in \mathbb{W}_2$ from the variable $\omega_1 \in \mathbb{W}_1$ is a dynamical system $\mathbb{O} = (\mathbb{T}, \mathbb{W}_1 \times \mathbb{W}_2, O)$ satisfying the following. For

[9]There are community differences on the definition of SLAM. Some believe that a technique should only be called SLAM if an offline optimisation procedure known as *loop closure* takes place to account for odometry drift.

[10]For example, certain scenes with repetitive patterns cannot be estimated in VSLAM but can be reconstructed if poses are known.

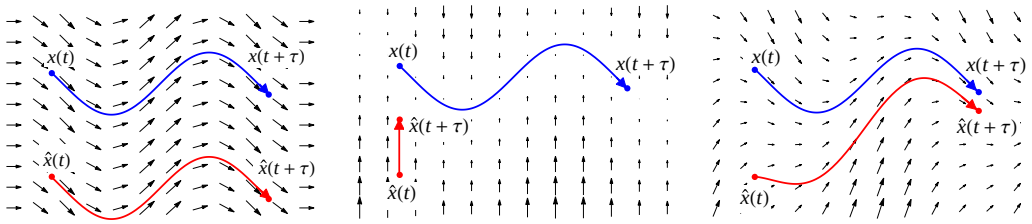


Figure 1.10: A standard observer design is given adding the dynamics of an internal model, shown left, which updates a point estimate based on its known dynamics, together with an innovation term, shown middle, which minimises the error in the estimate in response to system measurements. The resulting observer dynamics are shown right.

any $b \in \mathcal{B}$ and any[11] $w_2^0 \in \mathbb{W}_2$ there exists an $o \in O$ such that:

1. $\pi_2(o(0)) = w_2^0$,
2. For all $t \in \mathbb{T}$, $\pi_1(o(t)) = \pi_1(b(t))$,
3. $\lim_{t\to\infty} d(\pi_2(b(t)), \pi_2(o(t))) = 0$.

Intuitively, the first condition is a globality condition meaning the desired property of a system trajectory does not depend on its initial conditions, the second condition states that the observer dynamics may depend on variable w_1 representing measurements of either the inputs or outputs of the system to be observed, and the third condition is an asymptotic convergence condition that specifies the desired property of the observer. The *internal model principle* states that if $\mathbb{O}$ is an observer for w_2 from w_1 then $\mathcal{B} \subseteq O$. There is active research to determine the classes of systems for which the internal model principle holds [108].

In practice, observers and systems are typically governed by differential equations and are time-invariant. Usually such a dynamical system is governed by equations of the form

$$\dot{x}_t = f(x_t, u_t) \tag{1.1}$$
$$y_t = g(x_t, u_t), \tag{1.2}$$

where $u_t \in U$ is some known function called the *input*, $y_t \in Y$ is some variable that is measured by a sensor and is called the *output*, and $x_t \in X$ is a collection of variables that represent the relationship between the input and output at a particular

[11] For many situations global convergence of the state estimate to the state is not achievable. In such situations weaker notions than global convergence are necessary.

time called the *state*. The set U is known as the *input space*, and the set Y is known as the *output space*, and both are usually treated as Euclidean spaces or subsets of them. The set X is known as the *state space* and is usually a manifold. We assume that the function f has nice enough properties to guarantee existence and uniqueness of the function x_t whenever any unknown initial condition x_0 is provided. A *state observer* is a second dynamical system of the form

$$\dot{\hat{x}}_t = h(\hat{x}_t, u_t, y_t), \tag{1.3}$$

that satisfies

$$\lim_{t\to\infty} d(\hat{x}_t, x_t) = 0 \tag{1.4}$$

no matter what the initial condition $\hat{x}_0$ or input signal u_t is, except perhaps on some submanifold of zero measure. The variable $\hat{x}_t \in X$ is known as the *state estimate*, but it is common to refer to this variable as the observer itself.

Often, observers are constructed from knowledge of the dynamics f of the system and the construction of an error function. An *error function* is a function

$$\epsilon : X \times Y \times U \to \mathbb{R}_+$$

for which, given any input-output pair (u_t, y_t) the function

$$\epsilon(\cdot, y_t, u_t) : X \to \mathbb{R}$$

has exactly one local minimum, which is at x_t, and $\epsilon(x_t, y_t, u_t) = 0$. A common type of observer is a *gradient observer*. A gradient observer is an observer where the function h in (1.3) satisfies

$$h(\hat{x}_t, u_t, y_t) := f(\hat{x}_t, u_t) - \nabla_1\epsilon(\hat{x}_t, y_t, u_t) \tag{1.5}$$

for some differentiable error function ϵ. Here $\nabla_1\epsilon(\hat{x}_t, y_t, u_t)$ refers to the gradient with respect to the first argument alone, so that this object is an element of $T_{\hat{x}_t}X$. A gradient observer always satisfies the internal model principle.

In order to model a given dynamical system, it is often easier to first model the state of that system as inhabiting a general system. A *system* is a pair $(\mathbb{W}, \mathcal{B})$ where $\mathbb{W}$ is a set of easily-definable objects, and $\mathcal{B}$ is simply a subset of $\mathbb{W}$ called the behaviour of the system. Once a system is defined, a dynamical system is usually characterised by a certain set of parametrised paths in the more general system, subject to some constraints. For the case of scenes, given intrinsic camera parameters

ϕ, a *light-field system* is a pair $(\mathbb{X} \times I \times SE(3), \mathcal{B})$, where the behaviour $\mathcal{B}$ consists of the subset of elements $((X, \beta), I, \xi) \in \mathbb{X} \times I \times SE(3)$ such that:

1. $\xi \in \Sigma_X$, and
2. $I = \mu(\xi, \cdot) = L_{(X,\beta)} \circ \psi_\phi(\xi, \cdot)$,

where $L_{(X,\beta)}$ is the light-field generated by the coloured scene (X, β), *cf.* Sections 1.3.1 and 1.3.2.

A *light-field dynamical system* is a triple $(\mathbb{T}, \mathbb{X} \times I \times SE(3), \mathcal{D})$, and at time t the 'input' into the system is the pose $\xi_t \in SE(3)$, the 'output' is the image $I_t \in I$ and the state is a coloured scene $(X, \beta) \in \mathbb{X}$. Typically, a light-field dynamical system is characterised by paths in a light-field system parametrised with respect to time $\mathbb{T}$, and so the behaviour $\mathcal{D}$ is given by the set of functions $w \in \mathbb{W}^{\mathbb{T}}$ such that $w_t \in \mathcal{B}$, where $\mathcal{B}$ is the behaviour of a light-field system. Other constraints on the system such as continuity of w with respect to t may also be applied.

1.4 Thesis Structure

This thesis examines the scene estimation problem from the perspective of systems theory and observer design. The primary sensor used in this thesis, which will be examined in detail in Chapter 2, is the light-field camera. The remainder of this thesis is divided into two parts.

In Part I, the geometry of the sensor model that is used in the scene estimation task is examined in detail. The sensor used in this work is the light-field camera, and its geometry, and projection model is examined in detail in Chapter 2, and is based on the works [79] and [78]. The projection model derived is not the standard ray-projection models provided in other works but a point-projection model. The relation between the parameters of this point-projection model and a ray-projection model is provided. A solution to the depth estimation problem using light-field camera data is also derived, and necessary and sufficient conditions for this estimation method are provided. Chapter 3 uses this camera model to derive a calibration procedure for estimating the parameters of the camera model derived in Chapter 2. The resulting calibration procedure produces better results than previous methods on a variety of metrics and these results were published in [79].

Part II of this thesis investigates the design of observers for estimating scenes from light-field data using both an explicit and implicit representation of scenes. Chapter 4 is based on previously published work [77] where an observer is designed that estimates scenes using an explicit point cloud representation. The observer is

tested in simulation and the asymptotic convergence of each point in the point cloud estimate to a point on the scene is proved analytically. The resulting proof requires significant topological considerations to avoid edge cases, as the trajectories of points depend on the motion of the camera and the update function is a switching system. In Chapter 5 an observer is designed that instead uses an implicit representation on a variety of different function spaces, published in [76]. Not only is this representation a dense representation of the scene, the assumptions of the scene are weaker than in the explicit case, and the convergence is strengthened from asymptotic to finite-time.

Part I

Sensor Models and Calibration

Chapter 2

Plenoptic Geometry and Disparity Estimation

This chapter develops the sensor model for the light-field camera that is partially based on the findings in [79] and [78], although some of the work in this chapter is new at the time of writing. An accurate sensor model is essential to relate the output produced by a sensor to the output of the system that the sensor is measuring. In this thesis, the system that is being measured is a scene together with the colouring defined on it (in this thesis this pair is called a *coloured scene*), the output of this system is the plenoptic function, the sensor measuring this output is a light-field camera, and the output of the sensor is a raw light-field. This chapter provides a projection model for the camera that maps points on the scene to 'plenoptic disc features' that can be readily estimated from the raw light-field images. This projection model depends on a number of parameters, and a method for estimating these parameters in provided in Chapter 3.

Before feeding a sequence of raw light-fields directly into an observer framework, it is worth considering how much information about the scene is already contained in a single light-field, and whether there are efficient techniques for extracting this information. Such considerations lead naturally to the concept of disparity. Disparity estimation can be performed efficiently on light-field data by exploiting the natural geometry that arises in this data, and an efficient technique for estimating disparity is provided in Section 2.5.

2.1 Background

Understanding of light-field geometry has evolved in recent decades. There are two main types of projection models for a light-field camera: a point projection model and a ray projection model. It is the fact that points do not map to individual rays through the projection model that distinguishes light-field cameras from monocular

cameras.

Early work on light-field geometry typically adopted a ray projection model that related rays of light in front of the camera to the data produced by the camera. The seminal work in this area introduced the standard two-plane parametrisation of light-fields in which one plane defines the perspective from which an image is taken, and another plane to specify a pixel within that image [56]. This projection model was developed further by Dansereau *et al.* [22] by the introduction of camera matrices reminiscent of those present in traditional works on monocular and stereo camera calibration [33].

In this chapter, a point-to-feature projection model is treated as the elementary relation between the output of a coloured scene and the data produced by a light-field camera. Because the aperture of a light-field camera is circular, these features will also be circular. This observation was first noted in [83], but the projection model in that work was ray-based and point depths were computed by minimising an error function over the set of rays that have been determined to pass through the same point. The advantage of the point-to-feature projection model is clear in 3D reconstruction tasks where the goal is to estimate the location of points on a scene. The point projection model for light-field cameras is invertible, unlike with a monocular camera where the primitive is a perspective projection and depth information is lost.

Verifying that a model of the camera is plausible is typically conducted by calibration. Calibration, discussed in more detail in Chapter 3, is the task of minimising the error between real data generated by a sensor and the prediction of what the data should be given a sensor model. The minimisation is conducted with respect to the parameters of the chosen model for that sensor, and these parameters are called the calibration parameters of the model.

Because a light-field camera images a scene from a large number of slightly-varying perspectives, it is possible to estimate depth without feature-matching by instead estimating the infinitesimal change in an image due to an approximately infinitesimal change in perspective. This allows for a dense estimate of depth. The use of light-field cameras for the purpose of depth estimation was first considered by Adelson and Wang [2]. That work contains several major contributions, including the first design of a lenslet-based light-field camera wherein an array of lenslets are placed in between an imaging sensor and a focus lens. A second major contribution of that work was the observation that the disparities of points on the recorded images, and hence their depths, could be obtained by calculating light-field gradients.

Estimating depth from light-fields is a well-established area of research [37, 117, 42, 46, 112, 129, 99, 110, 97, 96, 20]. A good survey of the state-of-the-art is given by Johannsen *et al.* [45], and the rapid development of this area has led to the release of

several benchmarking datasets and toolkits in recent years [37], [113].

The conventional framework for light-field based disparity estimation is based on constructing cost-volumes. A per-pixel cost function is constructed that attains its minimum at correct disparities. A cost-volume is constructed by varying the disparity hypothesis over some range of values. An optimal surface within this cost volume is determined by minimising an energy functional, that may depend on other parameters as in graph-cut based approaches. Approaches that follow this framework are robust and accurate. However the construction of the cost-volume is a computationally expensive procedure and many of the more accurate methods can take over half-an-hour to compute a single disparity map, as reported by the authors of those methods on the benchmark [37].

Learning-based approaches are computationally faster [96], although this is only the case when training time is not taken into account. A recent method reports 6 days required for training on optimised hardware [96]. There are also no theoretical guarantees that a training-based method will generalise well to situations radically different from the data it was trained on.

This chapter presents several contributions to the understanding of light-field geometry including:

1. A novel projection model that relates points imaged by the camera to *plenoptic disc features* present in the raw camera data, as previously reported in [79].

2. The discovery of a partial differential equation that all disparity maps must satisfy, as previously reported in [78].

3. A mathematical proof of the necessary and sufficient conditions that a colouring must satisfy in order for depth to be estimated from first-order light-field data.

2.2 Physical Camera Model

In this section, we formulate a projection model of the camera based on its physical optics.

2.2.1 Projection Through a Thin Lens

In this sub-section we express all points in the body-fixed frame **C** of the camera. A point P expressed in this frame has coordinates (P^x, P^y, P^z). We model the focus lens positioned in front of the micro-lens array (MLA) as a thin lens. For thin lenses, every point P on one side of the lense corresponds to another point Q, for which all

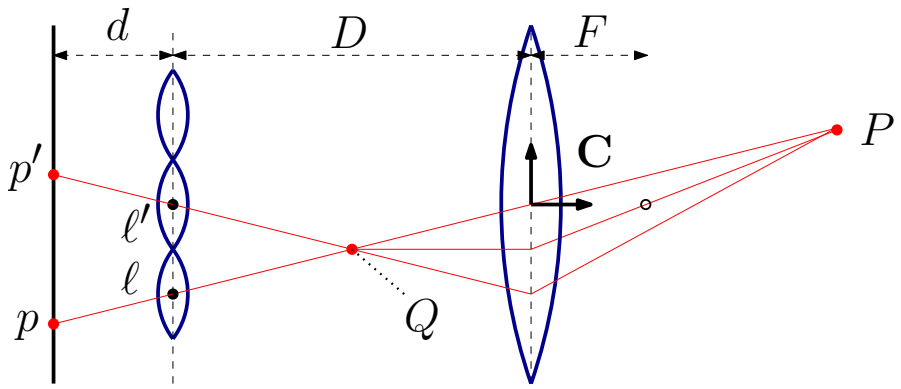


Figure 2.1: A point P with image point Q is shown. Two lenslets ℓ and ℓ' are shown with the pixels p and p' of the perspective projections of the point Q through each respective lenslet.

the rays of light passing through P pass through Q and vice-versa (see Fig. 2.1). The point Q is called the *image point* of P. For points P in front of the camera, so that $P^z > 0$, we have that the image point Q satisfies

$$\frac{1}{F} = \frac{1}{P^z} - \frac{1}{Q^z}, \tag{2.1}$$

where F is the focal length of the focus lens. Because the image point Q always lies on a line passing through P and the optical centre, we can determine the position of Q to be given by

$$Q = \left(\frac{F}{F - P^z}\right) P. \tag{2.2}$$

2.2.2 Projection Through a Micro-Lens Array

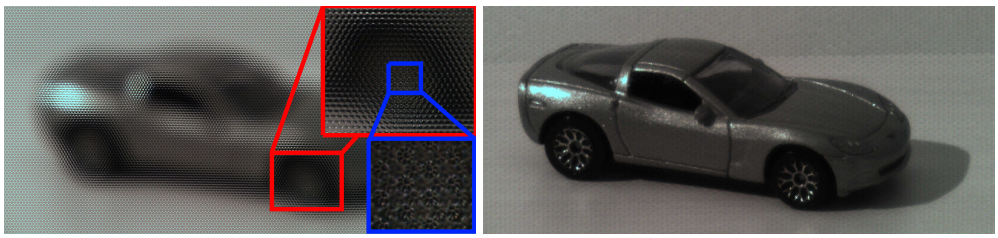


Figure 2.2: (Left) A raw light-field image of a scene. Zoomed portions of this raw data are highlighted to show the image consisting of thousands of densely-packed lenslet images each consisting of hundreds of pixels. (Right) An image extracted from the raw data.

Lenslet-based light-field cameras are constructed by positioning a micro-lens array between a focus lens and an imaging plane. The raw data of a light-field cam-

era appears as an hexagonal array of smaller circular *subimages* stitched together (Fig. 2.2). A subimage in the raw data is the image produced by a single lenslet ℓ. The *pixel image coordinates* (p^u, p^v) of a pixel p are counted positively from the top left corner of the raw light-field image. We assign to each lenslet ℓ *lenslet image coordinates* $(\ell^u, \ell^v) \in \mathbb{R}^2$ given by the image coordinates of the apparent centre of the lenslet on the raw image.

Lenslets ℓ and pixels p are also represented by their *physical coordinates* expressed in the body-fixed frame $\mathbf{C}$ of the camera, denoted $\ell = (\ell^x, \ell^y, \ell^z)$ and $p = (p^x, p^y, p^z)$, respectively. We assume that the MLA is parallel to the main focus lens of the camera, so that all the lenslets have a constant displacement $\ell^z = -D$, and so $\ell = (\ell^x, \ell^y, -D)$, see Fig. 2.1.

The physical coordinates ℓ of the optical centre of the lenslet are related to its lenslet image coordinates (ℓ^u, ℓ^v) by the equation

$$\ell = \left(S^u \frac{D}{D+d}(\ell^u - c^u), S^v \frac{D}{D+d}(\ell^v - c^v), -D \right). \tag{2.3}$$

In this equation, S^u and S^V are the physical scales of the MLA in metres per pixel in the u and v directions respectively, (c^u, c^v) are the pixel coordinates of the optical centre of the camera, and d is the distance between the MLA and the imaging plane. When $S^u = S^v$, we instead use the parameter S. The parameter $\frac{D+d}{S}$ is often referred to as f in other papers [75, 9], and in the literature is called the "focal length" of a pinhole camera model for the lenslet. However, the physical meaning of this parameter should not be confused with that of the focal length of a thin-lens.

Under the assumption that pixels are at a constant distance d from the micro-lens array, the physical coordinates of a pixel p with image coordinates (p^u, p^v) are given by

$$p = (S^u(p^u - c^u), S^v(p^v - c^v), -D - d). \tag{2.4}$$

Given physical coordinates ℓ of a lenslet and an image point Q, the location of the pixel that images Q through the lenslet ℓ is found by determining where the line passing through Q and ℓ intersects the pixel plane (Fig. 2.1). Using a similar-triangles argument, p is given by

$$p = \frac{d}{D + Q^z}(\ell - Q) + \ell. \tag{2.5}$$

The image coordinates of p are then found by solving (2.4) for (p^u, p^v).

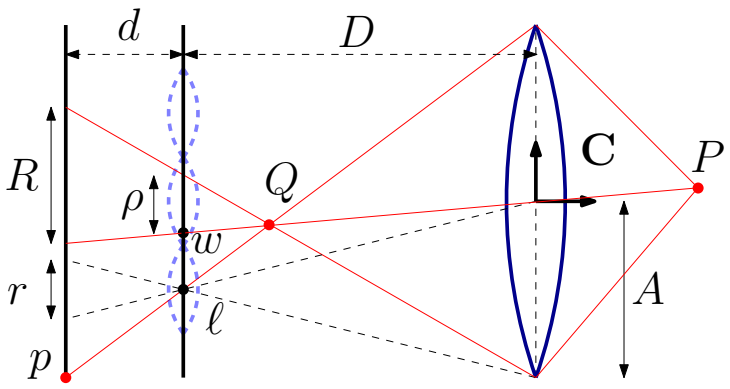


Figure 2.3: A plenoptic disc corresponding to a point P is entirely determined by the parameters w and ρ. A straight line passing through the optical centre of the focus lens and the image point Q is shown, and where this line intersects the pupilar plane is the plenoptic disc centre w. A lenslet on the boundary of W is labelled ℓ. The pixel p in the subimage of ℓ that images Q appears on the boundary of the subimage of ℓ. The radius ρ can be calculated from the aperture A using a similar-triangles argument. The figure also shows the subimage radius r.

2.3 Point Projection Model

In this section, we use an idealised model of a plenoptic camera that has a lenslet positioned at every point of the lenslet plane and a pixel at every point of the pixel plane. This model is used to derive a feature type that is unique to plenoptic cameras, namely plenoptic disc features. These features are similar in nature to 'circles of confusion' in conventional photography [87], although this term can also refer to a certain type of optical aberration [41]. Furthermore, we show that our plenoptic disc feature parametrisation is 3-dimensional and these features are in one-to-one correspondence with positions of points in the body-fixed frame of the camera. We call the function that maps points to these features the *plenoptic projection*.

In such a model, a point P is projected to a set of lenslet-pixel pairs (ℓ, p), where p is the projection of the image point Q through the lenslet ℓ, as given by (2.5). However, if the subimages of each of the lenslets are circular and equal in radius, the set of lenslet-pixel pairs contains no more information than the set W of lenslets ℓ for which the point P is visible to lenslet ℓ. We call the set W the *plenoptic disc* of a point P.

If the subimages of each of the lenslets are circular, the plenoptic disc will be circular, and thus can be represented entirely by an ideal centre with physical coordinates $w = (w^x, w^y, -D)$ and a signed radius ρ. This is because the set of lenslets ℓ for which a virtual point Q projects into the subimage of ℓ forms a disc in the pupilar plane, see Fig. 2.3.

The relationship between an image point Q and the plenoptic disc data is as follows. Let A be the physical aperture radius of the circular focus lens of the camera. A lenslet physical coordinate ℓ is on the boundary of the set W if there is a line passing from ℓ through Q such that the intercept of this line with the focus lens has length A, see Fig. 2.3. By similar triangles, we find that

$$\rho = \frac{D + Q^z}{-Q^z} A. \tag{2.6}$$

The physical centre of W is given by

$$w = -\frac{D}{Q^z} Q. \tag{2.7}$$

Since the physical radius ρ is related to a radius R in the raw image by the relation $\rho = S\frac{D}{D+d}R$, then using $A = \frac{D}{d}Sr$, where r is the lenslet subimage radius expressed in pixels, together with (2.6) and (2.1), we find that

$$\frac{1}{P^z} = \frac{D - F}{DF} - \frac{d}{r(D+d)D} R, \tag{2.8}$$

where P is the real point corresponding to the image point Q. The parameters $\frac{D-F}{DF}$ and $-\frac{d}{r(D+d)D}$, are equal to $-\frac{K_1}{K_2}$ and $-\frac{1}{rK_2}$, respectively, using the notation of [9].

Because a physical camera may have different scales S^u and S^v for the pixels this will result in two different parameters $f^u = \frac{D+d}{S^u}$ and $f^v = \frac{D+d}{S^v}$. We obtain the relation of a point P and the plenoptic disc data (w^u, w^v, R) as

$$P = -\frac{rK_2}{rK_1 + R} \left(\frac{w^u - c^u}{f^u}, \frac{w^v - c^v}{f^v}, 1 \right). \tag{2.9}$$

The parameters f^u, f^v, c^u, c^v, K_1 and K_2 are the intrinsics we estimate for from plenoptic disc data. They are sufficient to provide point estimates using (2.9). This relation is bijective, and determining the plenoptic disc data (w^u, w^v, R) determines entirely the point P corresponding to it, and vice-versa, if the extrinsics and intrinsics of the camera are known. The projection of a point P to the triple (w^u, w^v, R) is called the *plenoptic projection*, denoted Π, and is given by

$$\Pi(P) = \left(-f^u \frac{P^x}{P^z} + c^u, -f^v \frac{P^y}{P^z} + c^v, -\frac{rK_2}{P^z} - rK_1 \right). \tag{2.10}$$

In summary, we model a plenoptic camera in terms of a projection that sends a point P to a triple (w^u, w^v, R), called the *plenoptic disc data*, where (w^u, w^v) are lenslet coordinates, called the *plenoptic disc centre*, and R is a signed radius called the

plenoptic disc radius.

Because the triple (w^u, w^v, R) can be determined purely from raw light-field data, we can use this feature data, together with knowledge of the true positions of the feature points they correspond to, to estimate the intrinsics and extrinsics of the camera using (2.10).

2.3.1 Plenoptic Point Projection Matrix

Note that the plenoptic point projection may be equivalently expressed by a matrix equation. Define the matrix H as:

$$H := \begin{pmatrix} -f^u & 0 & c^u & 0 \\ 0 & -f^v & c^v & 0 \\ 0 & 0 & -rK_1 & -rK_2 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.11}$$

Then, Equation (2.10) is equivalently stated as:

$$\begin{pmatrix} w^u \\ w^v \\ R \\ 1 \end{pmatrix} = \frac{1}{P^z} H \begin{pmatrix} P^x \\ P^y \\ P^z \\ 1 \end{pmatrix}. \tag{2.12}$$

Note that H is invertible:

$$H^{-1} := \begin{pmatrix} -\frac{1}{f^u} & 0 & 0 & \frac{c^u}{f^u} \\ 0 & -\frac{1}{f^v} & 0 & \frac{c^v}{f^v} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{K_2 r} & -\frac{K_1}{K_2} \end{pmatrix}. \tag{2.13}$$

2.3.2 Distortion Model

We model the effect of plenoptic disc distortion with a first-order approximation

$$(w^u, w^v) = (1 + k_1\delta^2)((w_d^u, w_d^v) - (c^u, c^v)) + (c^u, c^v) \tag{2.14}$$

where (w^u, w^v) are the undistorted plenoptic disc coordinates and (w_d^u, w_d^v) are the distorted coordinates and δ is the distance in pixels from the distorted plenoptic disc centre to the optical centre $\delta = \left|\left|(w_d^u, w_d^v) - (c^u, c^v)\right|\right|$. This one-parameter lens distortion model corrects the majority of the observed distortion in the raw Lytro images, however higher-order radial distortion models can also be used.

2.4 Ray Projection Model

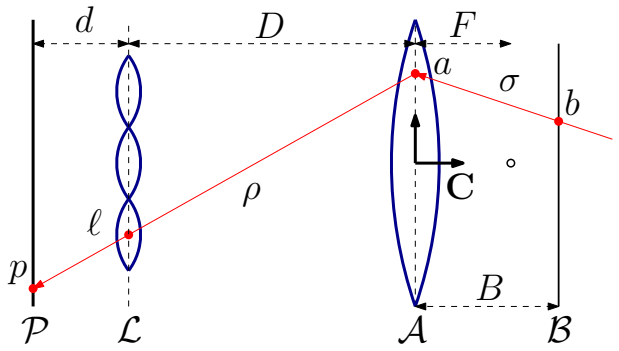


Figure 2.4: A cross-section of a plenoptic camera. A ray of light σ passes through the plane of constant distance B at point b and enters the aperture of the camera at point a. The ray σ is refracted by the focus lens to a ray ρ. The refracted ray passes through the lenslet ℓ and is imaged by pixel p.

There are two ways of formulating the projection model for plenoptic cameras. One way, is to note that every point in front of the camera is in one-to-one correspondence with a set of lenslet-pixel pairs with a certain geometry, known as a plenoptic disc. Another way is to model the camera as mapping rays of light to individual lenslet-pixel pairs. The choice of projection model depends on the application, and in this formulation, we will use the latter model.

We first assign coordinates to rays σ expressed in the body-fixed frame of the camera, see Fig 2.4. We model rays of light as lines in $\mathbb{R}^3$. If a ray passes through the aperture plane $\mathcal{A}$ of the camera, then we assign coordinates to this ray σ given by the intersect $a = (a^x, a^y, a^z)$ of the ray with the aperture plane $\mathcal{A}$, and the intersect $b = (b^x, b^y, b^z)$ of the ray with a second plane $\mathcal{B}$ parallel to $\mathcal{A}$ situated at some constant distance B from $\mathcal{A}$, see Fig. 2.4. Because these planes have constant z-components, as $a^z = 0$ and $b^z = B$, the coordinates we assign to the ray σ are given by $(a^{xy}, b^{xy}) := ((a^x, a^y), (b^x, b^y))$.

The effect of the focus lens of the camera is that this ray σ is refracted to another ray ρ that passes through some lenslet with position $\ell = (\ell^x, \ell^y, \ell^z)$ in the MLA and is measured by some pixel with position $p = (p^x, p^y, p^z)$ in the pixel array, see Fig 2.4. Similary as before, because the lenslets and pixels have constant depths $\ell^z = -D$, and $p^z = -D - d$, we represent the coordinates of the refracted ray ρ by $(\ell^{xy}, p^{xy}) := ((\ell^x, \ell^y), (p^x, p^y))$. The relation of the pair (a^{xy}, b^{xy}) with the lenslet-pixel pair (ℓ^{xy}, p^{xy}) is entirely determined by the intrinsics of the camera.

The aperture-intersect a^{xy} of the ray ρ is equal to the aperture-intersect a^{xy} of the ray σ. This intersect is found by finding where the line passing through ℓ and p

intersects $\mathcal{A}$, and has solution

$$a^{xy} = \ell^{xy} + \frac{D}{d}(\ell^{xy} - p^{xy}) \tag{2.15}$$

by a similar-triangles argument. To express this in terms of lenslet and pixel image-coordinates, we substitute some earlier identities. Considering only a^x for now, we have:

$$\begin{aligned} a^x &= \ell^x + \frac{D}{d}(\ell^x - p^x) \\ &= \frac{D+d}{d}\ell^x - \frac{D}{d}p^x \\ &= \frac{(D+d)D}{d}\left(\frac{\ell^x}{D} - \frac{p^x}{D+d}\right) \\ &= \frac{K_2}{f^u}\left(\frac{f^u\ell^x}{D} + c^u - \frac{f^u p^x}{D+d} - c^u\right) \\ &= \frac{K_2}{f^u}(\ell^u - p^u). \end{aligned}$$

As the derivation for a^y is similar, we have that

$$a^{xy} = K_2\left(\frac{\ell^u - p^u}{f^u}, \frac{\ell^v - p^v}{f^v}\right). \tag{2.16}$$

To determine the coordinate b of the ray σ, we first need to choose the distance B that separates the plane $\mathcal{B}$ from $\mathcal{A}$. There are two choices of B in particular that are important, $B = \frac{FD}{F-D}$, which results in the simplest conversion formula from image coordinates to two-plane coordinates, and $B = 1$, which can simplify theoretical analysis. The factor $\frac{FD}{F-D}$ is an important factor that repeatedly emerges in work on plenoptic cameras. Its physical meaning is the depth of the image of the lenslet plane through the thin-lens. In either case, we need to find one point other than a that the ray σ also passes through, as this will determine the ray entirely. The image point of any point on the ray ρ will be a point on σ. In the case that $B = \frac{FD}{F-D}$, we may simply calculate the image point of the lenslet ℓ^{xy}, because this point lies on the ray σ at distance $\frac{FD}{F-D}$. Therefore, when $B = \frac{FD}{F-D}$, we have that:

$$b^{xy} = \frac{F}{F-D}\ell^{xy}. \tag{2.17}$$

In terms of image coordinates, we may use Equation 2.9, noting that the plenoptic disc radius corresponding to image points on the lenslet plane is equal to 0. There-

fore, when $B = \frac{FD}{F-D}$, we have that:

$$b^{xy} = -\frac{K_2}{K_1}\left(\frac{\ell^u - c^u}{f^u}, \frac{\ell^v - c^v}{f^v}\right). \tag{2.18}$$

Let us denote the b^{xy} given by Equation (2.18) as b_0^{xy}. For an arbitrary B, we find the intersect of the ray that passes through both a^{xy} and b_0^{xy} with the plane $\mathcal{B}$ at distance B from the aperture plane $\mathcal{A}$. The line defined by this ray, when parametrised by depth, is given by

$$\sigma(z) := a^{xy} + z\frac{K_1}{K_2}(b_0^{xy} - a^{xy}),$$

and the intersect at B is simply

$$b^{xy} = a^{xy} + B\frac{K_1}{K_2}(b_0^{xy} - a^{xy}).$$

Therefore, considering for the sake of simplicity only b^x for now, by substituting in Equations (2.16) and (2.18) we have:

$$\begin{aligned} b^x &= K_2\left(\frac{\ell^u - p^u}{f^u}\right) + B\frac{K_1}{K_2}\left(-\frac{K_2}{K_1}\frac{\ell^u - c^u}{f^u} - K_2\frac{\ell^u - p^u}{f^u}\right) \\ &= (K_2 - BK_1)\left(\frac{\ell^u - p^u}{f^u}\right) - B\frac{\ell^u - c^u}{f^u} \\ &= \frac{K_2 - BK_1 - B}{f^u}\ell^u - \frac{K_2 - BK_1}{f^u}p^u + \frac{B}{f^u}c^u. \end{aligned}$$

A similar equation holds for b^y.

These relations are expressed as matrix equations through the use of homogenous coordinates. In doing so, we obtain an essential matrix similar to what was derived in Dansereau *et al.*[22], that is related to the intrinsic parameters of Bok *et al.*[9] by:

$$\begin{pmatrix} a^x \\ a^y \\ b^x \\ b^y \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{K_2}{f^u} & 0 & -\frac{K_2}{f^u} & 0 & 0 \\ 0 & \frac{K_2}{f^v} & 0 & -\frac{K_2}{f^v} & 0 \\ \frac{K_2 - BK_1 - B}{f^u} & 0 & -\frac{K_2 - BK_1}{f^u} & 0 & \frac{B}{f^u}c^u \\ 0 & \frac{K_2 - BK_1 - B}{f^v} & 0 & -\frac{K_2 - BK_1}{f^v} & \frac{B}{f^v}c^v \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \ell^u \\ \ell^v \\ p^u \\ p^v \\ 1 \end{pmatrix}. \tag{2.19}$$

2.5 Disparity

The radius of a plenoptic disc is entirely determined by a property called disparity. Disparity is a quantity that is assigned to a lenslet-pixel pair that is in one-to-one correspondence with depth. This quanitity determines the set of lenslet-pixel pairs

that are imaging the same point on a scene.

In the previous section, we introduced the *image coordinates* of lenslets and pixels. However, some derivations are made easier with a different choice of parametrisation. In this section, instead of parametrising a lenslet in terms of its apparent centre pixel location in the raw image, which is a parametrisation in units of pixels, we parametrise it in units of lenslets. As unused symbols are in short supply, we use bold-face symbols for these coordinates, and because it will be necessary to differentiate these coordinates, for the sake of readability superscripts will not be used either, unlike with the previously introduced lenslet-pixel coordinates. The *lenslet coordinates* $(\mathbf{s}, \mathbf{t})$ of a lenslet in this formulation are given by $(\mathbf{s}, \mathbf{t}) = \frac{1}{2r}(\ell^u - c^u, \ell^v - c^v)$, where r is the subimage radius. The coordinates $(\mathbf{s}, \mathbf{t})$ are in units of lenslets. Furthermore, the coordinates of a pixel are expressed relatively to the lenslet whose image that pixel is a part of. The *pixel offset coordinates* $(\mathbf{u}, \mathbf{v})$ are given by $(\mathbf{u}, \mathbf{v}) = (p^u, p^v) - (\ell^u, \ell^v)$. The $(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v})$ coordinates are called a lenslet-pixel pair and define a ray in space: the ray that passes through the lenslet with coordinates $(\mathbf{s}, \mathbf{t})$, and through the pixel with coordinates $(\mathbf{u}, \mathbf{v})$ in the pixel plane of the lenslet with coordinates $(\mathbf{s}, \mathbf{t})$.

Now, suppose that there is some surface X in front of the camera, and that for every point $P \in X$ we assign some quantity $F(P)$. Suppose furthermore that we define a function f on the two-plane coordinates by setting $f(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v})$ equal to $F(P)$, where P is the first point on the scene that the ray parametrised by $(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v})$ passes through. It is clear that for every point $P \in X$, there is a ray passing through it from every direction. Therefore, there is a subset of two-plane coordinates that necessarily have the same f value.

This subset is entirely determined by three parameters, two for the apparent position of the point P in the central-aperture image, and one for a quantity known as disparity, which we denote here as δ. Disparity δ is defined here as the quantity assigned to a lenslet-pixel pair $(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v})$ such that if the ray given by $(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v})$ passes through a point P, then so do all the rays with coordinates $(\mathbf{s} + \delta\Delta\mathbf{u}, \mathbf{t} + \delta\Delta\mathbf{v}, \mathbf{u} + \Delta\mathbf{u}, \mathbf{v} + \Delta\mathbf{v})$ for all pixel displacements $(\Delta\mathbf{u}, \Delta\mathbf{v})$. Because disparity is a quantity that converts a pixel displacement to a lenslet displacement, its units are in lenslets per pixel (lens/pix).

Since all of the aforementioned rays pass through the same point on the scene, we therefore have the relation

$$f(\mathbf{s} + \delta\Delta\mathbf{u}, \mathbf{t} + \delta\Delta\mathbf{v}, \mathbf{u} + \Delta\mathbf{u}, \mathbf{v} + \Delta\mathbf{v}) = f(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v}), \tag{2.20}$$

for any $\Delta\mathbf{u}$ and $\Delta\mathbf{v}$ and $\delta = \delta(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v})$. In particular, this defines a level set, and the gradient of f is orthogonal to this level set. By expressing $(\Delta\mathbf{u}, \Delta\mathbf{v}) = K\omega$ for some

$\omega \in S^2$, $K > 0$ we therefore obtain that

$$\delta\omega \cdot \nabla_{\mathbf{s},\mathbf{t}} f + \omega \cdot \nabla_{\mathbf{u},\mathbf{v}} f = 0 \tag{2.21}$$

which, for any $\omega \in S^2$, has the solution for δ as

$$\delta = -\frac{\omega \cdot \nabla_{\mathbf{u},\mathbf{v}} f}{\omega \cdot \nabla_{\mathbf{s},\mathbf{t}} f}. \tag{2.22}$$

When the original scene function F is given by colours of points, the resulting function f we denote as L and is a Lambertian light-field. The purpose of our formulation is to emphasise the definition of disparity as a property of the level sets of functions defined on the two-plane parametrisation, formed by propogating another function through space along straight lines.

2.5.1 Disparity Field Estimation

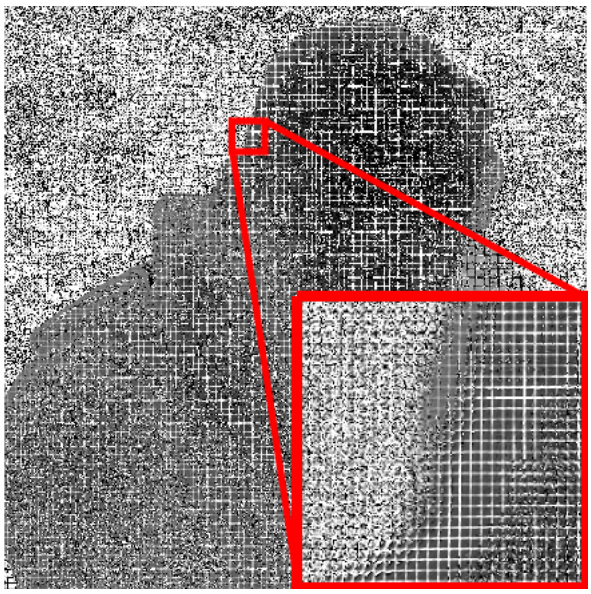


Figure 2.5: Disparity map of raw image given by convolutions. In zoomed box, edges of lenslet microimages can be seen where gradients become unreliable.

To estimate the gradients of the light-field $\nabla L = (\nabla_{\mathbf{s}} L, \nabla_{\mathbf{t}} L, \nabla_{\mathbf{u}} L, \nabla_{\mathbf{v}} L)$ along each of the cardinal directions $\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v}$, we use 4-dimensional sobel operators. These operators are constructed in the following way. To estimate the gradient of L along the $\mathbf{s}$ axis, we first construct 4 4D arrays $h'_{\mathbf{s}}, h_{\mathbf{t}}, h_{\mathbf{u}}, h_{\mathbf{v}}$ of dimensions $(3, 3, r, r)$, where r is the diameter of a lenslet. The array $h'_{\mathbf{s}}$ is 0 everywhere except along the central $\mathbf{s}$ axis of the array, where it contains the coefficients $(-1, 0, 1)$ of a finite-difference

approximation. Similarly, along each of the central $\mathbf{t}$, $\mathbf{u}$, and $\mathbf{v}$ axes of the arrays $h_{\mathbf{t}}$, $h_{\mathbf{u}}$, and $h_{\mathbf{v}}$ respectively, are the values of a triangle filter $(1, 2, 1)$. The 4D sobel filter kernel $H_{\mathbf{s}}$ in the $\mathbf{s}$ direction is given by $H_{\mathbf{s}} := h'_{\mathbf{s}} * h_{\mathbf{t}} * h_{\mathbf{u}} * h_{\mathbf{v}}$. Equivalent processes are done for the other Sobel filter kernels $H_{\mathbf{t}}$, $H_{\mathbf{u}}$, and $H_{\mathbf{v}}$. Finally, reshaping the resulting arrays into 2D matrices gives image kernels that estimate these gradients on raw light-field image data.

The resulting convolution kernels will be used on convolutions on the raw images to compute light-field gradients in each of the cardinal directions of the light-field, and so the gradient ∇L of the lightfield can be approximated at every lenslet-pixel pair. The accuracy of a disparity field using (2.34) depends on the magnitude of the demoninator of the quotient. Hence, we take $\omega = \nabla_{\mathbf{s},\mathbf{t}} L$. An initial disparity estimate can then be computed by

$$\delta_0 = -\frac{(H_{\mathbf{s}} * L, H_{\mathbf{t}} * L) \cdot (H_{\mathbf{u}} * L, H_{\mathbf{v}} * L)}{||(H_{\mathbf{s}} * L, H_{\mathbf{t}} * L)||^2} \tag{2.23}$$

Here, the convolutions are taken over all three colour channels, so that, for example

$$H_{\mathbf{s}} * L = \left(H_{\mathbf{s}} * L^R, H_{\mathbf{s}} * L^G, H_{\mathbf{s}} * L^B\right),$$

where L^R, L^G, and L^B are the red, green and blue colour channels, respectively. It should be noted that this technique is essentially the same method as described by Adelson and Wang [**?**]. However, this initial estimate can be improved in regions of low texture by using the result of

$$\delta_0 = -\frac{(H_{\mathbf{s}} * \tilde{L}_\sigma, H_{\mathbf{t}} * \tilde{L}_\sigma) \cdot (H_{\mathbf{u}} * \tilde{L}_\sigma, H_{\mathbf{v}} * \tilde{L}_\sigma)}{\left|\left|(H_{\mathbf{s}} * \tilde{L}_\sigma, H_{\mathbf{t}} * \tilde{L}_\sigma)\right|\right|^2}, \tag{2.24}$$

where $\tilde{L}_\sigma$ is the original lightfield blurred by some gaussian kernel with standard deviation σ. The result of (2.23) is replaced by the result of (2.24) in the regions where $||(H_{\mathbf{s}} * L, H_{\mathbf{t}} * L)||$ is beneath some threshold τ, and

$$\left|\left|(H_{\mathbf{s}} * \tilde{L}_\sigma, H_{\mathbf{t}} * \tilde{L}_\sigma)\right|\right| < ||(H_{\mathbf{s}} * L, H_{\mathbf{t}} * L)||.$$

The purpose of the latter condition is to avoid replacing disparities near occlusion boundaries, because here the gaussian blurred images have high norm in the lenslet directions.

2.5.2 Disparity Accumulation

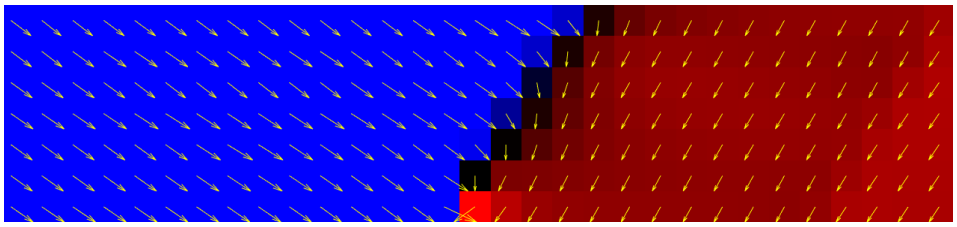


Figure 2.6: Epipolar image of a disparity field near an occlusion boundary. Near disparities are shown in red, whereas far disparities are shown in blue. Disparities are propogated to the centre sub-aperture image using flow lines of Burgers' equation. Shown in the yellow arrows is the vector field $(\delta, 1)$.

While the initial depth estimates are fast, they are often noisy. A method of supressing this noise and generating an estimate of confidence is derived here.

Using the method in the previous section, we obtain a disparity map $\delta(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v})$ that assigns a disparity to every lenslet-pixel pair. Remember that true disparity is independent of the texture of the scene, and so Eq. (2.34) holds for any texture of the scene that satisfies the Lambertian property. This means that for a fixed scene and fixed camera the true disparity δ is constant even if the colouring applied to the scene varies. Therefore, if we set each point on the scene to have a colour equal to the disparity associated with that point, we obtain the equation

$$\delta(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v}) = -\frac{\omega \cdot \nabla_{\mathbf{uv}} \delta(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v})}{\omega \cdot \nabla_{\mathbf{st}} \delta(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v})}, \tag{2.25}$$

which holds for any $\omega \in S^2$. These equations are equivalent to

$$\delta \cdot D_{\mathbf{s}} \delta + D_{\mathbf{u}} \delta = 0 \tag{2.26}$$

$$\delta \cdot D_{\mathbf{t}} \delta + D_{\mathbf{v}} \delta = 0. \tag{2.27}$$

These equations are geometric constraints that necessarily hold for any disparity map δ. Equations (2.26) and (2.27) are well-known in the partial differential equation literature as inviscid Burgers' equations. These equations are known to model shock-waves and in this setting, the discontinuities of this map occur precisely along the apparent location of occlusion boundaries in the light-field. As a consequence of these equations, we also obtain an analogue of Eq. (2.20)

$$\delta(\mathbf{s} + \delta\Delta\mathbf{u}, \mathbf{t} + \delta\Delta\mathbf{v}, \mathbf{u} + \Delta\mathbf{u}, \mathbf{v} + \Delta\mathbf{v}) = \delta(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v}). \tag{2.28}$$

This equation is used to accumulate disparity estimates from the entire raw light-field

image to the disparity estimate for the central sub-aperture view $\delta(\mathbf{s}, \mathbf{t}, 0, 0)$. This is done in the following way. For each lenslet-pixel pair $(\mathbf{s}_0, \mathbf{t}_0, \mathbf{u}_0, \mathbf{v}_0)$ in the raw light-field image, calculate the intersect of the line that passes through $(\mathbf{s}_0, \mathbf{t}_0, \mathbf{u}_0, \mathbf{v}_0)$ with slope $\delta(\mathbf{s}_0, \mathbf{t}_0, \mathbf{u}_0, \mathbf{v}_0)$ with the plane $\{(\mathbf{s}, \mathbf{t}, 0, 0) : (\mathbf{s}, \mathbf{t}) \in \mathbb{R}^2\}$. The lenslet coordinates of this intersect $(\mathbf{s}_i, \mathbf{t}_i)$ is given by

$$(\mathbf{s}_i, \mathbf{t}_i) = (\mathbf{s}_0 - \delta \mathbf{u}_0, \mathbf{t}_0 - \delta \mathbf{v}_0)\,. \tag{2.29}$$

Thus, for each lenslet-pixel pair $(\mathbf{s}_0, \mathbf{t}_0, \mathbf{u}_0, \mathbf{v}_0)$, there is an associated lenslet $(\mathbf{s}_i, \mathbf{t}_i)$ whose central pixel should have disparity $\delta(\mathbf{s}_0, \mathbf{t}_0, \mathbf{u}_0, \mathbf{v}_0)$. Of course, from the initial disparity estimate in Section 2.5.1, there is already a disparity assigned to this lenslet-pixel pair, and many other lenslet-pixel pairs will also be assigned to this lenslet that all have different disparities associated to them. Thus, for each central lenslet $(\mathbf{s}_i, \mathbf{t}_i)$ we obtain a set $\{\delta_k\}_{k=1}^{K_{(\mathbf{s}_i, \mathbf{t}_i)}}$ of disparities that have been propogated to this lenslet. The number $K_{(\mathbf{s}_i, \mathbf{t}_i)}$ is the number of disparities in this set and depends on $(\mathbf{s}_i, \mathbf{t}_i)$, and also implicitly on the disparity map δ.

We produce an accumulated estimate of disparity from these sets $\{\delta_k\}_{k=1}^{K_{(\mathbf{s}_i, \mathbf{t}_i)}}$ by taking the median $\mu_{(\mathbf{s}_i, \mathbf{t}_i)}$ of these sets. Furthermore, the standard deviation $\sigma_{(\mathbf{s}_i, \mathbf{t}_i)}$ of each set provides a confidence measure of this accumulated estimate.

We can also construct a cost-volume $V(\mathbf{s}_i, \mathbf{t}_i, \delta)$ from the sets $\{\delta_k\}_{k=1}^{K_{(\mathbf{s}_i, \mathbf{t}_i)}}$ by fitting probability distributions to their histograms and setting V to be equal to the log-likelihood of these probability distributions.

2.5.3 Sufficient Conditions for Depth from Light-Field Gradients

In this section, we show that depth measurements may in principle be perfectly estimated using light-field data. To show this, we require an assumption on the colour distribution of our light-field measurement.

Assumption 1. *The light-field $L : \mathcal{L} \times \mathcal{P} \to [0, 1]^3$ is differentiable and satisfies*

$$||\nabla \mu_{\mathbf{s},\mathbf{t}}(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v})|| > 0$$

for all $(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v}) \in \mathcal{L} \times \mathcal{P}$.

This assumption is fulfilled for Lambertian scenes whose colouring has non-zero gradient everywhere. The existence of such a colouring for any given (smooth) scene surface is guaranteed by a theorem of Hirsch [36], and an example for a scene contained in the unit cube $[0, 1]^3$ is given by assigning to each point P on the scene

the colour P. The following proposition proves correctness of depth estimates from light-field data given this assumption.

Proposition 1. *Let $\phi = (K_1, K_2, f^u, f^v, c^u, c^v)$ be the intrinsic parameters of a light-field camera without lens distortion. Use Assumption 1 and define the function $\delta : \mathcal{L} \to \mathbb{R}^+$ as*

$$\delta(\mathbf{s}, \mathbf{t}) := -\frac{\nabla_{\mathbf{s},\mathbf{t}}\mu(\mathbf{s}, \mathbf{t}, 0, 0) \cdot \nabla_{\mathbf{u},\mathbf{v}}\mu(\mathbf{s}, \mathbf{t}, 0, 0)}{||\nabla_{\mathbf{s},\mathbf{t}}\mu(\mathbf{s}, \mathbf{t}, 0, 0)||^2}. \tag{2.30}$$

Then, the function $\lambda : \mathcal{L} \to \mathbb{R}$ defined as

$$\lambda(\mathbf{s}, \mathbf{t}) := -\frac{K_2}{K_1 + \delta(\mathbf{s}, \mathbf{t})}$$

is equal to the depth of the first point on the scene surface that lies along the ray with coordinates $(\mathbf{s}, \mathbf{t}, 0, 0)$.

Proof. In Section 2.3, it was shown that the depth of a point P with $\pi(P) \in \mathcal{L}$ expressed in body-fixed coordinates $\mathbf{C}$ of the camera is given by

$${}^{\mathbf{C}}P^z = -\frac{K_2}{K_1 + \frac{R(P)}{r}}, \tag{2.31}$$

where $R(P)$ is the plenoptic disc radius of P, and r is the subimage radius. Now, observe that:

$$\begin{pmatrix} \mathbf{u}_1 - \mathbf{u}_2 \\ \mathbf{v}_1 - \mathbf{v}_2 \end{pmatrix} = \frac{r}{R(P)} \begin{pmatrix} \mathbf{s}_1 - \mathbf{s}_2 \\ \mathbf{t}_1 - \mathbf{t}_2 \end{pmatrix}$$

holds for all pairs of lenslet-pixel coordinates $(\mathbf{s}_1, \mathbf{t}_1, \mathbf{u}_1, \mathbf{v}_1)$ and $(\mathbf{s}_2, \mathbf{t}_2, \mathbf{u}_1, \mathbf{v}_2)$ imaging the same point P. Comparing this to the defining equation

$$\begin{pmatrix} \Delta\mathbf{s} \\ \Delta\mathbf{t} \end{pmatrix} = \delta(P) \begin{pmatrix} \Delta\mathbf{u} \\ \Delta\mathbf{v} \end{pmatrix}$$

for the disparity $\delta(P)$ of P, it follows that

$$\delta(P) = \frac{R(P)}{r}$$

and hence Equation (2.31) can be rewritten as

$${}^{\mathbf{C}}P^z = -\frac{K_2}{K_1 + \delta(P)}. \tag{2.32}$$

The light-field measurement $\mu : \mathcal{L} \times \mathcal{P} \to [0,1]^3$ is constant on the level set

$$\{(\mathbf{s} + \delta(Q)\Delta\mathbf{u}, \mathbf{t} + \delta(Q)\Delta\mathbf{v}, \Delta\mathbf{u}, \Delta\mathbf{v}) : (\Delta\mathbf{u}, \Delta\mathbf{v}) \in \mathbb{R}^2\},$$

where Q is the first point on the scene surface ∂X that lies along the ray with coordinates $(\mathbf{s}, \mathbf{t}, 0, 0)$. The gradient of μ is nonzero by Assumption 1 and orthogonal to this level set at $(\mathbf{s}, \mathbf{t}, 0, 0)$. By expressing $(\Delta\mathbf{u}, \Delta\mathbf{v}) = \rho\omega$ for some $\omega \in S^2$, $\rho > 0$ we therefore obtain that

$$\delta(Q)\omega \cdot \nabla_{\mathbf{st}}\mu + \omega \cdot \nabla_{\mathbf{u,v}}\mu = 0 \tag{2.33}$$

which, for any $\omega \in S^2$, has the solution

$$\delta(Q) = -\frac{\omega \cdot \nabla_{\mathbf{uv}}\mu}{\omega \cdot \nabla_{\mathbf{st}}\mu}. \tag{2.34}$$

Letting $\omega = \nabla_{\mathbf{s,t}}\mu(\mathbf{s}, \mathbf{t}, 0, 0)$, we obtain (2.30), and the result follows by substituting into Equation (2.32). □

2.5.4 Necessary Conditions for Depth from Light-Field Gradients

In the previous section, a proof that depth may be estimated from light-field gradients for a lenslet-based light-field camera if the scene is Lambertian and textured was given. In this section, we prove under some additional assumptions that it is actually *neccessary* that the scene is textured and Lambertian. For the sake of simplicity, this proof only considers the two-dimensional setting, and assumes that the colouring of the scene is monochromatic. We begin by first defining some of the objects and properties that will be used in this proof.

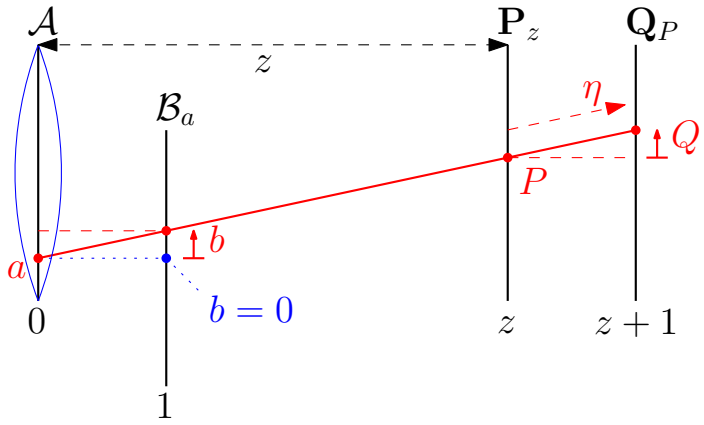


Figure 2.7: A two-plane camera together with a planar scene $\mathbf{P}_z$ at depth z. The direction component of the colouring has been reparametrised to match the geometry of the camera.

Definitions

We define the camera in terms of a two-plane parametrisation. Such a construction was shown to be possible in Section 2.4. To do this, we first define the set $A := [a_{\min}, a_{\max}] \subset \mathbb{R}$ and, when A is given, the *pupilar plane* $\mathcal{A} := \{(a, 0) \in \mathbb{R}^2 : a \in A\}$. Then, we define the set $B := [b_{\min}, b_{\max}] \subset \mathbb{R}$ and, when $a \in A$ and B are given, the *retinal plane* $\mathcal{B}_a := \{(b + a, 1) \in \mathbb{R}^2 : b \in B\}$. The parameter a determines the point $(a, 0)$ on the aperture that a ray is seen, and the parameter $b \in B$ determines the direction of that ray. Define the map $\eta : B \to S^1$ as $\eta(b) := \frac{(b,1)}{||(b,1)||}$. Then the physical ray corresponding to coordinates $(a, b) \in A \times B$ is $((a, 0), \eta(b))$. This defines the simplified camera geometry that will be used in this section, and is illustrated in Fig. 2.7.

We will assume that the scene class $\mathbb{X}$ used in this proof consists of scenes (X, β) where $X \subset \mathbb{R}^2$ and $\beta : X \times S^1 \to \mathbb{R}$. This corresponds to two-dimensional scenes with monochromatic colourings. As defined in Chapter 1, $\mathbf{X}$ is the set of scene geometries. However, in this section we will also make use of the set of possible colourings of a given scene. This is given by defining for each scene $X \in \mathbf{X}$, the cross-section

$$B_X := \{\beta : X \times S^1 \to \mathbb{R} \mid (X, \beta) \in \mathbb{X}\}$$

consisting of the set of colourings β such that $(X, \beta) \in \mathbb{X}$. Throughout this section, we will make use of planar scenes, defined as follows.

Definition 1. *A* planar scene *is given by*

$$\mathbf{P}_{z_0} := \{(x, z) \in \mathbb{R}^2 \mid z = z_0\}$$

for some z_0.

Note that we may also parametrise the colourings of planar scenes in terms of the two-plane parametrisation. Such a reparametrised colouring will be denoted $\overline{\beta}$.

Definition 2. *Define for each* $(\mathbf{P}_z, \beta) \in \mathbb{X}$ *the map* $\overline{\beta} : \mathbb{R}^2 \to \mathbb{R}$ *as* $\overline{\beta}(p, q) := \beta((p, z), \eta(q))$.

Because we will use the assumption that we may view a scene from potentially any angle, it may be the case that the measured light-field generated by a scene (X, β) has as its domain only a subset of $A \times B$. This is because we do not assume that the camera is contained within the environment of the scene. Furthermore, because in Theorem 2 we implicitly assume that $\nabla L(a, b)$ exists, we will also need to restrict the lenslet-pixel pairs to those for which this is true.

Definition 3. *For each* $(X, \beta) \in \mathbb{X}$, *the set* $\mathcal{S}_X$ *is the subset of lenslet-pixel pairs* $(a, b) \in A \times B$ *such that:*

1. $((a,0),\eta(b) \in \Sigma_X$ *and* $(a,0) \notin X$.
2. $\nabla L(a,b)$ *exists.*
3. $\nabla\gamma(a,b)$ *exists.*

The set $\mathcal{S}_X$ is the subset of coordinates $(a,b) \in A \times B$ for which the ray corresponding to that coordinate is in the environment of the scene X, and the light-field and depth map is differentiable there. In particular, this excludes lenset-pixel pairs pertaining to rays that intersect occlusion boundaries. Using this domain, we may safely define for each $(X,\beta) \in \mathbb{X}$, the *depth-map* $\gamma_X : \mathcal{S}_X \to \mathbb{R}$ that returns the depth of the nearest point on X along the ray $((a,0),\eta(b))$. Additionally, we define for each $(X,\beta) \in \mathbb{X}$, the *light-field* $L_{(X,\beta)} : \mathcal{S}_X \to \mathbb{R}$ as

$$L(a,b) := \beta\left(a + \lambda_X\left(a,\eta(b)\right)\eta(b),\eta(b)\right),$$

where $\lambda_X(a,\eta(b))$ is the range of the scene along ray $((a,0),\eta(b))$ (see also Section 1.3.1). Note that for a given scene the domain of the depth-map and light-field of that scene are equal. The subscripts in these definitions will be dropped when clear from context.

For differentiable colourings, a surface ray being Lambertian is equivalent to the angular derivative of the light-field at that ray being 0. Thus, in this section we call a coloured scene (X,β) *Lambertian* if every surface ray (P,η) satisfies $D_2\beta(P,\eta) = 0$. It is this property, along with the following, that will be shown to be neccessary conditions of extracting depth from light-fields. A coloured scene (X,β) is said to be *spatially textured* if every surface ray (P,η) satisfies $D_1\beta(P,\eta) \neq 0$. Additionally, a coloured scene (X,β) is said to be *textured* if every surface ray (P,η) satisfies $D\beta(P,\eta) \neq 0$.

This section uses the assumption that for certain motions, a scene may be moved rigidly along that motion. The way in which a rigid-body motion acts on a scene is defined as follows.

Definition 4. *Define for each* $\xi \in SE(2)$ *and each* $(X,\beta) \in \mathbb{X}$ *the set*

$$\xi \cdot X := \{P \in \mathbb{R}^2 : P = \xi \cdot P',\ f.\ s.\ P' \in X\},$$

and the function

$$\xi \cdot \beta(P,\eta) = \beta(\xi^{-1} \cdot (P,\eta)).$$

Finally, we define the notion of visibility and viewability of a surface ray. A surface ray is visible if it currently being imaged, or equivalently that it lies in the visible set of the camera (see Section 1.3.3).

Definition 5. *A surface ray* (P, η_0) *of a scene* X *is said to be* visible *if there exists an* $(a, b) \in \mathcal{S}_X$ *such that*

$$P = (a + b\gamma_X(a, b), \gamma_X(a, b))$$

and

$$\eta_0 = \eta(b).$$

A surface ray is viewable if the scene may be moved in some way that makes it visible.

Definition 6. *A surface ray* (P, η_0) *of a scene* X *is said to be* viewable *if there exists a rigid body motion* ξ *such that*

$$\xi \cdot P = (a + b\gamma_{\xi \cdot X}(a, b), \gamma_{\xi \cdot X}(a, b))$$

and

$$\xi \cdot \eta_0 = \eta(b).$$

Note that the varying domain of $\gamma_{\xi \cdot X}$ *implicity restricts which surface rays are viewable.*

Finally, as we can only state anything about those rays that are viewable, we provide the following definition of "effectively Lambertian" and "effectively textured".

Definition 7. *A coloured scene is* effectively Lambertian *if every viewable ray is Lambertian, and* effectively textured *if every viewable ray is textured.*

Assumptions

To prove the necessary conditions on colourings required for depth from light-field gradients, several assumptions will be made. Some of these assumptions are possibly consequences of the other assumptions. The first is that the scene class contains the class of planar scenes. This assumption may potentially be either weakened or removed entirely.

Assumption 1. *Assume the scene class* $\mathbb{X}$ *satisfies: for each* $z > 0$ *the planar scene* $\mathbf{P}_z$ *at depth* z *is in* $\mathbf{X}$*.*

The following assumption states that the scene class is invariant to changes in contrast of the colourings. This assumption is reasonable because otherwise we would require a choice of precise contrast parameters on the camera recording the light-fields and merely scaling the colour values would break the method.

Assumption 2. *Assume the scene class* $\mathbb{X}$ *satisfies: for all* $(X, \beta) \in \mathbb{X}$*, we have* $(X, k \cdot \beta) \in \mathbb{X}$ *for each* $k \neq 0$*.*

The third assumption is that the colour assigned to a surface ray at some point on the scene does not depend on the colour assigned to a surface ray at another point on the scene.

Assumption 3. *Assume the scene class $\mathbb{X}$ satisfies: for every planar scene $\mathbf{P}_z \in \mathbf{X}$ where $z > 0$, and all $(P, \eta, c), (P', \eta', c') \in \mathbf{P}_z \times S^1 \times \mathbb{R}$ such that $P \neq P'$, there exists a colouring $\beta \in B_{\mathbf{P}_z}$ such that $\beta(P, \eta) = c$ and $\beta(P, \eta') = c'$.*

The fourth assumption is that each scene in the scene class may be moved freely and that the result is still in the scene class.

Assumption 4. *Assume the scene class $\mathbb{X}$ satisfies: for all $\xi \in SE(2)$ if $(X, \beta) \in \mathbb{X}$ then $(\xi \cdot X, \xi \cdot \beta) \in \mathbb{X}$.*

The fifth assumption is very likely to be a consequence of the previous assumptions. It states that for every planar scene in the scene class and every surface ray of the scene, there is a colouring on that planar scene such that the derivative of the colouring at that ray is non-zero.

Assumption 5. *Assume the scene class $\mathbb{X}$ satisfies: for all $z > 0$, $P \in \mathbf{P}_z$, $\eta \in S_1$, there exists a $\beta \in B_{\mathbf{P}_z}$ such that $D\beta(P, \eta) \neq 0$.*

Therorem and Proof

The following theorem states that the existence of some method that takes light-field gradients and returns depth implies that the scene being imaged is textured and Lambertian under the previous assumptions. Note that this theorem only applies to the estimation of depth directly from light-field gradients. However, there may be other maps that estimate depth under more relaxed conditions if higher-order differential information is used, or if global properties of the light-field are taken into account.

Theorem 2. *Let $\mathbb{X}$ be a scene class satisfying Assumptions 1 - 5. Suppose that there exists a function $F : \mathbb{R}^2 \to \mathbb{R}$ such that for every $(X, \beta) \in \mathbb{X}$, whenever $(a, b) \in \mathcal{S}_X$ we have that*

$$F(D_a L(a, b), D_b L(a, b)) = \gamma(a, b),$$

where $\gamma(a, b)$ is the depth of the scene along the ray $((a, 0), \eta(b))$. Then every scene $(X, \beta) \in \mathbb{X}$ is effectively textured and effectively Lambertian.

This proof requires the following lemmas.

Lemma 3. *For any $z > 0$ and any $a \in A$, and any $b \in B$, and any $v \in \mathbb{R} \setminus \{0\}$ there exists a $\beta \in B_{\mathbf{P}_z}$ such that $||DL(a, b)|| = v$ for the light-field L generated by β.*

Proof. By Assumption 5 and Eq. (2.35), for given any $z > 0$ and any $a \in A$, and any $b \in B$, we have that there exists a colouring $\beta \in B_{\mathbf{P}_z}$ such that $||DL(a,b)|| = v_0 \neq 0$. By Assumption 2, for all $v \neq 0$, $\frac{v}{v_0} \cdot \beta \in B_{\mathbf{P}_z}$, therefore, by Eqs. (2.36) and (2.37), the light-field L' generated by $\frac{v}{v_0}\beta \in B_{\mathbf{P}_z}$ satisfies $||DL'(a,b)|| = v$. □

The following lemma is an immediate consequence of Assumption 4. It is simply used to define special notation for the translation of a scene.

Lemma 4. *An immediate consequence of Assumption 4 is that the scene class $\mathbb{X}$ satisfies: if $(\mathbf{P}_z, \beta) \in \mathbb{X}$ then for all $x' \in \mathbb{R}$ and $z' > 0$, $(\mathbf{P}_{z'}, \beta \circ \tau_{(x',z')}) \in \mathbb{X}$ where $\tau_{(x',z')}((x,z),\eta) = ((x',z'),\eta)$. (Every translation of a coloured plane in the scene class to a location in front of the camera is in the scene class.)*

There are several steps on the path to proving Theorem 2. The first involves directly relating the light-field to the colouring of a planar scene.

Proposition 5. *For any $z > 0$, any $\mathbf{P}_z \in \mathbf{X}$, any $\beta \in B_{\mathbf{P}_z}$ and any $a \in A$, $b \in B$, the reparametrised colouring $\overline{\beta}$ satisfies equations* (2.35)*,* (2.36)*, and* (2.37)*.*

Proof. To see this, choose any $z > 0$ and consider the scene $\mathbf{P}_z$ and the cross-section $B_{\mathbf{P}_z}$. Any surface ray $(P,\eta) \in \mathbf{P}_z$ that points away from the plane $\mathbf{P}_0$ may be parametrised by $(u,v) \in \mathbb{R}^2$ explicitly through the invertible map $(u,v) \mapsto ((u,z),\eta(v))$. Therefore, for any $\beta \in B_{\mathbf{P}_z}$, we have for the L generated by β and the reparametrised colouring $\overline{\beta}$

$$L(a,b) = \overline{\beta}(a + bz, b). \tag{2.35}$$

Therefore, for any $\beta \in B_{\mathbf{P}_z}$, we have for the L generated by β and the reparametrised colouring $\overline{\beta}$

$$D_a L(a,b) = D_1\overline{\beta}(a + bz, b). \tag{2.36}$$

Therefore, for any $\beta \in B_{\mathbf{P}_z}$, we have for the L generated by β and the reparametrised colouring $\overline{\beta}$

$$D_b L(a,b) = D_1\overline{\beta}(a + bz, b)z + D_2\overline{\beta}(a + bz, b). \tag{2.37}$$

□

The next step of the proof involves establishing that each planar scene in the scene class is textured.

Proposition 6. *All viewable rays of planar scenes in $\mathbb{X}$ are spatially textured, meaning that if $(P_z, \beta) \in \mathbb{X}$, $D_1\overline{\beta}(p,q) \neq 0$ for all $z > 0$, and all $(p,q) \in \mathbb{R}^2$.*

Proof. By Lemma 3 We have that for any fixed choice of z, there exists a colouring β such that $DL(a,b) = (D_1\overline{\beta}(a + bz, b)z + D_2\overline{\beta}(a + bz, b), D_1\overline{\beta}(a + bz, b)) \neq (0,0)$.

Therefore, at least one of the following is therefore true: either $D_1\overline{\beta}(a+bz,b)z + D_2\overline{\beta}(a+bz,b) \neq 0$ or $D_1\overline{\beta}(a+bz,b) \neq 0$. Assume $D_1\overline{\beta}(a+bz,b)z + D_2\overline{\beta}(a+bz,b) \neq 0$ and $D_1\overline{\beta}(a+bz,b)z = 0$. Then $F(0, D_2\overline{\beta}(a+bz,b)) = z$. Then, due to Lemma 3, $k \cdot \beta \in B_{\mathbf{P}_z}$ for all $k \neq 0$, and we have that for all $v \in \mathbb{R} \setminus 0$, there exists a $\beta \in B_{\mathbf{P}_z}$ such that $D_2\overline{\beta}(a+bz,b) = v$, we have that $F(0,v) = z$ for all v. However, this same argument will hold for some other planar scene at any other depth $z' > 0$, which would imply that we would also have $F(0,v) = z'$ for all v by the same argument. Therefore, $D_1\overline{\beta}(a+bz,b) \neq 0$. Therefore β is spatially textured at every visible surface ray of P_z. Therefore, since all rigid body motions of the scene are in the scene class, β is spatially textured for any viewable $(p,q) \in \mathbb{R}^2$. □

The third step in the proof is establishing the Lambertian property for planar scenes.

Proposition 7. *All viewable rays of planar scenes in $\mathbb{X}$ are Lambertian, meaning that if $(P_z, \beta) \in \mathbb{X}$, $D_2\overline{\beta}(p,q) = 0$ for all $z > 0$, and all $(p,q) \in \mathbb{R}^2$*

Proof. Showing that any planar scene must be Lambertian requires additional steps. The aim is to show that there exists a function $g : \mathbb{R} \to \mathbb{R}$ such that for all $a, b, z > 0, \beta \in B_{\mathbf{P}_z}$, $D_2\overline{\beta}(a+bz,b) = g(D_1\overline{\beta}(a+bz))$ and then prove that this function is infact identically 0. However, this itself first requires proving the weaker statement that there exists a $h : \mathbb{R}^2 \to R$ such that $h(D_1\overline{\beta}(a+bz), z) = D_2\overline{\beta}(a+bz)$ for all a, b, z. Suppose for a contradiction that such a function h did not exist. Then, there are two colourings $\beta_1, \beta_2 \in B_{\mathbf{P}_{z_0}}$ such that:

$$\begin{aligned} D_1\overline{\beta}_1(a+bz_0,b) &= D_1\overline{\beta}_2(a+bz_0,b), \\ D_1\overline{\beta}_1(a+bz_0,b)z_0 + D_2\overline{\beta}_1(a+bz_0,b) &\neq D_1\overline{\beta}_2(a+bz_0,b)z_0 + D_2\overline{\beta}_2(a+bz_0,b), \end{aligned}$$

for at least one choice of $z_0 > 0$, $a \in A$, $b \in B$. For shorthand, we will use the notation

$$\begin{aligned} x_0 &= D_1\overline{\beta}_1(a+bz_0,b) = D_1\overline{\beta}_2(a+bz_0,b), \\ y_i &= D_2\overline{\beta}_i(a+bz_0,b), \\ w_i &= x_0z_0 + y_i. \end{aligned} \tag{2.38}$$

By multiplying such chosen $\beta_1, \beta_2 \in B_{\mathbf{P}_{z_0}}$ by a suitable $k \neq 0$, we may assume without loss of generality that $x_0 > 0$. Now, assume without loss of generality that $w_2 > w_1$. Then $y_2 > y_1$ and $w_2 = x_0z_0 + y_2 > y_2 > y_1$ because $x_0z_0 > 0$. Therefore, defining

$z_1 := \frac{w_2 - y_1}{x_0} > 0$, we have

$$\begin{aligned} w_2 &= x_0 z_0 + y_2 \\ &= x_0 z_1 + y_1. \end{aligned} \tag{2.39}$$

Because $z_1 > 0$, we may translate the colouring β_1 to

$$\beta_1' := \beta_1 \circ \tau_{(bz_1, z_1)} \in B_{\mathbf{P}_{z_1}},$$

ie. $\overline{\beta}_1'(a + bz_1, b) = \overline{\beta}_1(a + bz_0, b)$. Now, referring to Eqs. (2.39) and (2.38), we have that the reparametrised colouring $\overline{\beta}_1'$ satisfies

$$\begin{aligned} w_2 &= D_1\overline{\beta}_2(a + bz_0, b)z_0 + D_2\overline{\beta}_2(a + bz_0, b) \\ &= D_1\overline{\beta}_1'(a + bz_1, b)z_1 + D_2\overline{\beta}_1'(a + bz_1, b) \end{aligned}$$

and

$$x_0 = D_1\overline{\beta}_1'(a + bz_1, b) = D_1\overline{\beta}_2(a + bz_0, b).$$

Therefore, we have that there exist $x_0, w_1, w_2 > 0$ such that $w_1 \neq w_2$, and $F(x_0, w_1) = F(x_0, w_2) = z_0$. This implies that $F(x_0, w_2) = z_1 \neq z_0$. But this is a contradiction since $F(x_0, w_2) = z_0$. Therefore, there exists a function $h : \mathbb{R}^2 \to \mathbb{R}$ such that $h(D_1\overline{\beta}(a + bz, b), z)) = D_2\overline{\beta}(a + bz, b)$ for all $z > 0$, $a \in A$ and $b \in B$.

So, we have concluded that for planar scenes at depth z, $h(D_1\overline{\beta}(a + bz, b), z)) = D_2\overline{\beta}(a + bz, b)$ for some function h. It remains to show that such a h does not depend on z. Suppose that $D\overline{\beta}(a + bz, b) = (x, y)$. Then $h(x, z) = y$. Let $z' \neq z$, and $z' > 0$. Then $\beta' := \beta \circ \tau_{(bz,z)} \in B_{\mathbf{P}_{z'}}$ and $D\overline{\beta}'(a + bz', b) = (x, y)$. Therefore, if $h(x, z) = y$ for some $z > 0$, $h(x, z') = y$ for all $z' > 0$. Therefore, defining $g(x) := h(x, 1)$ gives the result.

Finally, we show that $g(x)$ is identically 0 therefore obtaining the result that $D_2\overline{\beta}(p, q) = 0$ for all reparmetrised colourings $\overline{\beta}$ generated from planar scenes $(\mathbf{P}_z, \beta)$. At this point, we have for all $z > 0, \beta \in B_{\mathbf{P}_z}, k \neq 0, P \in \mathbf{P}_z, Q \in \mathbf{Q}_z$:

$$\begin{aligned} g(k \cdot D_1[\overline{\beta}(p, q)]) &= g(D_1[k\overline{\beta}(p, q))]) \\ &= D_2[k\overline{\beta}(p, q)] \\ &= kD_2\overline{\beta}(p, q) \\ &= kg(D_1\overline{\beta}(p, q)). \end{aligned}$$

Therefore, g is linear and there exists a $c \in \mathbb{R}$ such that $g(x) = c \cdot x$. Now, For all $z > 0$ and all $P \in \mathbf{P}_z$ and all $Q \in \mathbf{Q}_z$ and all $\beta \in B_{\mathbf{P}_z}$, we have $g(D_1\overline{\beta}(p, q)) =$

$cD_1\overline{\beta}(p,q) = D_2\overline{\beta}(p,q)$, for some $c \in \mathbb{R}$ Therefore, every $\beta \in B_{\mathbf{P}_z}$ is a solution of the transport equation $cD_1\overline{\beta}(p,q) - D_2\overline{\beta}(p,q) = 0$. Therefore, from a standard result of transport equations [23], for every $\beta \in B_{\mathbf{P}_z}$ there is some function $f : \mathbb{R} \to \mathbb{R}$ such that $\overline{\beta}(p,q) = f(p+cq)$. Therefore, for every $\beta \in B_{\mathbf{P}_z}$, $\overline{\beta}$ is constant along the lines $\{(s,-cs) : s \in \mathbb{R}\} \subset \mathbf{P}_z \times \mathbf{Q}_z$. However, if $c \neq 0$, this contradicts our assumption that every $B_{\mathbf{P}_z}$ contains at least one β such that $\overline{\beta}(p,q) \neq \overline{\beta}(p',q')$ if $p \neq p'$. Therefore, $c = 0$, and so for any $z > 0, P \in \mathbf{P}_z, Q \in \mathbf{Q}_z, \beta \in B_{\mathbf{P}_z}$, we have $D_2\overline{\beta}(p,q) = 0$, and so $\overline{\beta}$ is Lambertian. □

At this point we have proven that every colouring of a planar scene in the scene class is textured and Lambertian. It remains to show that the same is true for every other scene in the scene class.

Proof. (Theorem 2)

To show that these conclusions of Propositions 6 and 7 are true of any other scene in the scene class, we will make use of the fact that if $\operatorname{sgn}(x) = \operatorname{sgn}(w)$ then $F(x,w) = \frac{w}{x}$. To show this, we first note that for any planar scene $\mathbf{P}_z$, we have that $F(D_1\overline{\beta}(a+bz,b), D_1\overline{\beta}(a+bz,b)z + D_2\overline{\beta}(a+bz,b)) = z$, and $D_2\overline{\beta}(a+bz,b) = 0$. Now note that for any a,b, and any $z > 0$ and any $x \neq 0$ there exists a $\beta \in B_{\mathbf{P}_z}$ such that $D_1\overline{\beta}(a+bz,b) = x$, because $D_1\beta(a+bz) \neq 0$ and $k \cdot \beta \in B_{\mathbf{P}_z}$ for all $k \neq 0$. Therefore, the function F satisfies

$$F(x, x \cdot z) = z \tag{2.40}$$

for every $x \neq 0$ and every $z > 0$. Therefore, we must have that $\operatorname{sgn}(x \cdot z) = \operatorname{sgn}(x)$ because $z > 0$. Therefore, for every point in the set

$$\{(x,w) \in \mathbb{R}^2 \mid x \neq 0, \operatorname{sgn}(x) = \operatorname{sgn}(w)\},$$

the function F satisfies $F(x,w) = \frac{w}{x}$.

Finally, let us use the notation (X', β') for a general scene in the scene class. We can now show that for a given $(X', \beta') \in \mathbb{X}$, if the light-field L' generated from this coloured scene satisfies $F(D_bL'(a,b), D_aL'(a,b)) = \gamma(a,b)$, then β' is textured and Lambertian. We have for a general scene that

$$D_aL'(a,b) = D_1\overline{\beta}'(a+b\gamma'(a,b),b) + D_1\overline{\beta}'(a+b\gamma'(a,b),b) \cdot b \cdot D_1\gamma'(a,b), \tag{2.41}$$

$$\begin{aligned} D_bL'(a,b) = {} & D_1\overline{\beta}'(a+b\gamma'(a,b),b) \cdot (\gamma'(a,b) + bD_2\gamma'(a,b)) \\ & + D_2\overline{\beta}'(a+b\gamma'(a,b),b). \end{aligned} \tag{2.42}$$

For any given $(X', \beta') \in \mathbb{X}$ and for any given viewable surface ray $(P', \eta') \in X' \times S^1$, there exists a $\xi \in SE(2)$ and a $z_0 > 0$ such that: the embedded tangent space T_PX of

$X = \xi \cdot X'$ at $P = \xi \cdot P'$, satisfies

$$\mathbf{P}_{z_0} = T_P X,$$

and

$$\xi \cdot (P', \eta') = (a + bz_0, \eta(b))$$

for some $(a, b) \in A \times B$. Let $X = \xi \cdot X'$, $\beta = \xi \cdot \beta'$, $P = \xi \cdot P'$, $\eta = \xi \cdot \eta'$, $\gamma = \xi \cdot \gamma'$ for a $\xi \in SE(2)$ and $z_0 > 0$ satisfying the previous property. For the ray parametrised by $(a, b) \in A \times B$ that corresponds to the surface ray $(P, \eta) = ((a + bz_0, z_0), \eta(b))$, we have that $D\gamma(a, b) = (0, 0)$. Therefore, by using Eqs (2.41) and (2.42), for such an (a, b), we have that

$$D_a L(a, b) = D_1 \overline{\beta}(a + b\gamma(a, b), b), \tag{2.43}$$

$$D_b L(a, b) = D_1 \overline{\beta}(a + b\gamma(a, b), b)\gamma(a, b) + D_2 \overline{\beta}(a + b\gamma(a, b), b). \tag{2.44}$$

There are now two cases: either

$$\left|D_2 \overline{\beta}(a + b\gamma(a, b), b)\right| \geq \left|D_1 \overline{\beta}(a + b\gamma(a, b), b)\gamma(a, b)\right|$$

or

$$\left|D_2 \overline{\beta}(a + b\gamma(a, b), b)\right| < \left|D_1 \overline{\beta}(a + b\gamma(a, b), b)\gamma(a, b)\right|.$$

Suppose the former case that $\left|D_2 \overline{\beta}(a + b\gamma(a, b), b)\right| \geq \left|D_1 \overline{\beta}(a + b\gamma(a, b), b)\gamma(a, b)\right|$. Let $\{(X_z, \beta_z)\}_{z>0}$ be the family of scenes given by freely translating (X, β) freely along the line parametrised by (a, b) so that

$$(X_z, \beta_z) := \tau_{(bz,z)}(X, \beta).$$

We have that for each (X_z, β_z),

$$\gamma(a, b) = z$$

yet

$$D_2 \overline{\beta}_z(a + b\gamma(a, b), b) = D_2 \overline{\beta}(a + b\gamma(a, b), b)$$

and

$$D_1 \overline{\beta}_z(a + b\gamma(a, b), b) = D_1 \overline{\beta}(a + b\gamma(a, b), b).$$

Therefore, for all $z > 0$ we have $\left|D_2 \overline{\beta}_z(a + bz, b)\right| \geq \left|D_1 \overline{\beta}_z(a + bz, b)z\right|$. But because $D\overline{\beta}_z(a + bz, b)$ is invariant to z, the only solution is that $D_1 \overline{\beta}_z(a + bz, b) = D_2 \overline{\beta}_z(a + bz, b) = 0$. But then $F(0, 0) = z$ for all z, but this is a contradiction because F is a function. Therefore $\left|D_2 \overline{\beta}(a + b\gamma(a, b), b)\right| < \left|D_1 \overline{\beta}(a + b\gamma(a, b), b)\gamma(a, b)\right|$.

Therefore, the only possible case is that $D_1\overline{\beta}(a + b\gamma(a,b), b) \neq 0$ since $\gamma(a,b) > 0$. This implies that $\text{sgn}(D_aL(a,b)\gamma(a,b)) = \text{sgn}(D_bL(a,b))$, and so

$$F(D_aL(a,b), D_bL(a,b)) = \frac{D_bL(a,b)}{D_aL(a,b)}.$$

Therefore,

$$\gamma(a,b) = \gamma(a,b) + \frac{D_2\overline{\beta}(a + b\gamma(a,b), b)}{D_1\overline{\beta}(a + b\gamma(a,b), b)},$$

which implies that

$$D_2\overline{\beta}(a + b\gamma(a,b), b) = 0.$$

Therefore, the surface ray (P, η) is textured and Lambertian. Note that the fact that rigid body transformations of the scene do not alter these properties for the transformed ray (P', η'), following directly from the fact that $\xi \cdot \beta(\xi \cdot P', \xi \cdot \eta') = \beta(P', \eta')$ for all $(P', \eta') \in X \times S^1$. Because rigid body motions of scenes will not alter these properties, (P', η') is a textured and Lambertian surface ray of (X', β') as well. □

2.6 Conclusion

In this chapter, a thorough investigation of light-field cameras, their history, and their respective geometry was provided. These investigations resulted in a point-to-disc projection model that is invertible, and will be used in Chapter 3 to perform calibration. These investigations were followed by examining the concept of disparity and a method of estimating disparity from raw light-field data, by taking into account the partial differential equation that disparity fields satisfy. Theoretical derivations of necessary and sufficient conditions have been provided that under certain mild assumptions, any depth estimation technique relying on light-field gradients alone must assume that the scene being imaged is textured and Lambertian. These proofs will apply to equivalent scenarios in the general structure-from-motion problem. Extensions to this work are described in Chapter 6.

Chapter 3

Plenoptic Camera Calibration

This chapter proposes a new method for estimating calibration parameters of plenoptic cameras by minimizing the nonlinear plenoptic reprojection error. The plenoptic disc features defined in Chapter 2 are in a natural one-to-one correspondence with physical points in front of the camera. We exploit the intrinsic geometry of plenoptic cameras in a novel projection model that relates the plenoptic disc features to physical points. The resulting calibration quality, as quantified by mean reprojection error and 3D reconstruction error, outperforms recently published results. The work in this chapter is based on the paper [79].

3.1 Background

Calibration estimation is a fundamental problem in computer vision. Accurate calibration, both intrinsic and extrinsic, is essential for the generation of metrically correct[1] scene reconstructions, as well as being crucial in other preprocessing tasks. Due to the complexity of the lenslet geometry of plenoptic cameras, existing multi-camera calibration methods are not directly effective and there is a growing literature aimed at developing effective models for calibration [125, 101, 128, 126, 44, 35, 100, 43, 9, 75, 7].

Most existing calibration techniques, for both plenoptic and other sorts of cameras, consist of three main steps and are based on estimating a projective transformation that models the camera for a ray-based model of light. The first step takes in as data raw images, and estimates the locations of features in these images generating a list of correspondences between frames. The second step is initialisation, generating an initial estimate of the calibration parameters. A cost function, typically mean reprojection error, is then minimised in the third, optimisation step.

This chapter proposes a new calibration method for plenoptic light-field cameras

[1] By 'metrically correct' we mean that the the distances between points in the scene reconstruction are equal to the distances between the corresponding points on the scene being measured.

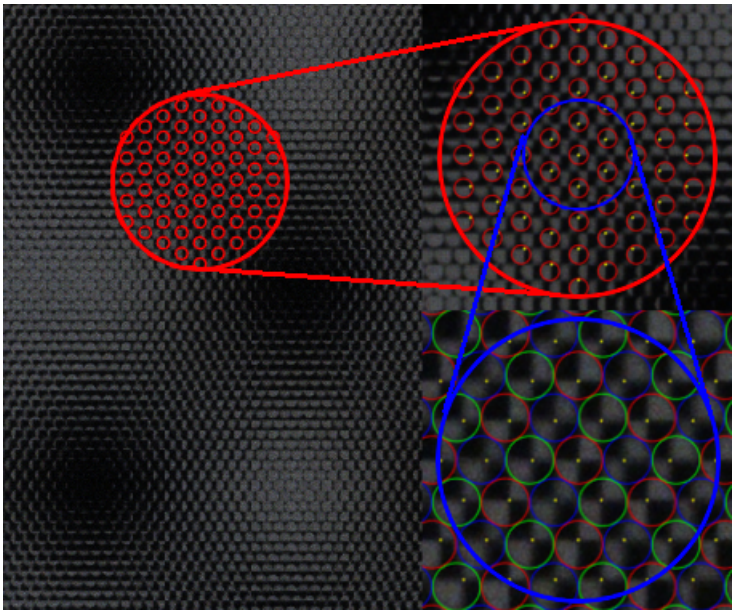


Figure 3.1: A zoomed view of a raw light-field image taken by a Raytrix R42 camera. In the large red circle is a plenoptic disc containing the set of lenslets that can see a specific feature. The yellow dots highlight the detected feature. Subimages corresponding to different lenslet types are shown in the blue circle in different colors.

that outperforms existing methods on the key performance measures of mean reprojection error and 3D reconstruction error. The first step of most existing calibration methods requires matching of subimage feature points between subimages, and data association between these image points between different lightfields (image frames) of a known target. As there can be dozens of subimages in which a given feature is visible per frame (see Fig. 3.1), the identification of subimage features becomes an onerous task [75]. This process is made more difficult by the fact that the different focal lengths used for multi-focus light-field cameras [61] mean that many of the subimages are out-of-focus for each of the images. Furthermore, since each subimage has a small resolution there is limited information available to make highly accurate feature extraction, especially when a feature point approaches the edge of a subimage. Reliable and robust extraction of accurate subimage feature points for lightfield calibration is a key limitation to existing calibration methods. Our proposed method avoids this problem by exploiting the geometric point projection model proposed in Chapter 2 in order to extract the plenoptic disc features corresponding to the corners of a checkerboard. This feature type has similarities to a feature type suggested by Dansereau *et al.*[22], however that paper does not use such feature types for calibration.

Exploiting this feature parametrisation, together with the simplified 3-intrinsic

parameter model originally proposed by Bok *et al.*[9], vastly simplifies the optimisation problem used in the final step. The initialisation procedure is based on the work by Bok *et al.*[9], reformulated to use our plenoptic disc features. The cost function minimised in our final step is a *plenoptic reprojection error*, that minimises distance between the coordinates of the plenoptic disc features used as data and the expected coordinates of these features given camera parameter estimates.

The mean reprojection and reconstruction errors that we obtain from this approach outperform state-of-the-art results [9, 75]. In summary, the main contributions in this chapter are:

- Derivation of a feature estimator for plenoptic disc features of checkerboards.
- A calibration method based on this geometry that outperforms existing state-of-the-art techniques in terms of accuracy and robustness.

3.1.1 Previous Work

Seminal work in the calibration of lenslet based plenoptic cameras was published in 2013 [22, 44]. Dansereau *et al.* used the Lytro plenoptic camera and derived a camera rectification formulation that allowed a simple optimisation algorithm for image calibration. A similar approach is undertaken more recently in Zhe *et al.*[43]. An advantage of this approach is that the resulting calibration optimisation tends to be more robust, however, the parameters identified are less directly associated with physical parameters of the camera. Moreover, the approach is less well suited to multi-focal light-field cameras.

Johannsen *et al.* [44] formulated a general reprojection model in terms of the physical parameters of a Raytrix camera. This work considered a relatively simple model of lens distortion and required careful initialisation of the optimisation to converge. Strobl *et al.* [100] recognised the fragility of the calibration optimisation and proposed a step-wise calibration approach where first the focal length and optical centre of the main lens is determined (as well as some distortion parameters) before the internal offset of the Micro-Lens Array (MLA) from the sensor and main lens respectively are determined. Sun *et al.* [101] use a similar approach, where they hand determine the ratio of MLA distance to sensor with respect to MLA distance to the image plane for a specific point, allowing them to effectively identify the relative focal length of the main lens separately form the calibration process. Another recent contribution is proposed by Zeller *et al.*[125, 126]. Although the focus of these papers is on visual odometry, they require a calibrated camera to provide metric reconstructions. The depth calibration proposed in Zeller *et al.*[126] uses a separate

optimisation process. Another direction stemming from this approach has lead to the consideration of more sophisticated models of the lens distortion and non-planarity of the MLA. Heinze *et al.* [35] consider more sophisticated models of the distortion of the main lens. Zhang *et al.* [128] consider a detailed model of the lenslet array geometry that calibrates for non-planarity of the array. Lenslet based plenoptic cameras, however, are constructed with careful attention to the coplanarity of the lenslet array and the image plane [74], and for cameras such as the Raytrix R42, this additional complexity is not required.

All the above papers require matching of point features across multiple images and multiple subimages. Although many of the methods use standard feature extraction methods to automate the matching process, there are necessarily errors in the identification and data association of these features. Bok *et al.* [9] introduced novel line features to improve the automation and accuracy of the feature identification. More recently Nousias *et al.* [75] developed corner based features along with an end-to-end calibration process. Both these papers have achieved improved performance through automation and accurate identification of feature correspondences. These papers provide a good benchmark for the evaluation of the present work, particularly since they minimize a mean reprojection error criterion as we do.

3.2 Plenoptic Camera Calibration

Calibration of a plenoptic camera has three main blocks, see Fig. 3.2: the first is a feature estimation block. In this chapter, we estimate the plenoptic disc data corresponding to corners of a checkerboard (cf. Sections 2.3 and 3.2.1). The second block is an initialisation block, that produces a calibration parameter estimate (cf. Section 3.2.2). The third block is a non-linear optimisation routine that refines the initial estimate produced by the second block (cf. Section 3.2.3).

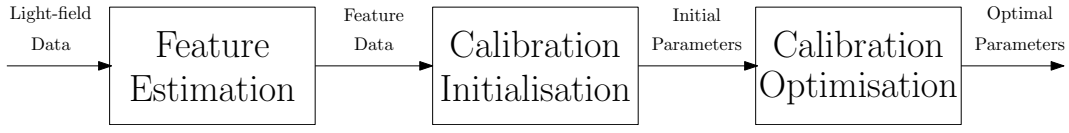


Figure 3.2: A block diagram of a generic calibration method.

3.2.1 Feature Estimation

We propose a novel plenoptic feature estimation method that avoids problems associated with identifying features in the low-resolution subimages by instead using higher-resolution sub-aperture images [29] that are computed from raw light-field

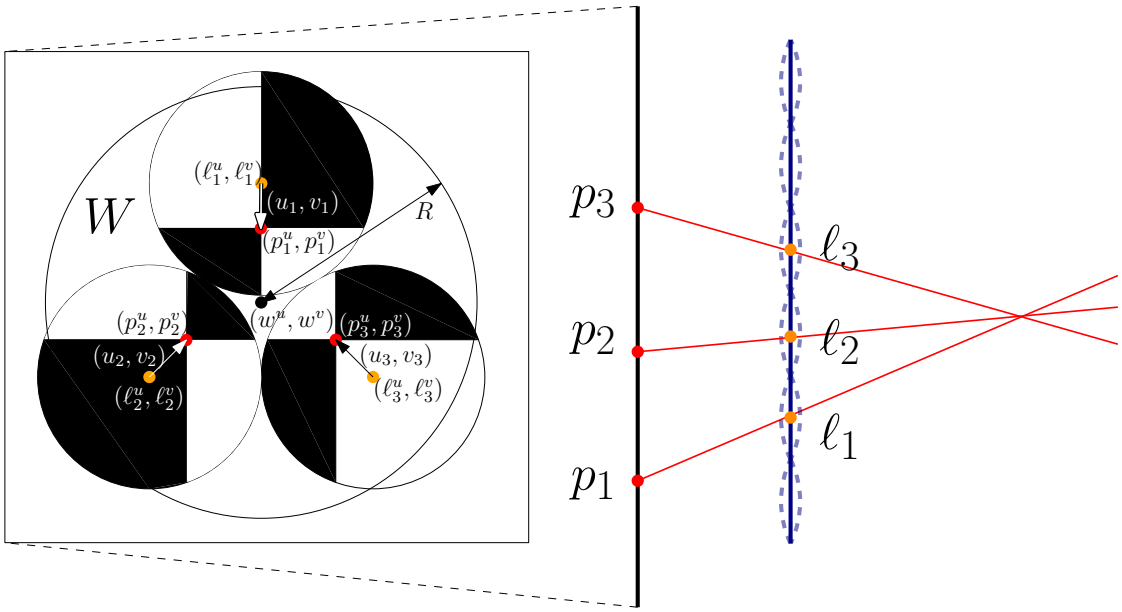


Figure 3.3: The centre (w^u, w^v) and radius R of a plenoptic disc W is shown on a raw light-field image, cf. also Fig. 3.5. The three lenslets in W are labelled by their lenslet coordinates (ℓ_1^u, ℓ_1^v), (ℓ_2^u, ℓ_2^v), (ℓ_3^u, ℓ_3^v) that are in the centres of the subimages of these lenslets. The pixels (p_1^u, p_1^v), (p_2^u, p_2^v), and (p_3^u, p_3^v) within the subimages of each of these lenselets corresponding to the same feature point are depicted. These pixels have offsets (u_1, v_1), (u_2, v_2), and (u_3, v_3) from the subimage centre, repectively.

data. Other papers [9, 75] extract features from the low-resolution subimages. An additional advantage of the proposed method is lower memory usage. Our high-resolution Raytrix data is ill-suited for the method proposed by Bok *et al.*[9], as the template size used in that paper grows geometrically with subimage radius, resulting in that algorithm failing to terminate in our experiments.

We assume that the calibration grid consists of M interior corners, and that a corner point has body-fixed-frame coordinates P_i indexed by $i = 1, \ldots, M$. The first step of calibration is estimation of the plenoptic disc feature data (w_i^u, w_i^v, R_i) as defined in Section 2.3, corresponding to each of the corner points P_i of the calibration grid for the given raw light-field image. This step is repeated for each raw light-field image.

The feature estimation process starts by obtaining a set of N sub-aperture images I_k indexed by $k = 1, \ldots, N$ by selecting from each lenslet subimage the pixel with constant offset (u_k, v_k) from the subimage centre and stitching the resulting image together. Because the lenslets are arranged in an hexagonal lattice, generating a rectangular sub-aperture image from the constant offset pixels of each subimage requires interpolation. The dimensions of the subimage are $\frac{U}{r}$ and $\frac{V}{r}$, where $U \times V$

is the dimension of the raw light-field image and r is the lenslet subimage radius.

For the interpolation, the colour $I_k(q^u, q^v)$ assigned to pixel (q^u, q^v) in the sub-aperture image I_k is given by finding the three nearest lenslet coordinates (ℓ_1^u, ℓ_1^v), (ℓ_2^u, ℓ_2^v), and (ℓ_3^u, ℓ_3^v), of lenslets in the subimage to $r(q^u, q^v)$. Then, $r(q^u, q^v)$ is expressed as a convex sum of these coordinates so that $r(q^u, q^v) = \alpha_1(\ell_1^u, \ell_1^v) + \alpha_2(\ell_2^u, \ell_2^v) + \alpha_3(\ell_3^u, \ell_3^v)$, where $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Then we assign the colour to pixel q that respects this convex sum, so that $I_k(q^u, q^v) = \alpha_1 L(\ell_1^u + u_k, \ell_1^v + v_k) + \alpha_2 L(\ell_2^u + u_k, \ell_2^v + v_k) + \alpha_3 L(\ell_3^u + u_k, \ell_3^v + v_k)$.

Running a standard checkerboard detector on each of the resulting sub-aperture images I_k gives a list of detected checkerboard corner features $F_k = \{(q_{i,k}^u, q_{i,k}^v)\}_{i=1}^M$ appearing in the image. Note that the set F_k of point features is indexed by the offset index k, whereas individual point features within F_k are indexed by both the point index i and the offset index k.

Scaling any detected corner feature $(q_{i,k}^u, q_{i,k}^v) \in F_k$ by r gives the lenslet coordinates $(\ell_{i,k}^u, \ell_{i,k}^v)$ of the lenslet $\ell_{i,k}$ for which P_i is visible in the subimage of $\ell_{i,k}$ with offset (u_k, v_k) from the subimage centre $(\ell_{i,k}^u, \ell_{i,k}^v)$. As P_i is visible in the subimage of $\ell_{i,k}$, P_i projects to some $p_{i,k}$ defined by equations (2.2) and (2.5). This pixel has pixel coordinates $(p_{i,k}^u, p_{i,k}^v) = (\ell_{i,k}^u, \ell_{i,k}^v) + (u_k, v_k)$, because $q_{i,k}$ was found in a sub-aperture image generated with constant offset (u_k, v_k).

Therefore, from a raw light-field image we can obtain a collection of lenslet-pixel pairs $\{((\ell_{i,k}^u, \ell_{i,k}^v), (p_{i,k}^u, p_{i,k}^v))\}_{i=1,k=1}^{M,N}$ corresponding to the N corners as seen in M sub-aperture images.

Now, for any two of these obtained lenslet-pixel pairs $((\ell_{i,k}^u, \ell_{i,k}^v), (p_{i,k}^u, p_{i,k}^v))$ and $((\ell_{i,k'}^u, \ell_{i,k'}^v), (p_{i,k'}^u, p_{i,k'}^v))$ corresponding to some P_i, we note that

$$\begin{pmatrix} p_{i,k}^u - p_{i,k'}^u \\ p_{i,k}^v - p_{i,k'}^v \end{pmatrix} = \left(1 + \frac{r}{R_i}\right) \begin{pmatrix} \ell_{i,k}^u - \ell_{i,k'}^u \\ \ell_{i,k}^v - \ell_{i,k'}^v \end{pmatrix}. \tag{3.1}$$

Note that when $(u_{k'}, v_{k'}) = (0, 0)$, we have $(\ell_{i,k'}^u, \ell_{i,k'}^v) = (w_i^u, w_i^v)$, because a point feature will appear in the centre of the subimage of the plenoptic disc centre. Therefore, (3.1), with $(u_{k'}, v_{k'}) = (0, 0)$ provides a linear system of equations that can be used to estimate the plenoptic disc feature data (w_i^u, w_i^v, R_i) by solving

$$\begin{pmatrix} -1 & 0 & -\frac{u_k}{r} & \ell_k^u \\ 0 & -1 & -\frac{v_k}{r} & \ell_k^v \end{pmatrix} \begin{pmatrix} w_i^u \\ w_i^v \\ R_i \\ 1 \end{pmatrix} = 0, \tag{3.2}$$

where there are 2 rows in the data matrix for each offset index k for which the corner

P_i is successfully detected in the sub-aperture image I_k. In practice, we only use sub-aperture images I_k for which all M checkerboard corners are successfully detected.

In summary, the method for obtaining the plenoptic disc feature data estimates for a single raw light-field image is as follows:

1. For each pixel offset (u_k, v_k), generate the sub-aperture image I_k.

2. For each sub-aperture image I_k, run a standard corner detector to obtain a set of sub-aperture image features $q_{i,k}$.

3. Compute the corresponding lenslet-pixel pairs $((\ell^u_{i,k}, \ell^v_{i,k}), (p^u_{i,k}, p^v_{i,k}))$.

4. Using the lenslet-pixel pairs corresponding to a given point feature P_i, find the least-squares estimate of (w^u_i, w^v_i, R_i) by solving (3.2).

These steps are applied to every raw light-field image in the dataset.

3.2.2 Calibration Initialisation

Initialisation parameters are found by deriving a linear system of equations that perfect data from a single light-field image must satisfy, and solving the system for gathered data in a least-squares sense. Let $\xi_j \in SE(3)$ denote the pose of the camera with respect to the fixed frame $\mathbf{O}$ when it captures the raw light-field image frame j. Let ${}^{\mathbf{O}}P_i$ be the position of the corner of a checkerboard expressed in the coordinates of the fixed frame $\mathbf{O}$. At frame j, the corner P_i has coordinates $P_{i,j}$ in the body-fixed frame of the camera. Using (2.10) with $f^u = f^v = f$, we obtain the relations

$$w^u P^z_{i,j} + f P^x_{i,j} = 0 \tag{3.3}$$

$$w^v P^z_{i,j} + f P^y_{i,j} = 0 \tag{3.4}$$

$$(rK_1 + R)P^z_{i,j} + rK_2 = 0. \tag{3.5}$$

Denoting $\xi_j^{-1} = (\Omega_j, c_j)$ with Ω_j the rotational part and c_j the translational part corresponding to the location of the optical centre of the camera for frame j, we have

$$\begin{pmatrix} P^x_{i,j} \\ P^y_{i,j} \\ P^z_{i,j} \end{pmatrix} = \begin{pmatrix} \Omega^{11}_j & \Omega^{12}_j & \Omega^{13}_j \\ \Omega^{21}_j & \Omega^{22}_j & \Omega^{23}_j \\ \Omega^{31}_j & \Omega^{32}_j & \Omega^{33}_j \end{pmatrix} \begin{pmatrix} {}^{\mathbf{O}}P^x_i \\ {}^{\mathbf{O}}P^y_i \\ {}^{\mathbf{O}}P^z_i \end{pmatrix} + \begin{pmatrix} c^x_j \\ c^y_j \\ c^z_j \end{pmatrix}. \tag{3.6}$$

Substituting the expressions for $P_{i,j}$ into (3.3) - (3.5), we obtain the system

$$\left(\begin{pmatrix} w^u_{i,j} & 1 & 0 & 0 \\ w^v_{i,j} & 0 & 1 & 0 \\ R_{i,j} & 0 & 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} {}^{\mathbf{O}}P^x_i & {}^{\mathbf{O}}P^y_i & 1 \end{pmatrix}\right) \begin{pmatrix} \Omega^{31}_j \\ \Omega^{32}_j \\ c^z_j \\ -f\Omega^{11}_j \\ -f\Omega^{12}_j \\ -fc^x_j \\ -f\Omega^{21}_j \\ -f\Omega^{22}_j \\ -fc^y_j \\ rK_1\Omega^{31}_j \\ rK_1\Omega^{32}_j \\ rK_2 + rK_1c^z_j \end{pmatrix} = 0. \tag{3.7}$$

Here, $\otimes$ denotes the Kronecker product. For each frame j, this system is solved for the vector which satisfies the equation in a least sqaures sense. Extracting the initial intrinsics and extrinsics from the resulting singular vectors follows the same procedure as in Bok *et al.*[9].

3.2.3 Calibration Optimisation

The plenoptic disc feature data estimates are used as data in a non-linear optimisation routine, where the intrinsics and extrinsics of the camera are the parameters being estimated. Initial parameter estimates are given using the linear solution obtained in Section 3.2.2. The separate f-parameters f^u and f^v are initialised with the same f given in the initialisation step, the initial estimate of the optical centre (c^u, c^v) is the centre of the light-field image, and the initial lens distortion parameter k_1 is 0.

The error function that is minimised in this routine comes from (2.10), and (3.6), where the plenoptic disc feature data $(w^u_{i,j}, w^v_{i,j}, R_{i,j})$ have been estimated using the method discussed in Section 3.2.1, and world-frame point locations ${}^{\mathbf{O}}P_i$ are known. Let $\phi = (K_1, K_2, f^u, f^v, c^u, c^v, k_1)$ be the intrinsics of the camera, $\Xi = (\xi_j)_{j=1}^T$ be the list of extrinsics of the camera where T is the total number of frames used, and $\mathrm{Y} = ((w^u_{i,j}, w^v_{i,j}, R_{i,j}, {}^{\mathbf{O}}P_{i,j}))^{M,T}_{i,j=1,1}$ be the known data. We minimise the *plenoptic reprojection error*, given by

$$\epsilon(\phi, \Xi; \mathrm{Y}) = \sum_{i,j} (\Pi_\phi(P_{i,j}) - (w^u_{i,j}, w^v_{i,j}, R_{i,j}))^2 \tag{3.8}$$

where $P_{i,j} = \xi_j^{-1}\, {}^{\mathbf{O}}P_i$, given by (3.6), and Π_ϕ denotes the plenoptic projection (2.10) with lens distortion modelled by (2.14), and parameters given by the intrinsics ϕ.

Little is currently known of the analytical properties of (3.8). In practice, minimising this error using standard optimisation techniques such as Levenberg-Marquardt gives good results. However questions such as whether the estimates tend to a global minimum – or even whether a unique global minimum exists at all – given the initial estimate provided by solving (3.7) are good candidates for future investigations.

3.3 Results

In this section we compare the proposed calibration algorithm to existing state-of-the-art methods [9, 22, 75]. Our code is publicly available[2].

3.3.1 Experimental Data

For our obtained datasets, a Raytrix R42 camera was used with a Kowa LM35SC 35mm focus lens [52]. The approximate focus distance was set to 0.25m, 0.5m, and 1m. We call these datasets R-A, R-B, and R-C, respectively.

A standard checkerboard was used as a calibration grid for the various experiments. As at shorter focal distances, the camera needed to be closer to the calibration target, the grid sizes for datasets R-A, R-B, and R-C were 4mm, 6mm, and 15.5mm, respectively. These datasets contained 24, 22, and 18 images, respectively. For datasets R-A and R-C the checkerboards contained 15 by 10 feature points. For dataset R-B, the checkerboard contained 6 by 8 feature points. The obtained resolutions of the raw light-field images were 7716 pixels (width) by 5364 pixels (height). De-bayering and colour-correction was conducted upon capture using Raytrix software.

Lenslet types were identified in the raw light-field images using a standard method. Since the pinhole models for each lenslet type are identical, we consider only calibration for the lenslet types where the calibration grid is in best focus. Exploiting the multi-focal arrangement would likely improve the feature-extraction process but was not considered in this work.

We also compared results on the datasets given by Dansereau *et al.*[22]. In those datasets, referred to here as L-B, L-D, and L-E, a Lytro camera was used. These datasets were chosen because each used a different focal distance and contained light-fields at a wide variety of poses and raw light-field images of varying degrees of focus.

The cells in Table 3.1 for Nousias *et al.* [75] comparing against the Lytro datasets were left blank in Table 3.1 because the small resolution of the Lytro subimages produces poor results for the feature-detection method of Nousias *et al.*[75]. As the

[2]Available at: https://github.com/sgpobrien/PlenCalToolbox

method proposed in Nousias *et al.*[75] relies on detecting features in subimages, it is expected that their method would not fare well with these datasets. It is noted that the focal lenses of Lytro cameras have far higher lens distortion than the focal lenses used by Raytrix cameras. Since the method in Nousias *et al.*[75] does not model lens distortion, comparing on Lytro data would be an unfair comparison. As dataset R-A also contained many in-focus lightfields where features are not visible in subimages, we did not compare against the method of Nousias *et al.*[75] on this dataset.

As the method proposed by Dansereau *et al.* [22] relies on a particular formatting of the raw light-field images, it does not produce estimates for the Raytrix datasets, and as such the corresponding cells in Table 3.1 are left blank. On our version of Matlab, running the code of Dansereau *et al.*[22] on dataset L-D resulted in an exception being thrown to do with inconsistent orientations of the detected checkerboard, so these cells are left blank. For the same formatting reason, comparison of Bok *et al.* [9] on the Raytrix datasets was not conducted. We did not run the code of Bok *et al.*[9] on the dataset L-D because, as the authors note in that paper, their feature detection method does not work for in-focus light-fields, of which there are many in dataset L-D.

3.3.2 Performance Measures

The aim of these experiments was to test the accuracy of our calibration under a variety of conditions. However, the sparsity of Table 3.1 indicates that there is no standard performance measure to verify calibration methods for plenoptic cameras. Comparisons were made difficult by the wide variety of types of errors reported between each of the calibration methods. Our method calculates the widest variety of errors, rather than only the error being optimised. Note that any method that optimises a given error should have a natural advantage when that error is used as a performance measure. As such, we compare against a variety of errors rather than our optimised plenoptic reprojection error. Mean 3D reconstruction error (M3DE) is calculated by taking the average of the distances of point estimates from the actual points and dividing by the depth of the actual point in the estimated camera coordinate system. Note that plenoptic cameras allow single-image 3D reconstructions, making this a sensible measure. Mean reprojection error (MRE) is the average of the distances between extracted subimage corner feature coordinates and reprojected corner feature coordinates on the raw image, an example of which is given in Fig. 3.5. Mean sub-aperture reprojection error (MSRE) is the average of distances between extracted feature coordinates in sub-aperture images and reprojected feature coordinates onto those sub-aperture images.

As the code provided by each of the other methods [75, 22, 9] does not produce 3D reconstructions we derived several methods based on the projection models used in these papers and the obtained feature data of these methods. For Nousias *et al.*[75], the reconstruction method finds the point that best fits their projection model, given a set of lenslet-pixel pairs known to correspond with that point. The reconstruction method used for Dansereau *et al.* [22] is based on calculating for all the lenslet-pixel pairs corresponding to a point P, the ray corresponding to that lenslet-pixel pair and finding the point that minimises the sum of distances to all of these rays. We were unable to find a reliable reconstruction method for the data obtained by Bok *et al.* [9].

The MSRE method used for Dansereau *et al.* [22] in Table 3.1 is based on solving their projection model given known ideal checker positions and pixel offsets for the unknown lenslet coordinates, which has a direct solution in their projection model, then applying their distortion model.

3.3.3 Discussion

In Table 3.1 we show the results of our calibration method compared against other existing state-of-the-art methods. Our method runs on the widest variety of datasets, and most consistently produces the smallest errors. The code we compared our method to were supplied by Nousias *et al.* [75], Dansereau *et al.* [22], and Bok *et al.* [9].

In Table 3.1 we compare each of the methods on different performance measures where it was possible to do so with the feature data produced by these methods. We first compared the methods on the measure of M3DE. As we were not able to provide reconstructions for the method of Bok *et al.* [9], this column is left blank. On this metric our method outperforms all the other methods except on dataset L-E, where the method of Dansereau *et al.* [22] performs better. This is likely due to their method implementing better preconditioning and higher-order lens distortion. One of our reconstructions together with extrinsics is shown in Fig. 4.4. The high M3DEs for Nousias *et al.* [75] are likely due to a flaw in their implementation discussed in the following paragraph.

In Table 3.1 we then compare our method with the method of Nousias *et al.* [75] on the measure of MRE. Although it was not possible to calculate this error using the feature data provided by the other methods [22, 9], our results for this measure are still shown. One likely factor affecting the accuracy of Nousias *et al.* [75] is preconditioning. It is noted that appropriate centering and scaling of parameters is often essential in order for Matlab-based optimisation algorithms to converge [67]. The algorithm implemented in Nousias *et al.*[75] does not implement any centering

Table 3.1: Table of error results. Best results per row are shown in bold. Measures that could not be computed are left blank.

Dataset	Ours	Nousias [75]	Dansereau [22]	Bok [9]
Mean 3D Reconstruction Error (%)				
R-A	**0.5206**			
R-B	**0.4482**	28.5775		
R-C	**1.4274**	53.7746		
L-B	**1.8642**		2.0419	
L-D	**4.2736**			
L-E	8.7459		**5.9599**	
Mean Reprojection Error (pixels)				
R-A	**0.9743**			
R-B	**0.2619**	2.0104		
R-C	**0.3832**	4.6925		
L-B	**0.3467**			
L-D	**1.2443**			
L-E	**0.2802**			
Mean Sub-aperture Reprojection Error (pixels)				
R-A	**0.5750**			
R-B	**1.0751**			
R-C	**0.6588**			
L-B	0.3427		**0.1775**	1.3125
L-D	**0.3061**			
L-E	**0.3514**		0.7383	0.3552

or scaling, and the results for [75] reported in Table 3.1, can be improved for each of the datasets using both our feature data and scaling factors. These scaling factors did not significatly improve their results when using their feature data, however, suggesting that there may also be inaccuracies in their feature estimation step. A comparison between the intrinsics obtained using our method versus Nousias *et al.* is shown in Table 3.2. The accuracy of our feature estimation step and reprojections are demonstrated in Fig. 3.5.

In Table 3.1 we compare our results to the other methods on the measure of MSRE. Although it was not possible to calculate this error using the feature data of Nousias *et al.* [75], our results for these datasets are still shown. Our method outperforms the other proposed methods on this metric with the exception of dataset L-B, where it is beaten by Dansereau *et al.* [22], likely due to their higher-order approximation of lens distortion. Note that although the MSRE is smaller than ours for this cell, its M3DE is larger, demonstrating the non-transitive relation between these measures.

Table 3.2: Intrinsic Parameters for Dataset R-B.

Var.	Ours	Nousias
K_1	-13.1706	-10.23
K_2	1.14×10^4	1.13×10^4
f^u	3.21×10^4	3.18×10^4
f^v	3.21×10^4	3.18×10^4
c^u (pix)	2675	2681
c^v (pix)	4415	3857
k_1	-1.7×10^{-10}	0

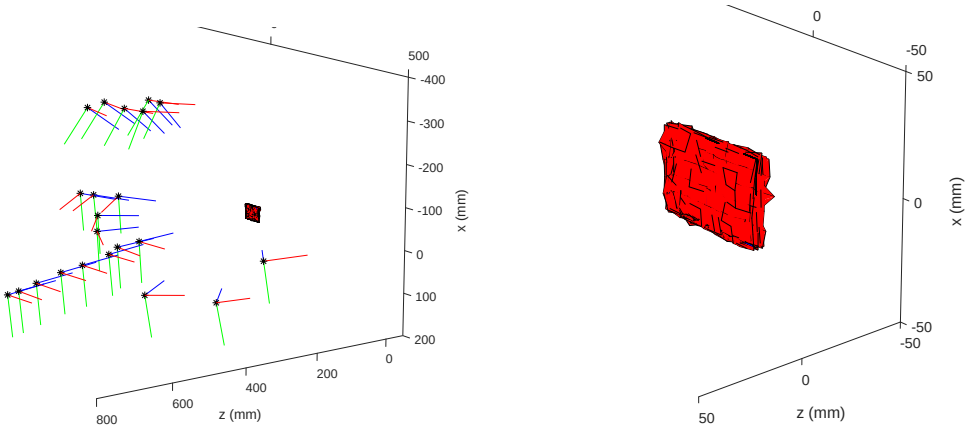


Figure 3.4: Calibration grid reconstruction and poses for dataset R-B. Camera faces forwards along blue axis.

3.4 Conclusion

In this chapter we develop a feature-extraction process to estimate the plenoptic discs associated with the corners of a checkerboard for use in plenoptic camera calibration, based on the geometry derived in Section 2.3. Our method produces both superior mean reprojection errors onto the raw light-field images and better mean reconstruction errors.

To the authors' knowledge, the proposed calibration implementation is the first that successfully and reliably runs with both Raytrix and Lytro data with only minor preprocessing required.

Along with better performance, our method provides a novel projection model that allows an easy translation between plenoptic disc features and physical 3D points, making it better suited for 3D reconstruction than ray-based approaches.

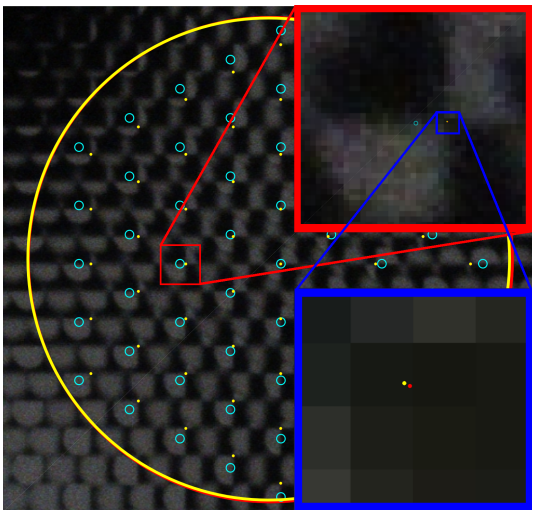


Figure 3.5: An example of a plenoptic disc feature and plenoptic reprojection of a point on a raw light-field image from dataset R-B. In the cyan circles are the lenslet coordinates (ℓ^u, ℓ^v) within a plenoptic disc W shown with red boundary and a reprojected plenoptic disc shown with yellow boundary. The red dots are the estimated subimage features corresponding to the plenoptic disc feature, and the yellow dots are the reprojected features.

Part II

Observers and Scene Reconstruction

Chapter 4

An Observer For Estimating an Explicit Scene Representation

This chapter proposes an observer for estimating point-cloud representations of scenes from a sequence of measurements acquired by a light-field camera. The observer is based on a gradient-descent methodology. A rigourous analysis of stability of the observer error is provided, and the observer is tested in simulation, demonstrating convergence behaviour. The work in this chapter was previously published in [77].

4.1 Introduction

In this chapter, we develop an observer for estimating a dense depth maps of an entire scene provided the camera motion and using light-field measurements as inputs. We follow a general design philosophy for observers by including dynamics of the depth map as an internal model and using the gradient of a disparity map as an innovation term. To the authors understanding, there is no prior work on applying the observer based approach to depth estimation using *plenoptic* camera data. The use of a moving camera combined with a dynamic observer is found in simulation to relax observability conditions, so that the knowledge of the motion of the light-field camera allows for the estimation of scenes that would not be observable using static depth estimation techniques due to insufficient texture on the scene. For such scenarios, points on the estimated scene will remain stationary until such a time when the camera is viewing these points in front of sufficiently textured regions of the scene. In this way we ensure that every point of the estimated scene converges to a point on the actual scene as long as we can guarantee that each point on the scene estimate is viewed at some time in the future.

In Section 4.2, we introduce additional light-field concepts specific to this chapter. In Section 4.3, we formulate the dynamics assigned to point estimates by the

observer. We then discuss in Section 4.4 details of the numerical implementation of the observer, and show its behaviour for a simple simulated scenario.

4.2 Notation and Terminology

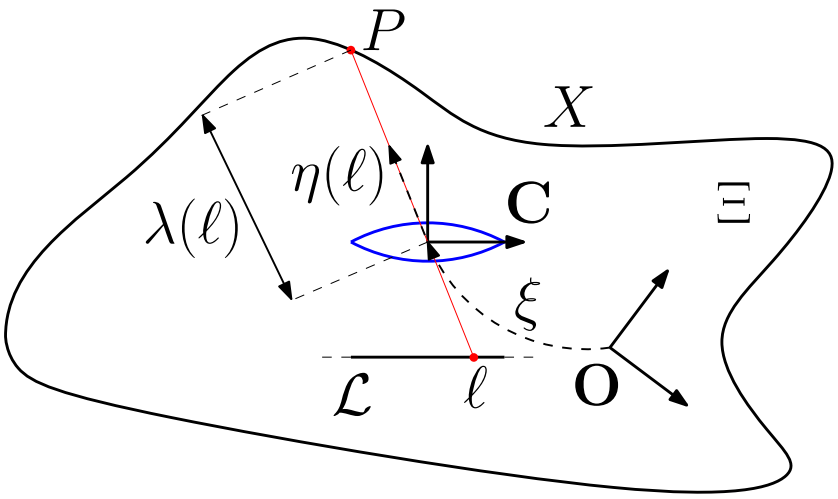


Figure 4.1: Illustration of the various notation used in this Chapter. Shown is a point P on a scene X defined by the boundary of the spatial environment Ξ. The camera pose ξ is defined with respect to reference frame $\mathbf{O}$, and the camera reference frame is $\mathbf{C}$. A lenslet ℓ on the lenslet plane $\mathcal{L}$ is shown, and the ray that passes through the center of the lenslet ℓ is $(\mathbf{c}, \eta(\ell))$. The distance of the scene X from $\mathbf{c}$ in direction $\eta(\ell)$ is $\lambda(\ell)$.

In this section we develop the geometric framework used to derive the photometric error term minimised by the observer. Although much of the notation and terminology used in this chapter was introduced in Chapters 1 and 2, some additional concepts specific to this chapter need to be introduced.

Previously, in Section 1.3.2, we had defined the set Ξ to be the set of possible extrinsic parameters of the camera. However because we do not impose restrictions on the rotational part of the pose of the camera, it is useful in this chapter to define the *spatial environment* Ξ as the set of possible positions of the camera, so that it is a subset of $\mathbb{R}^3$ rather than of $SE(3)$. This means that the possible extrinsic parameters of the camera, given by position-rotation pairs, are elements of $\Xi \times SO(3)$. All scenes X in this chapter are assumed to be given by the boundaries of spatial environments so that $X = \partial\Xi$ so that the camera is moving within the scene and looking outward. As in previous work on light-field cameras [20], we model the focus lens of the light-field camera as a thin-lens with *focal length* F. The pose ξ of the camera is given with respect to a fixed reference frame $\mathbf{O}$, and itself defines a body-fixed reference frame

C of the camera. The translational part of the pose is denoted **c** and is the position of the *optical centre* of the lens. The rotational part of the pose is denoted R (see Fig. 4.1). We define the camera as facing in the positive z-direction in the coordinate system **C**, and call the unit-vector pointing in this direction ν.

In Section 1.3.2, we had also introduced the concept of a *sensor plane* $\mathcal{S}$, and in Section 2.4 specified the geometry of this plane for a light-field camera. Remember that for a light-field camera, the sensor plane consists of a lenset plane, or pupilar plane, $\mathcal{L}$, and a pixel plane, or retinal plane $\mathcal{P}$, so that $\mathcal{S} = \mathcal{L} \times \mathcal{P}$. It is useful to define a map $\eta_\xi : \mathcal{L} \to S^2$ as the directional component of the ray that passes through both the optical centre of the camera and the lenslet ℓ. In this chapter, we also treat *range maps* slightly differently to what was previously introduced in Section 1.3.1. Instead of assigning ranges to every ray in the environment, we instead associate a range $\lambda_\xi(\ell)$ to every lenslet. We define $\lambda_\xi(\ell)$ as the distance from the optical centre **c** of the camera to the nearest point on the scene X in direction $\eta(\ell)$. When it is clear from context the subscript ξ will be dropped from both η and λ (see Fig. 4.1).

Similarly to the range map λ, it is convenient to define a "virtual" range map $\lambda' : \mathbb{R}^+ \times \mathcal{L} \to \mathbb{R}$ which defines the "virtual scene" $\iota(X)$. The algebra describing the perspective projection through each lenslet ℓ is simplified by expressing the distance of an image point $\iota(P)$ as its distance to the lenslet, see Fig. 4.2. Because of this, we define the virtual distance $\delta = \lambda'(\Delta, \ell)$, corresponding to a real distance $\Delta = \lambda(\ell)$ where $\ell \in \mathcal{L}$ is a lenslet, to be the distance of the point on the virtual scene $\iota(X)$ from the lenslet ℓ in direction $\eta(\ell)$. Note that $\lambda'(\Delta, \ell)$ can be negative, unlike the real distance Δ. With that, the virtual distance δ corresponding to distance Δ is given by

$$\delta = \lambda'(\Delta, \ell) = \frac{F \cdot (\Delta\eta(\ell) \cdot \nu))}{F - (\Delta\eta(\ell) \cdot \nu)} - \ell \cdot \eta(\ell). \tag{4.1}$$

Given a point $Q \in \mathbb{R}^3$ and a specified plane **P**, we define for any point $P \in \mathbb{R}^3$ that satisfies $P \cdot x < 0$ for all $x \in \mathbf{P}$ (meaning that Q is between P and **P**), the projection $\pi_Q^{\mathbf{P}}(P)$ as the point of intersection of the line passing through both Q and P with **P**. We omit **P** from the notation whenever the meaning is clear from context.

We define the map ϕ as $\phi(\ell', \delta, \ell) := \pi_{\ell'}(\ell + \delta\eta(\ell))$ where the plane **P** is taken to be the retinal plane $\mathcal{P}$ of the lenslet ℓ' (See also Section 2.2.2). The map ϕ is derived via a similar triangles argument and is explicitly given by[1]

$$\phi(\ell', \delta, \ell) = \frac{d}{\delta\eta(\ell) \cdot \nu}(\ell' - \ell - \delta\eta(\ell)) + \ell', \tag{4.2}$$

[1]Note that equation (4.2) is equivalent to what was previously derived as (2.5) by substituting $Q = \ell + \delta\eta(\ell)$ and $\delta\eta(\ell) \cdot \nu = D + Q^z$.

where d is the distance between the pupilar plane and the retinal plane, see Fig. 2.1. Given that each lenslet has the same limited subimage radius r (cf. Section 2.3), not all lenslets will have a given image point $\iota(P)$ visible in their subimages.

The set $W(\Delta, \ell)$ is the set of lenslets $\ell' \in \mathcal{L}$ for which $||\phi(\ell', \lambda'(\Delta, \ell), \ell) - p_{\ell'}|| < r$, where $||\cdot||$ denotes the Euclidean norm, i.e. the set of lenslets for which the image point $\iota(P) = \ell + \lambda'(\Delta, \ell)\eta(\ell)$ is visible.

4.2.1 Photometric Errors Associated With Distance Maps

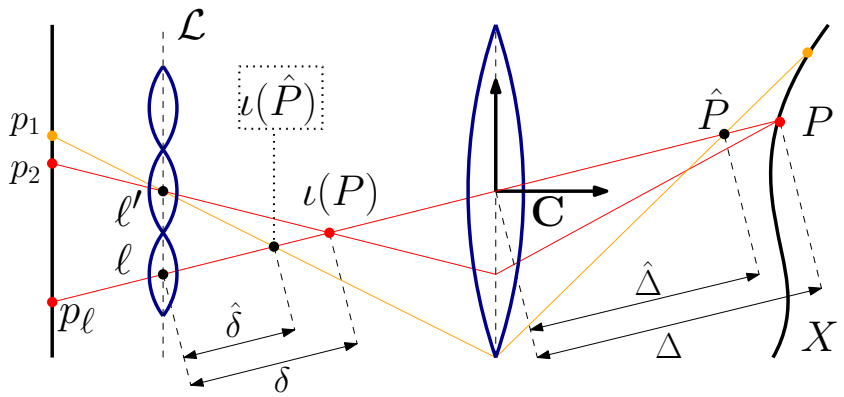


Figure 4.2: A true distance Δ is shown together with an incorrect distance estimate $\hat{\Delta}$. These distances correspond to virtual distances δ and $\hat{\delta}$, respectively. The ray with coordinates (ℓ', p_2) has the same colour as the ray with coordinates (ℓ, p_ℓ), but the ray with coordinates (ℓ', p_1) does not.

Now, we have developed the framework necessary to state the photometric error which will be minimised by the observer. Suppose that the camera is positioned somewhere in the spatial environment Ξ with pose ξ, that the true distance of the scene in direction $\eta(\ell)$ is $\Delta = \lambda(\ell)$, and that we have a distance estimate $\hat{\Delta}$ and the light-field image L.

The ray which passes through both the lenslet ℓ and the point $\Delta \cdot \eta(\ell)$ is the same ray of light which passes through ℓ and $\hat{\Delta} \cdot \eta(\ell)$ for any distance estimate $\hat{\Delta}$. Therefore, if the distance estimate $\hat{\Delta}$ is accurate, we should expect that all other rays passing through the point $\hat{\Delta} \cdot \eta(\ell)$ have the same colour, assuming a Lambertian constraint on the colouring β, see Fig. 4.2.

Therefore, the sum of absolute differences between the colours of all other rays passing through $\hat{\Delta} \cdot \eta(\ell)$ and the *central ray* associated with ℓ – that is the ray passing through both ℓ and the optical centre of the camera – should be minimised by accurate distance estimates.

We define the square of the absolute difference in colour between a central ray of

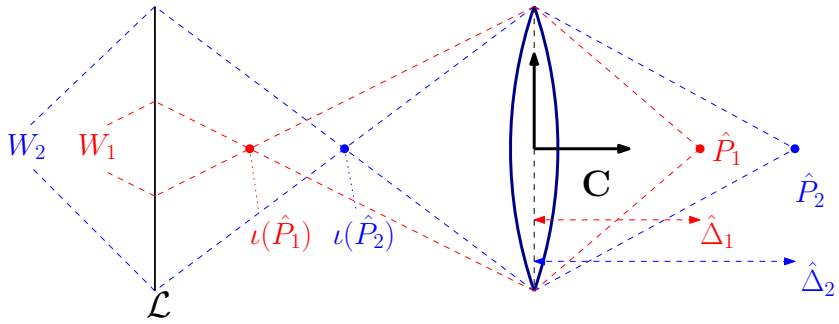


Figure 4.3: The plenoptic discs $W_1 = W(\hat{\Delta}_1, \ell)$ and $W_2 = W(\hat{\Delta}_2, \ell)$ corresponding to distance estimates $\hat{\Delta}_1$ and $\hat{\Delta}_2$ where $\hat{\Delta}_1 < \hat{\Delta}_2$.

a lenslet $\ell \in \mathcal{L}$ and a ray passing through both another lenslet $\ell' \in \mathcal{L}$ and a point estimate $\hat{\Delta} \cdot \eta(\ell)$ as the following *pairwise lenslet error* function e

$$e(\ell', \hat{\Delta}, \ell) := \left|\left|L(\ell, p_\ell) - L(\ell', \phi(\ell', \lambda'(\hat{\Delta}, \ell), \ell))\right|\right|^2,$$

where $||\cdot||$ denotes the Euclidean norm.

Because in practice, a plenoptic camera only has lenslets positioned on a subset $\mathcal{L}^* \subset \mathcal{L}$ that is non-empty, bounded, convex and open relative to $\mathcal{L}$, we will only update depths assigned to lenslets ℓ on this set $\mathcal{L}^*$. However, we will assume that we have light-field information available to us outside of this set in order to ensure differentiability properties of the error function. In practice, this means that for any bounded, convex and relatively open subset of lenslets there is a maximum distance for which we can ensure the local error function defined below is continuously differentiable.

Let $\hat{Q}^z = \iota(\hat{\Delta}\eta(\ell)) \cdot \nu$, be the z-component of the image of a point estimate $\hat{P}$ of distance $\hat{\Delta}$ corresponding to a lenslet ℓ. We propose that given a lenslet ℓ, a distance estimate $\hat{\Delta}$, and a light-field image L, the following local error function ϵ should be minimised by accurate estimates of the distance:

$$\epsilon(\hat{\Delta}, \ell) := \left(1 + \frac{D}{\hat{Q}^z}\right)^{-2} \int_{W(\hat{\Delta}, \ell)} e(\ell', \hat{\Delta}, \ell) d\ell'. \tag{4.3}$$

The purpose of the factor before the integral is to counteract the effect of the varying size of the plenoptic disc $W(\hat{\Delta}, \ell)$ which will otherwise result in smaller errors for smaller distance estimates, regardless of the correctness of these estimates, See Fig. 4.3.

It is the gradient of this error function with respect to estimated depth which will be used to update point estimates.

4.3 Observer Derivation

In this section, we use the error function $\epsilon(\hat{\Delta}, \ell)$ defined in the previous section to derive an observer based on the gradient of this error map. The trajectories of point estimates given by this observer are shown in the appendix to have limit points on the scene X, given some assumptions on the scene X, colouring β, and camera trajectory ξ_t.

Because the scene is stationary in reference frame **O**, it is easiest to express the dynamics of point estimates in this reference frame, as it makes the internal model term trivial, since for points P on the scene $\dot{P}(t) = 0$ in frame **O**. Therefore, the internal model term in the observer will also be trivial for all point estimates.

Because we are now expressing the various maps used in this derivation in frame **O**, we index several of the functions and variables which are dependent on time by t. These include the camera's pose ξ_t expressed in **O**, the pupilar plane $\mathcal{L}_t$ and subset $\mathcal{L}_t^*$ as subsets expressed in **O**, the camera's optical centre $\mathbf{c}_t$, the direction map η_t, and the light-field L_t.

For a given point $\hat{P} \in \mathbb{R}^3$ expressed in the fixed coordinate frame **O**, let $\ell_t = \pi_{\xi_t}(\hat{P})$, then we define[2]

$$v_t(\hat{P}) := \begin{cases} -\nabla_1 \epsilon(\hat{P} \cdot \eta_t(\ell_t), \ell_t)\eta(\ell_t), & \ell_t \in \mathcal{L}_t^*, \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

The observer updates a point estimate with starting position $\hat{P}_0$ according to the time-varying vector field v_t, so that

$$\dot{\hat{P}}_t := v_t(\hat{P}_t). \tag{4.5}$$

The piecewise definition of v_t reflects the fact that we are only updating depths for lenslets $\ell \in \mathcal{L}_t^*$. Note that it is assumed that it is possible to compute the gradient term in (4.4) exactly. This is purely for the sake of theoretical analysis. The effect of numerical error and sensor noise in the method is the topic of future work. Nonetheless, the method is shown to provide accurate point estimates in simulation, despite numerical error in the gradient computations see Section 4.4. A proof of convergence of solutions of (4.5) to the true values is given in Section 4.5.

[2]For a differentiable function $f : \mathbb{R}^n \to \mathbb{R}^m$, we define $\nabla_1 f(x_1, ..., x_n)$ to be the gradient of f with respect to its first argument alone, evaluated at $(x_1, ..., x_n)$.

4.4 Simulation

The observer derived in the previous section was verified in simultation for a simple scenario. In order to do this, synthetic light-field data was generated. In our simulations, light-field data was represented by a large $m \times M$ by $n \times N$ resolution image where $m \times n$ is the resolution of the subimage produced by a single lenslet ℓ, and $M \times N$ is the number of lenslets.

The light-field camera is modelled as a rectangular array of lenslets positioned in front of a rectangular array of pixels. The colour assigned to a pixel p in the subimage of lenslet ℓ is generated using ray-tracing. The pixel location is where the ray passing through p and ℓ is refracted to and can be calculated using Eq. (2.1). The colour assigned to the lenslet-pixel pair (ℓ, p) is then given by the colour $\beta(P)$ of the point P on the 3D scene where the refracted ray corresponding to (ℓ, p) intersects the scene.

In the current implementation, the scene estimates are represented using a point-cloud. Since we are only using a discrete number of lenslets and pixels, an appropriate discretisation of the point-estimate update in (4.5) must be calculated. The choice used in this chapter is as follows. For a given point-estimate $\hat{P}_t$ at time t, the perspective projection $\pi_{\xi_t}(\hat{P}_t)$ of the point-estimate onto the plane of distance D behind the optical centre $\mathbf{c}_t$ is first calculated. We then determine whether $\pi_{\xi_t}(\hat{P}_t)$ lies in $\mathcal{L}_t^*$. If not, it is assigned 0 velocity. Otherwise, if the projection is found to lie within the bounds of $\mathcal{L}_t^*$, we find the nearest lenslet ℓ to $\pi_{\xi_t}(\hat{P}_t)$ and assign to $\hat{P}_t$ the velocity $-\nabla_1 \epsilon(\hat{P}_t \cdot \eta_t(\ell), \ell)\eta_t(\ell)$ in accordance with (4.4). Once all velocities have been assigned to all points, we update the point estimates with these velocities using some positive gain K.

4.4.1 Results

In this simulation, the scene is a sphere and colour was assigned to every point on its surface based as a function of its Euclidean coordinates in $\mathbf{O}$.

The camera followed a path determined by a Lissajous figure and was made to always face outwards from the sphere. This path ensured that each point on the scene is viewed from slightly different perspectives multiple times, which assists with minimising the accumulation of numerical error which may occur from using the same frame multiple times. A practical application that allows essentially free design of camera trajectories is 3D scanning of environments for the purpose of map or model building. In the following simulation, the camera follows such a trajectory lasting 5000 frames.

The initial scene estimate is given by a surface generated from subdividing the

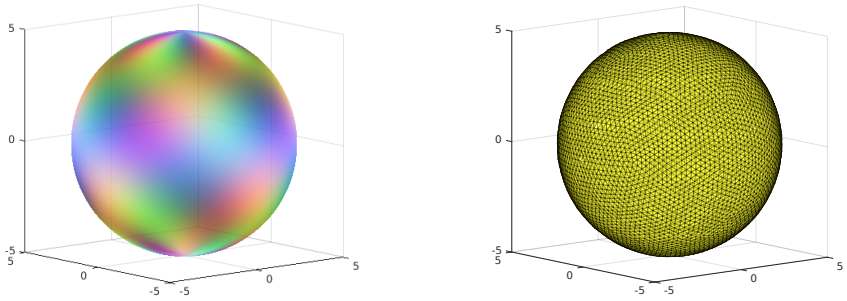


Figure 4.4: Actual scene with colouring (left), and final scene estimate at frame 5000 (right).

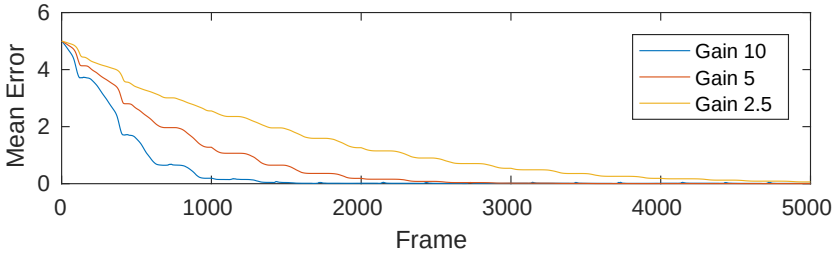


Figure 4.5: Transient response of the average distance of each point estimate from the scene for various gains up to frame 5000.

faces of an icosahedron [121]. The total error graph in Fig. 4.5 shows that with a well chosen gain the observer converges to the scene with a small steady-state error after around 2000 frames, which corresponds to 10–20 iterative updates of each point of the scene. The total error of a scene estimate is given here as the sum of the squares of the distances of each vertex on the scene estimate to the actual scene.

Since the field of view of the camera is small compared to the total area of the scene, a large number of frames are required in order to ensure convergence of the entire scene. A comparison of the scene shown side-by-side with the real scene is given in Fig. 4.4.

Choice of gain and camera trajectories were seen to be important factors when running the proposed algorithm on more challenging scenes. Too large a gain can result in overshoot, causing point estimates to oscillate or diverge, whereas too small a gain results in very slow convergence. A necessary condition for practical convergence of each point estimate to the scene appears to be that each point on the scene is repeatedly updated and repeatedly viewed from different perspectives, including perspectives that increase the visual contrast in a neighbourhood of the point. The first part of this statement is also corroborated by the conditions needed for the con-

vergence proof in Appendix 4.5, cf. Assumption 9.

4.5 Theoretical Analysis

In this section, we prove the convergence behaviour of the derived observer. We start with the assumptions required to do so.

Assumptions

The following list of assumptions are needed in the subsequent proof in order to ensure asymptotic convergence of a point estimate defined by (4.5) to the actual scene X. However, this does not mean that the listed assumptions are the weakest possible to ensure asymptotic convergence.

In order to avoid unnecessary discussions of the subtleties of solution concepts for differential equations with discontinuous right hand side [24], we assume existence and uniqueness of absolutely continuous solutions of (4.5) for all initial conditions. This will be the case for reasonable camera trajectories.

We denote the topological closure of a set $S \in \mathbb{R}^3$ by $\mathrm{cl}(S)$.

Definition 8. *The set $C^+(B, \hat{P})$ is the positive half-cone with apex $\hat{P} \in \mathbb{R}^3$ spanned by the bounded convex set $B \subset \mathbb{R}^3$, where $\hat{P} \notin B$, see Fig. 4.6. Formally, it is the set of $\hat{P}' \in \mathbb{R}^3$ for which there exists a point $\mathbf{c}' \in B$ and an $\alpha > 0$ such that $\hat{P}' - \hat{P} = \alpha(\hat{P} - \mathbf{c}')$. The set $C^+(B, \hat{P})$ is open whenever B is, does not contain the apex $\hat{P}$, and extends to infinity. We denote $C_0^+(B, \hat{P}) = C^+(B, \hat{P}) \cup \{\hat{P}\}$. The negative half-cone, $C^-(B, \hat{P})$, is defined as the set of $\hat{P}' \in \mathbb{R}^3 \setminus cl(B)$ for which there exists a point $\mathbf{c}' \in B$ and an $0 < \alpha < 1$ such that $\hat{P}' - \hat{P} = -\alpha(\hat{P} - \mathbf{c}')$. The set $C^-(B, \hat{P})$ is open whenever B is, does not contain the apex $\hat{P}$, is bounded and sits atop the base B.*

The following constant defines the minimum depth a point has if the image of that point lies between the focal lens and the pupilar plane:

$$\Delta_{\min} := \frac{1}{\inf_{\ell \in \mathcal{L}^*}(\eta(\ell) \cdot \nu)} \max\left(F, \frac{DF}{F - D}\right).$$

Assumption 6. *ξ_t is continuous in t and there exists an open ball $\mathbf{B} \subset \Xi$ centred at 0 in reference frame $\mathbf{O}$ such that both the optical centre $\mathbf{c}_t$ and the bounded cone $\{Q \in C^+(\mathcal{L}_t^*, \mathbf{c}_t) \mid Q \cdot \nu_t \leq \Delta_{min}\}$ are contained within $\mathbf{B}$ for all $t \geq 0$.*

This assumption ensures that the camera moves in a continuous fashion and never gets too close to the scene, allowing us to pick initial conditions of at least distance $\Delta_{\min}$ away from the focal lens of the camera.

Assumption 7. *Let P, x_1, $x_2 \in X$. If*

$$||x_1 - P|| > ||x_2 - P||$$

then

$$||\beta(x_1) - \beta(P)|| > ||\beta(x_2) - \beta(P)||$$

.

This assumption states that the colouring is monotonic. This is one assumption which may potentially be weakened in future work.

Assumption 8. *The scene X is a convex surface.*

This assumption may be weakened in future work to the scene being a star-shaped surface[3] with respect to **B** from Assumption 6.

It is convenient in the following proof to define the set of times for which a given point estimate $\hat{P}_t$ is seen by the camera.

Definition 9. *Given an initial condition $\hat{P}_0$ of the system* (4.5)*, define $T(\hat{P}_0)$ to be the set of times $t > 0$ for which $\pi_{\xi_t}(\hat{P}_t) \in \mathcal{L}_t^*$ and $\hat{P}_t \cdot \nu_t > 0$.*

Note that $t \in T(\hat{P}_0)$ implies that $\hat{P}_t \in C^+(\mathcal{L}_t^*, \mathbf{c}_t)$ and $\dot{\hat{P}}_t = -\nabla_1 \epsilon(\hat{P}_t \cdot \eta_t(\ell_t), \ell_t)\eta_t(\ell_t)$, where $\ell_t = \pi_{\xi_t}(\hat{P}_t) \in \mathcal{L}_t^*$.

Lastly, we wish to ensure that there is always a future interval of time for which a given point estimate, and a neighbourhood around it, will be seen by the camera. Let $B_r(P) \subset \mathbb{R}^3$ denote the open ball of radius $r > 0$ centred at $P \in \mathbb{R}^3$.

Assumption 9. *There exists a $\rho > 0$ and a $\Delta t > 0$ such that for a given initial condition $\hat{P}_0$, and all times $t > 0$ there exists a $t^+ > t$ such that $\pi_{\xi_s}(cl(B_\rho(\hat{P}_s))) \subset \mathcal{L}_s^*$ and $\hat{P}' \cdot \nu_s > 0$ for all $\hat{P}' \in cl(B_\rho(\hat{P}_s))$ and for all $s \in [t^+, t^+ + \Delta t]$. In particular, $[t^+, t^+ + \Delta t] \subset T(\hat{P}_0)$.*

Cone Geometry

Proposition 8. *Let B be a bounded convex set and $\hat{P} \notin cl(B)$. Then $\hat{P}' \in C^+(B, \hat{P})$ if and only if $\hat{P} \in C^-(B, \hat{P}')$.*

Proof. If $\hat{P}' \in C^+(B, \hat{P})$ then there exists an $\mathbf{c}' \in B$ and an $\alpha > 0$ such that $\hat{P}' - \hat{P} = \alpha(\hat{P} - \mathbf{c}')$ which implies $\hat{P} - \hat{P}' = \frac{-\alpha}{1+\alpha}(\hat{P}' - \mathbf{c}')$ and hence $\hat{P} \in C^-(B, \hat{P}')$. Conversely, if $\hat{P} \in C^-(B, \hat{P}')$ then there exists an $\mathbf{c}' \in B$ and an $0 < \alpha < 1$ such that $\hat{P} - \hat{P}' = -\alpha(\hat{P}' - \mathbf{c}')$ which implies $\hat{P}' - \hat{P} = \frac{\alpha}{1-\alpha}(\hat{P} - \mathbf{c}')$ and hence $\hat{P}' \in C^+(B, \hat{P})$. □

[3] By a convex or star-shaped *surface* we mean that the set is the boundary of a convex or star-shaped set respectively. Remember that a set X is called star-shaped with respect to a point Q if for every point $P \in \partial X$, the ray segment extending from Q though P only intersects ∂X at P. In contrast, a convex set X is one that is star-shaped with respect to *every* point Q in X.

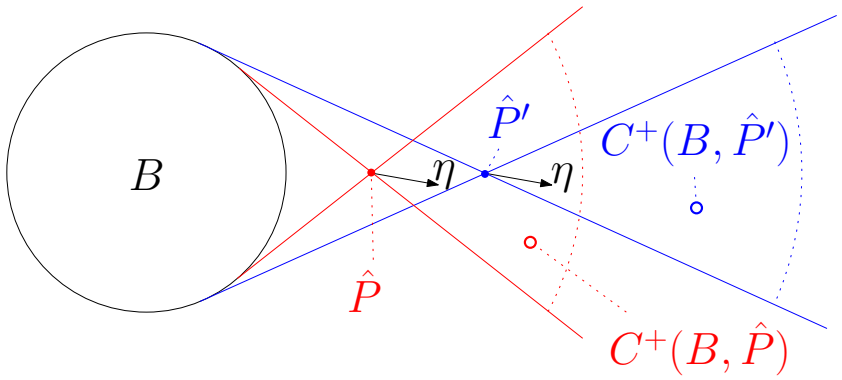


Figure 4.6: A Planar cut through B that contains both $\hat{P}$ and $\hat{P}'$.

Proposition 9. *Let B be an open ball. If $\hat{P}' \in C^+(B,\hat{P})$ then $cl(C^+(B,\hat{P}')) \subset C^+(B,\hat{P})$. Furthermore, if $\hat{P}' \in C^+(B,\hat{P})$ and $\hat{P}'+\eta \in C^+(B,\hat{P}')$ then $\hat{P}+\eta \in C^+(B,\hat{P})$. If $\hat{P}' \in C_0^+(B,\hat{P})$ then $C^+(B,\hat{P}') \subset C^+(B,\hat{P})$.*

Sketch of Proof. Picture a planar cut through B that contains both $\hat{P}$ and $\hat{P}'$ (see Fig. 4.6) and note that $C^+(B,\hat{P})$ is on the opposite site of $\hat{P}$ to B. Since $\hat{P}'$ is inside the open cone $C^+(B,\hat{P})$, the opening angles of $C^+(B,\hat{P}')$ are strictly smaller than those of $C^+(B,\hat{P})$ and the first result follows. Translating the cone $C^+(B,\hat{P}')$ to $C^+(B,\hat{P}') - \hat{P}' + \hat{P}$ results in a cone with apex $\hat{P}$ which has smaller opening angles than $C^+(B,\hat{P})$ and is therefore a subset of it, giving the second result. The third result follows from the first observing that $C^+(B,\hat{P}') = C^+(B,\hat{P})$ if $\hat{P}' = \hat{P}$. □

Proposition 10. *Let B be an open ball. If $\hat{P} \in C^-(B,\hat{P}')$ then $C^-(B,\hat{P}) \subset C^-(B,\hat{P}')$.*

Sketch of Proof. Picture a planar cut through B that contains both $\hat{P}$ and $\hat{P}'$ (see Fig. 4.6) and note that both $C^-(B,\hat{P})$ and $C^-(B,\hat{P}')$ are bounded by the spherical base B. Since $\hat{P}$ is inside the open cone $\hat{P} \in C^-(B,\hat{P}')$, the opening angles of $C^-(B,\hat{P})$ are strictly larger than those of $C^-(B,\hat{P}')$ and hence the cone $C^-(B,\hat{P})$ touches the spherical base inside $C^-(B,\hat{P}')$. The result follows. □

Proposition 11. *Suppose $0 \notin B_r(\mathbf{c})$. There exists a $c \in (0,1)$ such that $C^+(B_r(\mathbf{c}),0) = \{\hat{P} \in \mathbb{R}^3 : -\hat{P}\cdot\mathbf{c} > c\,||\hat{P}||\,||\mathbf{c}||\}$, see Fig 4.7.*

Proof. If $\hat{P} \in C^+(B_r(\mathbf{c}),0)$ then $\hat{P} \neq 0$ because by definition $C^+(B_r(\mathbf{c}),0)$ is open and does not contain its apex. Hence the statement that $\hat{P} \in C^+(B_r(\mathbf{c}),0)$ is equivalent to stating the existence of a line segment passing from $\hat{P}$ through 0 which intersects $B_r(\mathbf{c})$. This is equivalent to stating that $\left|\frac{\hat{P}}{||\hat{P}||}\cdot\mathbf{c}\right|^2 - ||\mathbf{c}||^2 + r^2 > 0$, and so $\frac{|\hat{P}\cdot\mathbf{c}|^2}{||P||^2||\mathbf{c}||^2} > 1 - \frac{r^2}{||\mathbf{c}||^2}$. Letting $c^2 = 1 - \frac{r^2}{||\mathbf{c}||^2}$, noting that $r < ||\mathbf{c}||$ and observing that by definition $-\hat{P}\cdot\mathbf{c} > 0$, the conclusion follows. □

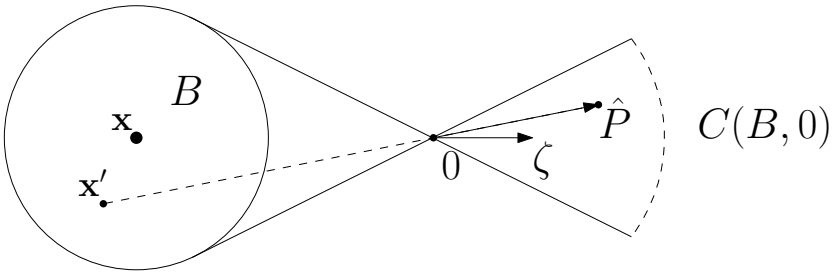


Figure 4.7: A cone generated by $B_r(\mathbf{c})$ through 0. There is a scalar $1 > c > 0$ and a unit vector ζ through the centre axis of the cone for which the dot product of any $\hat{P} \in C^+(B, 0)$ with ζ is at least $c\left|\left|\hat{P}\right|\right|$.

Proposition 12. *Let C be a right-angled cone with base radius b and height h. Let x be the apex of the cone. Then $C \subset B_\rho(x)$ where $\rho = 2\sqrt{b^2 + h^2}$.*

Sketch of proof. This follows from taking a planar cut of the cone containing its central axis, resulting in an isosceles triangle, and representing points in this triangle as a convex sum of the corners. □

Error Function

In the following, we prove that for each lenslet ℓ, the local error function $\epsilon(\hat{\Delta}, \ell)$ defined by (4.3) has a unique minimum at $\hat{\Delta} = \Delta$, where Δ is the true distance of the scene in direction $\eta(\ell)$ and the first argument of ϵ is restricted to $(\Delta_{\min}, \infty)$.

Lemma 13. *Let ϵ be the error function defined by* (4.3)*. Let Δ be the true distance of the scene X in direction $\eta(\ell)$. Then $\epsilon(\Delta, \ell) = 0$ and if $\Delta_{min} < \hat{\Delta}_1 < \hat{\Delta}_2 < \Delta$ or $\Delta_{min} < \Delta < \hat{\Delta}_2 < \hat{\Delta}_1$, we have that $\epsilon(\hat{\Delta}_1, \ell) > \epsilon(\hat{\Delta}_2, \ell) > 0$.*

Proof. Denote $P = \Delta\eta(\ell)$ and $Q = \iota(P)$, see Fig. 4.2. Firstly, if $\hat{\Delta} = \Delta$, then transforming the integral in (4.3) through the inverse projection map transforms the plenoptic disc $W(\hat{\Delta}, \ell)$ to a single point $\hat{P} = P$ on the scene, and so the error is 0 in this case.

Let $\mathcal{A}$ denote the focus lens, which is a disc of radius A (where A is the aperture) normal to ν. If $\hat{Q}$ is an image point estimate, and $a \in \mathcal{A}$, then let $\pi_{\hat{Q}}(a)$ denote the perspective projection of the point a through $\hat{Q}$ onto the pupilar plane $\mathcal{L}$.

Note that $\pi_{\hat{Q}}(a) = a + \frac{D}{\hat{Q}^z}\left(a - \hat{Q}\right)$, where $\hat{Q}^z = \hat{Q} \cdot \nu$. Therefore, $\left|\det D\pi_{\hat{Q}}(a)\right|$ only depends on $\hat{Q}$ and is given by $\left|\det D\pi_{\hat{Q}}(a)\right| = \left(1 + \frac{D}{\hat{Q}^z}\right)^2$.

Now, consider $\epsilon(\hat{\Delta}_1, \ell) - \epsilon(\hat{\Delta}_2, \ell)$, and note that in either case we have that $\left|\hat{\Delta}_1 - \Delta\right| >$

$\left|\hat{\Delta}_2 - \Delta\right|$. Then we have that

$$\begin{aligned}
&\epsilon(\hat{\Delta}_1, \ell) - \epsilon(\hat{\Delta}_2, \ell) \\
&= \int_{W(\hat{\Delta}_1, \ell)} e(\ell', \hat{\Delta}_1, \ell) \left(1 + \frac{D}{\hat{Q}_1^z}\right)^{-2} d\ell' \\
&\quad - \int_{W(\hat{\Delta}_2, \ell)} e(\ell', \hat{\Delta}_2, \ell) \left(1 + \frac{D}{\hat{Q}_2^z}\right)^{-2} d\ell' \\
&= \int_{\mathcal{A}} \left|\left|\beta(P) - \beta(\pi_{\hat{P}_1}^{-1}(a))\right|\right|^2 da \\
&\quad - \int_{\mathcal{A}} \left|\left|\beta(P) - \beta(\pi_{\hat{P}_2}^{-1}(a))\right|\right|^2 da \\
&> 0.
\end{aligned}$$

Here we have used Assumptions 7 and 8 and the fact that if the scene is convex then the further a point estimate $\hat{P}$ is from the scene, the further the projection of a point on the focus lens through $\hat{P}$ will be from the true point P. □

Point Trajectories

The first observation is that if $\hat{P} \in X$ is a point on the scene then $v_\tau(\hat{P}) = 0$ for all t by Lemma 13. This means that $\hat{P}_t = \hat{P}$ for all t is a trajectory of (4.5), and hence $\hat{P}_t = \hat{P}$ for *some* t implies $\hat{P}_t = \hat{P}$ for *all* t because solutions of (4.5) are assumed to be unique.

The following result additionally states that if the point estimate lies in Ξ for some time t, it stays in Ξ for all future times, and if it lies in $\Xi^c = \mathbb{R}^3 \setminus \mathrm{cl}(\Xi)$ it stays there.

Proposition 14. *If $\hat{P}_t \in X$ then $\hat{P}_\tau \in X$ for all τ. If $\hat{P}_t \in \Xi$ then $\hat{P}_\tau \in \Xi$ for all $\tau \geq t$. If $\hat{P}_t \in \Xi^c$ then $\hat{P}_\tau \in \Xi^c$ for all $\tau \geq t$.*

Proof. We have already shown the first statement at the beginning of Section 4.5. Assume $\hat{P}_t \in \Xi$ and assume for a contradiction that $\hat{P}_\tau \notin \Xi$ for some $\tau > t$. Because $\hat{P}$ as defined by (4.5) is continuous, there exists an $s \in [t, \tau]$ such that $\hat{P}_s \in X = \partial\Xi$. By the first statement it follows that $\hat{P}_{s'} \in X$ for all s', a contradiction to $\hat{P}_t \in \Xi$. The case $\hat{P}_t \in \Xi^c$ follows from a similar argument. □

The goal of the remainder of this section is to establish that if a point estimate $\hat{P}$ with initial condition $\hat{P}_0 \in \Xi$ has a limit point Q, then that limit point cannot be in $C^+(\mathbf{B}, \hat{P}_0) \cap \Xi$. A similar statement holds for the case where $\hat{P}_0 \in \Xi^c$ with the obvious modifications to all the intermediate statements and proofs.

We separate the following into three subsections. In the first subsection, we investigate general properties which must be true of any solution of (4.5) with $\hat{P}_0 \in \Xi$. In the second subsection, we show that every accumulation point of the trajectory $\hat{P}$ is a limit point. In the third subsection, we establish that the assumption that the limit point of the trajectory $\hat{P}$ is in $C^+(\mathbf{B}, \hat{P}_0) \cap \Xi$ results in a contradiction.

Properties of Point Estimates

We begin by investigating the time set $T(\hat{P}_0)$ from Definition 9.

Proposition 15. *$T(\hat{P}_0)$ is open.*

Proof. Let $t > 0$ and express the point estimate $\hat{P}_t$ in frame $\mathbf{C}$ as ${}^{\mathbf{C}}\hat{P}_t$. Let ${}^{\mathbf{C}}\pi_0$ be the perspective projection of points in front of the camera through the optical centre expressed in frame $\mathbf{C}$ (in which it has constant coordinates 0) onto the pupilar plane ${}^{\mathbf{C}}\mathcal{L}$ which is constant in the frame $\mathbf{C}$, as is ${}^{\mathbf{C}}\mathcal{L}^*$. Then, ${}^{\mathbf{C}}\hat{P}_t = \xi_t^{-1}\hat{P}_t$, which is continuous with respect to t since $\hat{P}_t$ and ξ_t are, the latter by Assumption 6. Since ${}^{\mathbf{C}}\pi_0$ is continuous, ${}^{\mathbf{C}}\pi_0({}^{\mathbf{C}}\hat{P}_t)$ is continuous with respect to t, and if ${}^{\mathbf{C}}\pi_0({}^{\mathbf{C}}\hat{P}_t) \in {}^{\mathbf{C}}\mathcal{L}^*$, there is a time interval (a_t, b_t) containing t such that ${}^{\mathbf{C}}\pi_0({}^{\mathbf{C}}\hat{P}_\tau) \in {}^{\mathbf{C}}\mathcal{L}^*$ for all $\tau \in (a_t, b_t)$. Now, $T(\hat{P}_0) = \bigcup_{t \in T(\hat{P}_0)} (a_t, b_t)$ which is open. □

The following proposition shows that for $t \in T(\hat{P}_0)$ the vector field in (4.5) points into the interior of a cone with apex $\hat{P}_t$ spanned by the ball $\mathbf{B}$ from Assumption 6.

Proposition 16. *Let $t \in T(\hat{P}_0)$ and $\hat{P}_t \in \Xi$ and $\hat{P}_t \in \mathbf{B}$, where $\mathbf{B}$ is from Assumption 6. Then $\hat{P}_t + \dot{\hat{P}}_t \in C^+(\mathbf{B}, \hat{P}_t)$.*

Proof. Let $\ell_t = \pi_{\xi_t}(\hat{P}_t)$ then $\dot{\hat{P}}_t = -\nabla_1 \epsilon(\hat{P}_t \cdot \eta_t(\ell_t), \ell_t)\eta_t(\ell_t)$ and $\nabla_1 \epsilon(\hat{P}_t \cdot \eta_t(\ell_t), \ell_t) < 0$ by Lemma 13. Therefore, $\dot{\hat{P}}_t$ is a positive multiple of $\eta_t(\ell_t)$ in this case and $\hat{P}_t + \eta_t(\ell_t) \in C^+(\mathbf{B}, \hat{P}_t)$ implies $\hat{P}_t + h\dot{\hat{P}}_t \in C^+(\mathbf{B}, \hat{P}_t)$ for all $h > 0$ as $C^+(\mathbf{B}, \hat{P}_t)$ is a cone. □

The following proposition gives the existence of some time interval $(t, t + \epsilon)$ for which the trajectory of a point estimate then remains within the cone $C^+(\mathbf{B}, \hat{P}_t)$ for all times within the time interval $(t, t + \epsilon)$. This is important for establishing the existence of a limit point for the trajectory.

Proposition 17. *Let $t \in T(\hat{P}_0)$ and $\hat{P}_t \in \Xi$ and $\hat{P}_t \notin \mathbf{B}$, where $\mathbf{B}$ is from Assumption 6. Then there exists an $\epsilon > 0$ such that $\hat{P}_{t+h} \in C^+(\mathbf{B}, \hat{P}_t) \cap \Xi$ for all $0 < h < \epsilon$.*

Proof. By Prop. 16, $\hat{P}_t + \dot{\hat{P}}_t \in C^+(\mathbf{B}, \hat{P}_t)$. Since $C^+(B, \hat{P}_t)$ is open, there exists a $\delta > 0$ such that $B_\delta(\hat{P}_t + \dot{\hat{P}}_t) \subset C^+(\mathbf{B}, \hat{P}_t)$. But then $B_{\delta h}(\hat{P}_t + h\dot{\hat{P}}_t) \subset C^+(\mathbf{B}, \hat{P}_t)$ for all $h > 0$ since $C^+(\mathbf{B}, \hat{P}_t)$ is a cone.

As t is in $T(\hat{P}_0)$ and $T(\hat{P}_0)$ is open by Prop. 15, we have $\dot{\hat{P}}_t = \lim_{h\to 0} \frac{\hat{P}_{t+h} - \hat{P}_t}{h}$. Hence there exists an $\epsilon > 0$ such that for all $0 < h < \epsilon$, we have $\left|\left|\hat{P}_{t+h} - (\hat{P}_t + h\dot{\hat{P}}_t)\right|\right| < \delta h$. It follows that $\hat{P}_{t+h} \in C^+(\mathbf{B}, \hat{P}_t)$ and by Prop. 14 also $\hat{P}_{t+h} \in \Xi$ for all $0 < h < \epsilon$. □

The following proposition uses the previous proposition to produce a stronger result: that for every time $t \in T(\hat{P}_0)$ and every time $\tau > t$, the point estimate $\hat{P}_\tau$ is contained in the cone $C^+(\mathbf{B}, \hat{P}_t)$.

Proposition 18. *Let $t \in T(\hat{P}_0)$ and $\hat{P}_t \in \Xi$ and $\hat{P}_t \notin \mathbf{B}$ where $\mathbf{B}$ is from Assumption 6. Then $\hat{P}_\tau \in C^+(\mathbf{B}, \hat{P}_t) \cap \Xi$ for all $\tau > t$.*

Proof. Assume, to arrive at a contradiction, that there exists $\tau > t$ with $\hat{P}_\tau \notin C^+(\mathbf{B}, \hat{P}_t) \cap \Xi$. By Proposition 17, $\hat{P}_{t+h} \in C^+(\mathbf{B}, \hat{P}_t) \cap \Xi$ for $h > 0$ sufficiently small. Since $\hat{P}$ is continuous in t, there is a smallest time $b \in (t, \tau)$ such that $\hat{P}_b \in \partial(C^+(\mathbf{B}, \hat{P}_t) \cap \Xi)$ and $\hat{P}_s \in C^+(\mathbf{B}, \hat{P}_t) \cap \Xi$ for all $s \in (t, b)$. By Prop. 14, $\hat{P}_b \in \Xi$ and hence $\hat{P}_b \in \partial(C^+(\mathbf{B}, \hat{P}_t) \cap \Xi) \cap \Xi = \partial C^+(\mathbf{B}, \hat{P}_t) \cap \Xi$. In particular, $\hat{P}_b \notin C^+(\mathbf{B}, \hat{P}_t)$.

If $b \in T(\hat{P}_0)$ then there exists a nonempty open interval $(a, b) \subset (t, b)$ such that $(a, b) \subset T(\hat{P}_0)$ as $T(\hat{P}_0)$ is open by Prop. 15. If $b \notin T(\hat{P}_0)$ then $s \notin T(\hat{P}_0)$ and therefore $\dot{\hat{P}}_s = 0$ for all $s \in [b', b]$, where $b' = \sup\{s \in T(\hat{P}_0) \,|\, s < b\}$, and there exists a nonempty open interval $(a, b') \subset (t, b')$ such that $(a, b') \subset T(\hat{P}_0)$. But then $\hat{P}_{b'} = \hat{P}_b \in \partial(C^+(\mathbf{B}, \hat{P}_t) \cap \Xi)$ and $b' = b$ as b was minimal. It follows that there exists a nonempty open interval $(a, b) \subset (t, b)$ such that $(a, b) \subset T(\hat{P}_0)$ also in this case.

In both cases we then have that there exists a nonempty open interval $(a, b) \subset T(\hat{P}_0)$ such that $\hat{P}_s \in C^+(\mathbf{B}, \hat{P}_t) \cap \Xi$ for all $s \in (a, b)$. By Prop 16, it follows that $\hat{P}_s + \dot{\hat{P}}_s \in C^+(\mathbf{B}, \hat{P}_s)$ for all $s \in (a, b)$, and by Prop. 9, $\hat{P}_t + \dot{\hat{P}}_s \in C^+(\mathbf{B}, \hat{P}_t)$ for all $s \in (a, b)$. Recall that $\hat{P}_b \notin C^+(\mathbf{B}, \hat{P}_t)$.

For the remainder of the argument we change coordinates such that $\hat{P}_t = 0$. This is so we can apply Proposition 11. In the new coordinates $||\mathbf{c}|| > r > 0$, where $\mathbf{c}$ is the centre of the ball $\mathbf{B}$ of radius r, by our assumption that $\hat{P}_t \notin \mathbf{B}$. We now have $\hat{P}_a \in C^+(\mathbf{B}, 0)$ and $\dot{\hat{P}}_s \in C^+(\mathbf{B}, 0)$ for all $s \in (a, b)$ but $\hat{P}_b \notin C^+(\mathbf{B}, 0)$. Because $\hat{P}$ is

absolutely continuous on the interval $[a, b]$ we have:

$$\begin{aligned}
-\hat{P}_b \cdot \mathbf{c} = & -\hat{P}_a \cdot \mathbf{c} + \int_a^b -\dot{\hat{P}}_s \cdot \mathbf{c}\, ds \\
& > c\,||\mathbf{c}||\,\left|\left|\hat{P}_a\right|\right| + \int_a^b c\,||\mathbf{c}||\,\left|\left|\dot{\hat{P}}_s\right|\right|\, ds \\
& \geq c\,||\mathbf{c}||\,\left|\left|\hat{P}_a\right|\right| + c\,||\mathbf{c}||\,\left|\left|\int_a^b \dot{\hat{P}}_s\, ds\right|\right| \\
& = c\,||\mathbf{c}||\,\left|\left|\hat{P}_a\right|\right| + c\,||\mathbf{c}||\,\left|\left|\hat{P}_b - \hat{P}_a\right|\right| \\
& \geq c\,||\mathbf{c}||\,\left|\left|\hat{P}_b\right|\right|
\end{aligned}$$

which implies $\hat{P}_b \in C^+(\mathbf{B}, 0)$ by Proposition 11 (note the $>$ sign on the second line). This is a contradiction to $\hat{P}_b \notin C^+(\mathbf{B}, 0)$ and it follows that $\hat{P}_\tau \in C^+(\mathbf{B}, \hat{P}_t) \cap \Xi$ for all $\tau > t$. □

The following two results are the main results of this subsection.

Proposition 19. *Let $\hat{P}_t \in \Xi$ and $\hat{P}_t \notin cl(\mathbf{B})$ where $\mathbf{B}$ is from Assumption 6. Then $\hat{P}_\tau \in C_0^+(\mathbf{B}, \hat{P}_t) \cap \Xi$ and $C^+(\mathbf{B}, \hat{P}_\tau) \subset C^+(\mathbf{B}, \hat{P}_t)$ for all $\tau \geq t$.*

Proof. Clearly $\hat{P}_\tau \in C_0^+(\mathbf{B}, \hat{P}_t)$ implies $C^+(\mathbf{B}, \hat{P}_\tau) \subset C^+(\mathbf{B}, \hat{P}_t)$ by Prop. 9, and $\hat{P}_\tau \in \Xi$ for all $\tau \geq t$ by Prop. 14. Hence we only need to prove $\hat{P}_\tau \in C_0^+(\mathbf{B}, \hat{P}_t)$ for all $\tau \geq t$. The case $\tau = t$ is immediate, so let $\tau > t$ for the remainder of the proof. Let $t \in T(\hat{P}_0)$ then the statement follows from Prop. 18. Let $t \notin T(\hat{P}_0)$ then $s \notin T(\hat{P}_0)$ and therefore $\dot{\hat{P}}_s = 0$ for all $s \in [t, t']$, where $t' = \inf\{s \in T(\hat{P}_0) \,|\, s > t\}$. Note that t' is finite by Assumption 9. It follows that $\hat{P}_s = \hat{P}_t$ for all $s \in [t, t']$ and there exists a nonempty open interval $(t', b) \subset T(\hat{P}_0)$. The case $\tau \leq t'$ is now immediate, so assume $\tau > t'$ for the remainder of the proof.

Recall $\hat{P}_{t'} = \hat{P}_t \notin \mathrm{cl}(\mathbf{B})$. Since $\hat{P}$ is continuous, there exists $b' \in (t', b)$ such that $\hat{P}_s \notin \mathrm{cl}(\mathbf{B})$ for all $s \in (t', b')$. Now, we have two cases: either $\tau \in (t', b')$ or $t \notin (t', b')$.

Assume $\tau \in (t', b')$ for now, recall that $\hat{P}'_t = \hat{P}_t$ and assume for a contradiction that $\hat{P}_\tau \notin C_0^+(B, \hat{P}_{t'})$. Then $\hat{P}_{t'} \notin C^-(\mathbf{B}, \hat{P}_\tau)$ by Prop. 8. Furthermore, $\hat{P}_\tau \in C^+(\mathbf{B}, \hat{P}_s)$ for all $s \in (t', \tau)$ by Prop. 18, and hence $\hat{P}_s \in C^-(\mathbf{B}, \hat{P}_\tau)$ for all $s \in (t', \tau)$ by Prop. 8. Since $\hat{P}_s$ is inside the open cone $C^-(\mathbf{B}, \hat{P}_\tau)$ and $\hat{P}_{t'} \notin C^-(\mathbf{B}, \hat{P}_\tau)$ and $\hat{P}_{t'} \notin \mathrm{cl}(\mathbf{B})$, it follows that there exists $\delta > 0$ such that $\left|\left|\hat{P}' - \hat{P}_{t'}\right|\right| \geq \delta$ for all $\hat{P}' \in C^-(\mathbf{B}, \hat{P}_s)$.

Repeating the argument, by Prop 18, $\hat{P}_s \in C^+(\mathbf{B}, \hat{P}_{s'})$ for all $s' \in (t', s)$, and hence $\hat{P}_{s'} \in C^-(\mathbf{B}, \hat{P}_s)$ for all $s' \in (t', s)$ by Prop. 8. This implies $\left|\left|\hat{P}_{s'} - \hat{P}_{t'}\right|\right| \geq \delta$ for all $s' \in (t', s)$ and hence $\lim_{s' \to t'} \left|\left|\hat{P}_{s'} - \hat{P}_{t'}\right|\right| \geq \delta$, which contradicts continuity of $\hat{P}$ at t'. Therefore, if $\tau \in (t', b')$ then $\hat{P}_\tau \in C_0^+(\mathbf{B}, \hat{P}_{t'}) = C_0^+(\mathbf{B}, \hat{P}_t)$.

For the second case, if $\tau \notin (t', b')$, then we take any $\tau' \in (t', b')$, and conclude using the previous argument that $\hat{P}_{\tau'} \in C_0^+(\mathbf{B}, \hat{P}_t)$. Using Prop. 18 we have that $\hat{P}_s \in C^+(\mathbf{B}, \hat{P}_{\tau'})$ for all $s > \tau'$, and this latter set is contained in $C^+(\mathbf{B}, \hat{P}_t)$ by Prop. 9, and therefore $\hat{P}_\tau \in C_0^+(\mathbf{B}, \hat{P}_t)$ also in this case. □

Proposition 20. *Let $\hat{P}_t \in \Xi$ and $\hat{P}_t \notin \mathbf{B}$ where $\mathbf{B}$ is from Assumption 6. Then there exists a $t^+ > t$ such that $\hat{P}_\tau \in C^+(\mathbf{B}, \hat{P}_t) \cap \Xi$ for all $\tau > t^+$.*

Proof. By Assumption 9 there exists a $t^+ > t$ such that $t^+ \in T(\hat{P}_0)$. By Prop. 19, $\hat{P}_{t^+} \in C_0^+(\mathbf{B}, \hat{P}_t) \cap \Xi$ and hence $\hat{P}_{t^+} \in \Xi$ and $\hat{P}_{t^+} \notin \mathbf{B}$. By Prop. 18, $\hat{P}_\tau \in C^+(\mathbf{B}, \hat{P}_{t^+}) \cap \Xi$ for all $\tau > t^+$. By Prop. 19, $C^+(\mathbf{B}, \hat{P}_{t^+}) \subset C^+(\mathbf{B}, \hat{P}_t)$ and the result follows. □

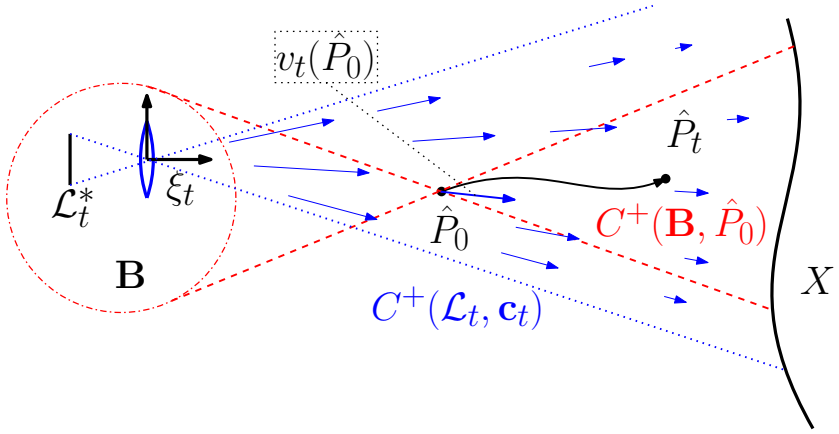


Figure 4.8: A initial point estimate $\hat{P}_0 \in \Xi$, $\hat{P}_0 \notin \mathbf{B}$ has its trajectory $\hat{P}$ contained in the pointed cone $C_0^+(\mathbf{B}, \hat{P}_0)$. The observer produces a vector field v_t which always points away from the optical centre of the camera. The set of points for which the vector field can be non-zero is the cone $C^+(\mathcal{L}_t^*, \mathbf{c}_t)$, where $\mathbf{c}_t$ is the optical centre.

We have now established that if $\hat{P}_0 \in \Xi$ and $\hat{P}_0 \notin \mathbf{B}$ then $\hat{P}_t \in C_0^+(\mathbf{B}, \hat{P}_0) \cap \Xi$ for all $t \geq 0$, see Fig. 4.8. Since the trajectory $\hat{P}$ is contained in a bounded set, it is a simple consequence of the Bolanzo-Weierstrass theorem that the trajectory has an accumulation point in the closure of that set.

Accumulation points are limit points

The following two propositions establish that any accumulation point[4] of the trajectory $\hat{P}$ must be a limit point.

[4]To avoid confusion with differing conventions, in this work we explicitly define the convention that for the trajectory $\hat{P} : \mathbb{R} \to \mathbb{R}^3$, a point Q is an *accumulation point* if for every $\delta > 0$ and every $t^+ > 0$ there exists a $\tau > t^+$ such that $||\hat{P}_\tau - Q|| < \delta$. This is in contrast with a *limit point* Q that must satisfy for every $\delta > 0$ *there exists* a $t^+ > 0$ such that *for all* $\tau > t^+$ we have that $||\hat{P}_\tau - Q|| < \delta$.

Proposition 21. *Let $\hat{P}_0 \in \Xi$ and $\hat{P}_0 \notin \mathbf{B}$ where $\mathbf{B}$ is from Assumption 6. If Q is an accumulation point of the trajectory $\hat{P}$ then $Q \in C^+(\mathbf{B}, \hat{P}_t)$ for all $t \geq 0$.*

Proof. Suppose for a contradiction that there were a $t \geq 0$ such that $Q \notin C^+(\mathbf{B}, \hat{P}_t)$. By Prop 20 there exists a $t^+ > t$ such that $\hat{P}_\tau \in C^+(\mathbf{B}, \hat{P}_t)$ for all $\tau > t^+$. By Prop. 9, $\mathrm{cl}(C^+(\mathbf{B}, \hat{P}_\tau)) \subset C^+(\mathbf{B}, \hat{P}_t)$ for all $\tau > t^+$ and since the latter set is open, $Q \notin C^+(\mathbf{B}, \hat{P}_t)$ has a strictly positive distance from all the former sets. In particular, there exists a $\delta > 0$ such that for all $\tau > t^+$, we have $||Q - \hat{P}_\tau|| > \delta$, which contradicts the assumption that Q is an accumulation point. □

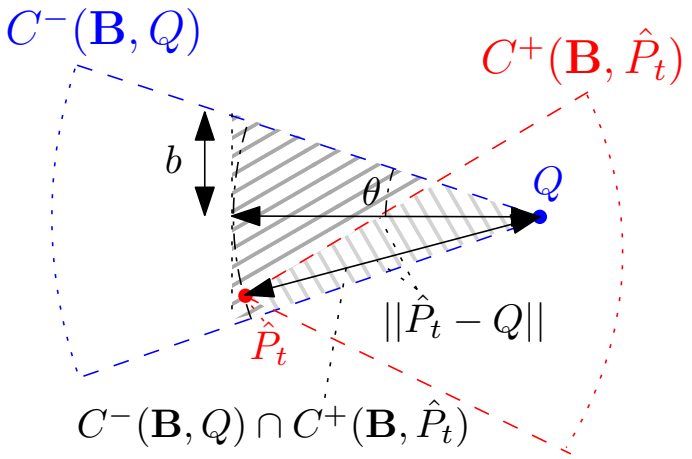


Figure 4.9: The cones $C^-(\mathbf{B}, Q)$ and $C^+(\mathbf{B}, \hat{P}_t)$ and their intersection are illustrated. In the darker grey shaded region is a right-angled cone containing the intersection with base radius b and height $||\hat{P}_t - Q||$.

Proposition 22. *Let $\hat{P}_0 \in \Xi$ and $\hat{P}_0 \notin cl(\mathbf{B})$ where $\mathbf{B}$ is from Assumption 6. Any accumulation point Q of the trajectory $\hat{P}$ is a limit point.*

Proof. Fix $t \geq 0$. By Prop. 19, $\hat{P}_t \in C_0^+(\mathbf{B}, \hat{P}_0) \cap \Xi$ and hence $\hat{P}_t \in \Xi$ and $\hat{P}_t \notin \mathrm{cl}(\mathbf{B})$. Again by Prop. 19, $\hat{P}_\tau \in C_0^+(\mathbf{B}, \hat{P}_t)$ for all $\tau > t$. By Prop. 21, $Q \in C^+(\mathbf{B}, \hat{P}_\tau)$ which by Prop. 8 implies $\hat{P}_\tau \in C^-(\mathbf{B}, Q)$, for all $\tau \geq t$. Therefore, $\hat{P}_t \in C^-(\mathbf{B}, Q)$ and $Q \in C^+(\mathbf{B}, \hat{P}_t)$, see Fig. 4.9, and $\hat{P}_\tau \in C^-(\mathbf{B}, Q) \cap C_0^+(\mathbf{B}, \hat{P}_t)$ for all $\tau > t$.

Let θ be the opening angle of the cone $C^-(\mathbf{B}, Q)$. The set $C^-(\mathbf{B}, Q) \cap C_0^+(\mathbf{B}, \hat{P}_t)$ is contained in a right-angled cone of base radius $b = ||\hat{P}_t - Q|| \tan\theta$ and height $||\hat{P}_t - Q||$ because $\left|\frac{Q}{||Q||} \cdot (\hat{P}_t - Q)\right| \leq ||\hat{P}_t - Q||$, see Fig. 4.9 and recall that $\mathbf{B}$ is centred at 0.

By Prop. 12, this right-cone is contained in an open ball around Q of radius $2\sqrt{1 + \tan^2\theta}\,||\hat{P}_t - Q||$. Thus $||\hat{P}_t - Q|| < \delta$ implies $||\hat{P}_\tau - Q|| < 2\sqrt{1 + \tan^2\theta} \cdot \delta$ for all $\tau > t$.

This implies that Q is a limit point, because given $\rho > 0$ there exists a $t \geq 0$ such that $||\hat{P}_t - Q|| < \rho/(2\sqrt{1+\tan^2\theta})$ since Q is an accumulation point, and hence $||\hat{P}_\tau - Q|| < \rho$ for all $\tau > t$. □

The limit point can not be in $C^+(\mathbf{B}, \hat{P}_0) \cap \Xi$

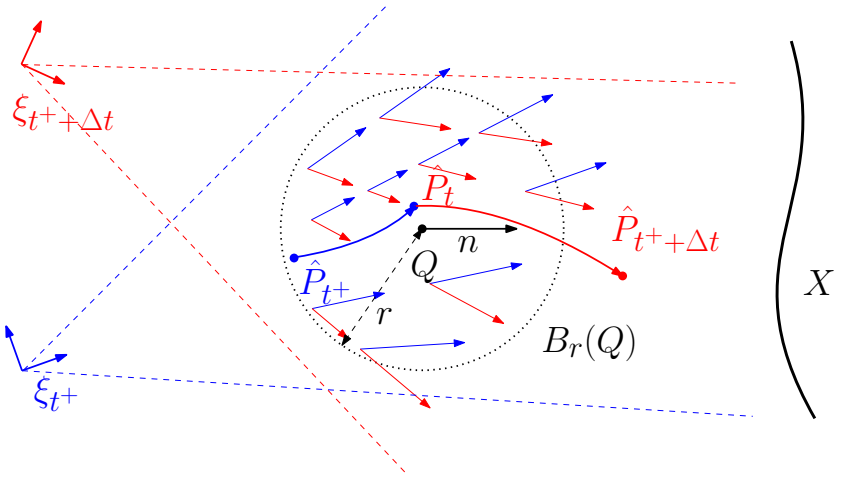


Figure 4.10: There is an open ball of radius $r > 0$ around a limit point $Q \in \Xi$ for which point estimates entering the ball eventually leave. In this diagram, the vector field v is shown for two different times shown in red and blue. There is a vector n and a $c > 0$ for which each of the vectors $v_\tau(\hat{P}')$ assigned to a point $\hat{P}'$ in the ball at a time $\tau \in [t^+, t^+ + \Delta t]$ satisfies $n \cdot v_\tau(\hat{P}') \geq c$.

The following proposition implies that if the trajectory $\hat{P}$ enters a certain nonempty open ball around the limit point Q it will eventually leave that ball, see Fig. 4.10.

Proposition 23. *Let $\hat{P}_0 \in \Xi$ and $\hat{P}_0 \notin \mathbf{B}$ where $\mathbf{B}$ is from Assumption 6. Let $Q \in \Xi$ be a limit point of the trajectory $\hat{P}$, and let Δt be the length of time from Assumption 9. Then there exists a direction n, a $c > 0$, an $r > 0$, and a sequence $(t_i^+)_{i=1}^\infty$ of times with $t_i^+ > 0$ for all $i \in \mathbb{N}$ and $\lim_{i\to\infty} t_i^+ = \infty$, such that for all $i \in \mathbb{N}$ and for all times $\tau \in [t_i^+, t_i^+ + \Delta t]$ and all points $\hat{P}' \in B_r(Q)$, $n \cdot v_\tau(\hat{P}') \geq c$.*

Proof. Let $\rho > 0$ be the radius from Assumption 9 and choose $0 < r < \frac{\rho}{2}$ such that $B_r(Q) \subset \Xi$ and $B_r(Q) \cap \mathbf{B} = \emptyset$. Such an r exists since Ξ is open and Q has a positive distance from $\mathbf{B}$ by Prop. 21. Because $\mathbf{B}$ and $B_r(Q)$ are both convex and non-intersecting, there exists a separating hyperplane $\mathbf{P}$ between them. Let n be the unit normal vector to this hyperplane pointing in the direction of Q.

Since Q is a limit point, there exists a time $t \geq 0$ such that $\hat{P}_\tau \in B_r(Q)$ for all $\tau > t$, and by Assumption 9, there exists a sequence $(t_i^+)_{i=1}^\infty$ with $t_i^+ > t \geq 0$ for all $i \in \mathbb{N}$

and $\lim_{i\to\infty} t_i^+ = \infty$, such that $\pi_{\xi_\tau}(B_\rho(\hat{P}_\tau)) \subset \mathcal{L}_\tau^*$ for all $i \in \mathbb{N}$ and $\tau \in [t_i^+, t_i^+ + \Delta t]$. Because $r < \frac{\rho}{2}$, this implies $\pi_{\xi_\tau}(\mathrm{cl}(B_r(Q))) \subset \mathcal{L}_\tau^*$ for all $i \in \mathbb{N}$ and $\tau \in [t_i^+, t_i^+ + \Delta t]$.

Now fix $\hat{P}' \in \mathrm{cl}(B_r(Q))$, $i \in \mathbb{N}$ and $\tau \in [t_i^+, t_i^+ + \Delta t]$ and let $\ell_\tau = \pi_{\xi_\tau}(\hat{P}')$. Then $\ell_\tau \in \mathcal{L}_\tau^*$ and hence $v_\tau(\hat{P}') = -\nabla_1\epsilon(\hat{P}' \cdot \eta_\tau(\ell_\tau), \ell_\tau)\eta_\tau(\ell_\tau)$. Because $\eta_\tau(\ell_\tau)$ points from $\ell_\tau \in \mathbf{B}$ into the direction of $\hat{P}' \in \mathrm{cl}(B_r(Q))$ on the other side of the hyperplane $\mathbf{P}$, and because $\nabla_1\epsilon(\hat{P}' \cdot \eta_\tau(\ell_\tau), \ell_\tau) < 0$ by Lemma 13, it follows that $n \cdot v_\tau(\hat{P}') > 0$.

Changing coordinates to the main lens $\mathcal{A}$ as in the proof of Lemma 13 gives

$$\nabla_1\epsilon(\hat{\Delta}, \ell) = \int_{\mathcal{A}} D_{\hat{\Delta}} \left|\left|\beta(P) - \beta(\pi_{\hat{\Delta}\eta}^{-1}(a))\right|\right|^2 da, \tag{4.6}$$

where $\eta = \eta(\ell)$ and $\pi_{\hat{\Delta}\eta}^{-1}(a)$ is the perspective projection from $\mathcal{A}$ through $\hat{P} = \hat{\Delta}\eta(\ell)$ to X. Note that the integrand is defined as the derivative of the sum of absolute differences of a composition of perspective projections and the smooth colouring β, and so the expression on the right hand side of (4.6) is a continuous function F of $\hat{\Delta}$ and η, as long as η points away from the main lens and towards P. It follows that $-F(\hat{\Delta}, \eta)\eta \cdot n$ attains its minimum $c > 0$ on the compact set $\{(\hat{\Delta}, \eta) \mid \hat{\Delta}\eta \in \mathrm{cl}(B_r(Q)) \text{ and } \hat{P}_0 + \eta \in \mathrm{cl}(C^+(\mathbf{B}, \hat{P}_0))\}$. Here we have used that n points towards Q and $Q \in C^+(\mathbf{B}, \hat{P}_0)$ by Prop. 21.

Since $\hat{P}' \in \mathrm{cl}(B_r(Q))$ and $\hat{P}_0 + \eta_\tau(\ell_\tau) \in \mathrm{cl}(C^+(\mathbf{B}, \hat{P}_0))$ by Prop. 19, it follows that $n \cdot v_\tau(\hat{P}') = -\nabla_1\epsilon(\hat{P}' \cdot \eta_\tau(\ell_\tau), \ell_\tau)\eta_\tau(\ell_\tau) \cdot n \geq c$. □

It now follows that there can not be a limit point of the trajectory $\hat{P}$ in $C^+(\mathbf{B}, \hat{P}_0) \cap \Xi$.

Lemma 24. *Let $\hat{P}_0 \in \Xi$ and $\hat{P}_0 \notin \mathbf{B}$ where $\mathbf{B}$ is from Assumption 6. Then the trajectory $\hat{P}$ has no limit point in the set $C^+(\mathbf{B}, \hat{P}_0) \cap \Xi$.*

Proof. Suppose for a contradiction that the point $Q \in C^+(\mathbf{B}, \hat{P}_0) \cap \Xi$ were a limit point of the trajectory $\hat{P}$.

Let Δt be the length of time from Assumption 9. By Prop. 23 there exists a direction n, a $c > 0$, an $r > 0$, and a sequence $(t_i^+)_{i=1}^\infty$ of times with $t_i^+ > 0$ for all $i \in \mathbb{N}$ and $\lim_{i\to\infty} t_i^+ = \infty$, such that for all $i \in \mathbb{N}$ and for all times $\tau \in [t_i^+, t_i^+ + \Delta t]$ and all points $\hat{P}' \in B_r(Q)$, $n \cdot v_\tau(\hat{P}') \geq c$.

Pick $r' < \min\{r, \frac{c \cdot \Delta t}{2}\}$ then there exists a time $t \geq 0$ such that $\hat{P}_\tau \in B_{r'}(Q)$ for all $\tau > t$ because Q is a limit point. Pick $i \in \mathbb{N}$ with $t_i^+ > t$ then $\hat{P}_{t_i^+ + \Delta t} \notin B_{r'}(Q)$ because $n \cdot v_\tau(\hat{P}') \geq c$ for all $\tau \in [t_i^+, t_i^+ + \Delta t]$ and all $\hat{P}' \in B_{r'}(Q) \subset B_r(Q)$, a contradiction. □

Convergence of Point Estimates to the Scene

Theorem 25. *Let $\hat{P}_0 \in \Xi$ and $\hat{P}_0 \notin cl(\mathbf{B})$, where $\mathbf{B}$ is from Assumption 6. Then there exists a point $P \in X$ such that $\lim_{t\to\infty} \hat{P}_t = P$.*

Proof. By Prop. 19, $\hat{P}_t$ is contained within $C_0^+(\mathbf{B}, \hat{P}_0) \cap \Xi$ for all $t \geq 0$. The Bolanzo-Weierstrass theorem implies that the trajectory $\hat{P}$ has an accumulation point Q within the closure of that set. By Prop. 22, Q is a limit point. By Lemma 24, $Q \notin C^+(\mathbf{B}, \hat{P}_0) \cap \Xi$ but by Prop. 21, $Q \in C^+(\mathbf{B}, \hat{P}_0)$. Therefore, $\hat{P}_t$ has a limit on $\partial\Xi = X$. □

The case $\hat{P}_0 \in \Xi^c$ follows along the same lines, replacing positive cones with negative cones where appropriate. The case $\hat{P}_0 \in X$ follows trivially from Prop. 14. Note that it is quite difficult to add further details about the properties of the point P that the point estimate $\hat{P}$ converges to. This is because the point P that the estimate $\hat{P}$ converges to depends on the trajectory of the camera. Referring back to (4.3), we find that any point estimate $\hat{P}$ on the scene will have a correct range estimate $\hat{\Delta}$ that will result in the photometric error being 0. This is because when the range estimate $\hat{\Delta}$ is correct, we have that the pairwise lenslet error $e(\ell', \hat{\Delta}, \ell) = 0$. Thus, any point on the scene is potentially a limit point of the point estimate $\hat{P}$.

Further note that Theorem 25 is a theoretical result based on perfect measurements of the gradient of the the error, as described by (4.4). When these gradients cannot be computed exactly, the point estimate cannot converge exactly to the scene. However, it is expected that the point estimate will still move to within some distance of the scene, where the distance depends on the noise in the error gradient. An exact analysis of the behaviour of the point estimate in the presence of noise in the error gradient calculation is a topic of future work.

4.6 Conclusion

In this chapter, an observer was developed that uses known camera trajectories and light-field measurements to produce estimates of depth maps. The proposed observer exploits the concept of plenoptic cameras as continuous sets of pinhole cameras to derive an innovation term given by the gradient of an integral error term. The asymptotic convergence of a point estimate to the true scene is proven for scenes satisfying some basic assumptions. The correctness of the observer algorithm is illustrated using a simulation of a simple scene. Future work includes experimentation with different, more robust error functions and experimentation with real light-field video camera data.

Chapter 5

An Observer for Estimating an Implicit Scene Representation

This chapter proposes a method of reconstructing the dense structure of scenes from visual or depth sensors that provably converges in finite time. We represent the scene as a superlevel set of a function that resides within some potentially infinite-dimensional function space. The observer state is determined by the parameters of the function that represents the scene. In experiments, we show that the observer exhibits convergence behaviour on a variety of different function spaces both in simulation and with real light-field camera data. This content of this chapter is based on the paper [76].

5.1 Background

The vast majority of the literature on the subject of estimating scenes and objects comes from the computer vision community. In that community, the goal is typically to produce dense reconstructions of objects from their sparse point-cloud representations, and the topic is known as 3D reconstruction or surface reconstruction. Typically, the point clouds that these methods use must be oriented, so that they specify positions in space and corresponding directions normal to the surface to be estimated. The points are assumed to lie on the boundary of the set representing the object and each direction specifies the direction of the gradient of the characteristic function of the set at the corresponding point. The approach introduced by Kazhdan *et al.*[48], known as Poisson surface reconstruction, then solves Poisson's equation using this gradient information in order to estimate the characteristic function. Later work exploits this information in order to simplify estimates of coefficients of Fourier series, as in Kazhdan *et al.*[47], or wavelet representations as in Manson *et al.*[65]. Recently, in Mescheder *et al.*[70], neural networks have been trained on large datasets

in order to produce a measurement-dependent characteristic function that takes as input a point and a measurement and returns a likelihood that the point is occupied given that measurement. All of the mentioned techniques take hours of computation time on dedicated hardware.

The robotics community has also developed techniques for solving this problem over the last few decades. A standard technique, known as occupancy grid mapping was introduced by Thrun and Bü [105]. This technique studies the assignment of function values to discrete voxels. Typically these techniques have a Bayesian flavour, coming from the perspective of machine learning, and the functions being estimated are conditional probability distributions. While the majority of these methods estimate functions defined on a regular voxel grid, recent progress has been made on continuous occupancy mapping techniques [89, 94]. Again, the literature on this topic comes from a probabilistic perspective. Although the experimental results of these methods are promising, theoretical guarantees of the correctness of 3D reconstruction or occupancy mapping techniques are not provided in these papers.

To the authors' knowledge, it has not before been recognised that occupancy grid mapping techniques are observers, albeit observers with trivial state dynamics. The ramifcations of this observation include potentially adding internal models to these techniques in order to produce online dense 4D reconstructions of evolving environments. 4D reconstruction is yet another developing topic within computer vision that concerns estimation of the dense geometry of a scene together with its time evolution. Most published methods on 4D scene reconstruction are performed offline in post-processing [73]. As with the occupancy mapping approaches, there is good experimental evidence that these methods produce accurate results, but theoretical proofs of convergence are not supplied.

In this chapter, we derive an observer that estimates characteristic functions of scenes, in a way that does not depend on the function class of which the characteristic function is assumed to be a member. We prove that the derived observer exhibits point-wise finite-time convergence from dense measurements of the scene, such as those that may be obtained from a light-field camera or laser range finder, under certain assumptions. We further show that interpretation of the function values should not help with the analysis, and that any update function with the right properties will result in a converging scene estimate. Finally, we demonstrate that the derived observer works in simulation and on real light-field camera data.

The remainder of this chapter is structured as follows: in Section 3.1 we develop a theoretical framework for the observer, starting with implicit representations of scenes using extended characteristic functions, parametrisations of the space of functions used to represent the scenes, errors of scene estimates, and measurements of

scenes. In Section 5.3 we develop a general observer for scene reconstruction that applies regardless of the scene representation chosen, and derive different instances of the observer for several chosen function classes: voxels, wavelets, and neural networks. In Section 5.4 we demonstrate that the observer exhibits convergence behaviour both in simulation and with real light-field camera data. In Section 5.5 we prove that this observer can estimate points on the scene in finite time, even if the function class chosen to represent scenes itself is infinite dimensional.

5.2 Notation and Terminology

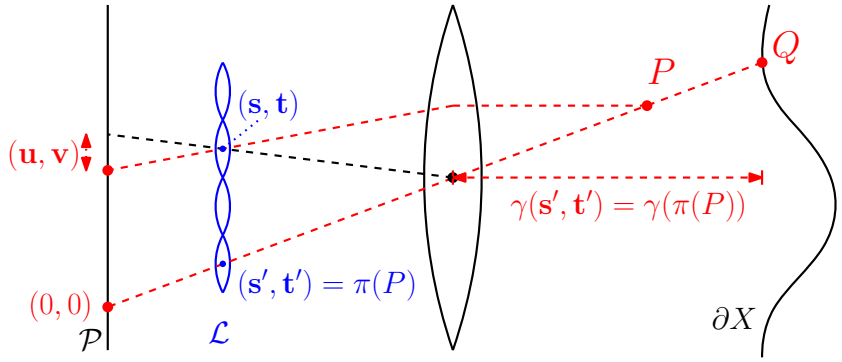


Figure 5.1: Light-field geometry: a point P is imaged by lenslets $(\mathbf{s}, \mathbf{t})$ and $(\mathbf{s}', \mathbf{t}')$. Since the ray that passes through $(\mathbf{s}', \mathbf{t}')$ and P passes through the optical centre of the focus lens, it has offset $(0, 0)$ and we set $\pi(P) = (\mathbf{s}', \mathbf{t}')$. The ray that passes through the lenslet $(\mathbf{s}, \mathbf{t})$ is refracted by the focal lens and appears in the subimage produced by the lenslet $(\mathbf{s}, \mathbf{t})$ at pixel $(\mathbf{u}, \mathbf{v})$. The depth $\gamma(\mathbf{s}', \mathbf{t}')$ assigned to lenslet $(\mathbf{s}', \mathbf{t}')$ is the depth of the point Q on the scene surface ∂X. Observe that ${}^{\mathsf{C}}P^z < \gamma(\pi(P))$ as the point P is in front of the scene.

Much of the notation used in this chapter was covered in Chapters 1 and 2. However, some additional concepts are introduced in this chapter which require separate notation and terminology, which will be introduced in this section.

In this chapter, we assume that the scene X is contained in some larger superset $M \subset \mathbb{R}^3$. We represent the scene X implicitly as the zero superlevel set of some function $\chi : M \to \mathbb{R}$, and the scene surface ∂X by the zero level set of the same function. By zero superlevel set, we mean the set of all points $P \in M$ such that $\chi(P) \geq 0$. We call the function χ an *extended characteristic function*. However, as a result of this representation method, there are likely to be many functions χ that represent the same scene because they have the same zero superlevel set and the same zero level set. Thus, there is an equivalence relation $\chi_1 \sim \chi_2$ if $\chi_1^{-1}(\mathbb{R}^+) = \chi_2^{-1}(\mathbb{R}^+)$ and $\chi_1^{-1}(0) = \chi_2^{-1}(0)$. Every element in the equivalence class $[\chi]$ represents the same

scene $X = \chi^{-1}(\mathbb{R}^+)$.

We exploit our representation of the scene as a function, and define the distance between a scene estimate $\hat{X} \subset M$ and the true scene $X \subset M$ with extended characteristic functions $\hat{\chi}$ and χ, respectively as:

$$E([\hat{\chi}], [\chi]) := \int_M ||\text{sgn}(\hat{\chi}(P)) - \text{sgn}(\chi(P))||^2 \, dP. \tag{5.1}$$

Note that the value of this error does not depend on which representatives $\hat{\chi}$ or χ of the equivalence classes $[\hat{\chi}]$ and $[\chi]$ are used, and so this distance is well-defined as a distance on equivalence classes of extended characteristic functions.

The observer derived in this chapter uses light-field measurements. For convenience, an illustration of the relavant light-field notation is provided in Figure 5.1. The coordinates used for lenslet-pixel pairs in this chapter are the $(\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v})$ coordinates, *cf.* Section 2.5. It was shown in Section 2.5 that disparity may be estimated from raw light-field data very efficiently, and the depth estimates used in this Chapter are extracted from raw light-field images using techniques described in that section.

We also parametrise depth maps with respect to the lenslet coordinates $(\mathbf{s}, \mathbf{t})$. In this notation, the depth-map $\gamma : \mathcal{L} \to \mathbb{R}^+$ is a map that takes the lenslet with coordinates $(\mathbf{s}', \mathbf{t}')$ and returns the depth ${}^{\mathbf{C}}Q^z$ expressed in the body-fixed frame $\mathbf{C}$ of the camera of the first point Q on the scene surface ∂X that lies along the ray with coordinates $(\mathbf{s}', \mathbf{t}', 0, 0)$, see Figure 5.1. It is also useful to define the centre perspective projection π that maps a point P in front of the camera to the location of the lenslet $(\mathbf{s}', \mathbf{t}')$ for which the ray $(\mathbf{s}', \mathbf{t}', 0, 0)$ passes through P, see Figure 5.1. The coordinates $(\mathbf{s}', \mathbf{t}')$ may or may not correspond to an actual lenslet in $\mathcal{L}$. If they do, i.e. if $\pi(P) \in \mathcal{L}$, then ${}^{\mathbf{C}}Q^z = \gamma(\pi(P))$ is the depth of the scene in direction of P.

Although the experimental results presented in this chapter use depth measurements obtained from light-field camera data, the design of this observer is applicable to other sensors, and it is not necessary that depth is explicitly computed. What is important is that the obtained measurements can be used to derive a parameter update with the correct properties. So long as the measurement μ can be used to define an update with the properties described in Section 5.3, the derived observer will produce an estimate of the extended characteristic function χ that asymptotically converges to the equivalence class $[\chi]$.

5.3 Observer design

In this section, we derive a class of observers for estimating implicit representations of scenes from depth measurements. The problem that we propose a solution to in

this section is: given the time-varying depth measurements γ_t and the pose trajectory of the sensor ξ_t, find at time t an estimate of the parameters $\hat{\theta}_t$ of the implicit representation $\chi_{\hat{\theta}_t}$ so that $\chi_{\hat{\theta}_t}^{-1}(\mathbb{R}^+)$ converges to the true scene X.

We assume that the true scene X actually resides within our scene class $\mathbf{X}$, so that there are parameters θ such that $\chi_\theta^{-1}(\mathbb{R}^+) = X$. We assume that the scene is stationary in our experiments and analysis. While an assumption of a stationary scene is a standard one in most SLAM and 3D reconstruction algorithms, our observer approach allows the introduction of non-trivial dynamics into the model. However, in this chapter we assume that the parameters that represent the true scene are constant, that is

$$\dot{\theta}_t = 0. \tag{5.2}$$

At time t, we receive a partial measurement of the scene surface in the form of a depth-map γ_t that is computed from the light-field measurement μ_t. At time t, we also have a parameter estimate $\hat{\theta}_t$ that determines a scene estimate $\hat{X}_t = \hat{\chi}_{\hat{\theta}_t}^{-1}(\mathbb{R}^+)$. From the depth measurement γ_t, we may compute the error of the current parameter estimates given the current depth estimates by computing what the sign of the current extended characteristic function estimate is, and what the depth measurement says the sign of the function value should be:

$$\epsilon(\hat{\theta}_t, \gamma_t) := \int_{\pi_t^{-1}(\mathcal{L})} (\operatorname{sgn}(\hat{\chi}_{\hat{\theta}_t}(P)) - \operatorname{sgn}({}^{\mathbf{C}}P^z - \gamma_t(\pi_t(P))))^2 dP. \tag{5.3}$$

Ideally, the observer dynamics would be written in the standard innovation term form $\dot{\hat{\theta}}_t = -\nabla_1 \epsilon(\hat{\theta}_t, \gamma_t)$, where the internal model term is zero according to (5.2). The gradient of the integrand may not be well-defined due to the presence of the 'sgn' function, however, we still pursue the idea of updating $\hat{\theta}_t$ in the direction that minimises the error $\epsilon(\hat{\theta}_t, \gamma_t)$. One potential alternative approach that may circumvent the issue of the 'sgn' function in the gradient that warrants further investigation is to use a Clarke derivative instead. However, this idea is left as future work.

To do this, we initialise the extended characteristic estimate so that

$$\chi_{\hat{\theta}_0}(P) = 0 \text{ for all } P \in M. \tag{5.4}$$

and calculate $\dot{\hat{\theta}}_t$, so that the following is satisfied:

$$\operatorname{sgn}(\dot{\chi}_{\hat{\theta}_t}(P)) = \begin{cases} \operatorname{sgn}({}^{\mathbf{C}}P^z - \gamma_t(\pi_t(P))) & \pi_t(P) \in \mathcal{L} \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

Note that this is not a definition for a single observer, but a constraint that defines an entire class of observers. In this chapter, we consider any system such that $\dot{\hat{\theta}}_t$ satisfies (5.5), rather than defining a single system that is uniquely determined by some given error dynamics. Under certain mild conditions (see Section 5.5), any observer $\chi_{\hat{\theta}_t}$ that satisfies (5.5) will converge in finite-time (see Prop. 27).

The way in which the derivative of the parameters is computed depends on the choice of representation used. In this chapter, we test this method using voxel, wavelet, and neural network representations and show that there are practical ways of implementing this method for each of these choices. Sometimes, in the case where the relationship between the parameters and function outputs is trivial as in voxels, or linear as in wavelets, there are techniques for simplifying the calculation of new parameters from the old parameters. However, even when the relationship between the parameters and function outputs is nonlinear, as in neural networks, there are still techniques for easily computing the update of the parameters. Further note that any actual implementation of such an observer will be finite-dimensional as real computers only have a finite amount of memory to store the parameters. However, by proving convergence for the infinite-dimensional case in Section 5.5, we show that there is no upper bound on the number of parameters or level of detail that may be desired in practice.

5.3.1 Voxel Representation

In this section, we demonstrate the estimation of an extended characteristic function using a voxel representation. In this case, the set of functions is given by $\{\chi : M \to \mathbb{R}\}$, and M is a discrete grid of 3D points $P = (i, j, k) \in \mathbb{Z}^3$ where $i_{\min} \leq i \leq i_{\max}, j_{\min} \leq j \leq j_{\max}, k_{\min} \leq k \leq k_{\max}$. A function $\chi \in \mathcal{F}$ is determined by the parameters $\theta_P := \chi(P)$ where $P \in M$. In this case, the extended characteristic function is updated directly with

$$\dot{\hat{\theta}}_P := \begin{cases} \operatorname{sgn}({}^{\mathbf{C}}P^z - \gamma_t(\pi_t(P))) & \pi_t(P) \in \mathcal{L} \\ 0 & \text{otherwise.} \end{cases} \tag{5.6}$$

5.3.2 Curvelet Representation

Curvelets were constructed with the goal of finding sparse representations of functions that have discontinuities along C^2-curves, as is common in image processing and computer vision tasks. Whereas classical wavelets are functions that, under translation and scaling, form a basis of $L^2(\mathbb{R})$, curvelets are functions that under translation, parabolic scaling, and rotation form a *Parseval frame* of $L^2(\mathbb{R}^2)$. A Parseval frame for $L^2(\mathbb{R}^2)$ is a family of functions $\{\phi_i\}_{i=1}^{\infty}$ that satisfy Parseval's identity, namely that for all $\psi \in L^2(\mathbb{R}^2)$ we have that $\sum_{i=1}^{\infty} |\langle \psi, \phi_i \rangle|^2 = ||\psi||^2$. In the curvelet literature, curvelets are often said to form a tight frame, however this is not to be confused with other notions of a tight frame which only require Parseval's relation to hold up to a constant scale (see [17]).

We will not give a complete description of how a curvelet is constructed in this chapter, a more comprehensive overview of curvelets is provided by Candes *et al.*[13]. Curvelets are typically constructed by taking a mother curvelet φ, and defining the curvelet family $\varphi_{j,k,l}$, that depends on the parameters $j \in \mathbb{N}$, $l \leq 2^j \in \mathbb{N}$, and $k \in \mathbb{Z}^2$. The family of functions $\varphi_{j,k,l}(x) := 2^{3j/2}\varphi(D_j R_{j,l} x - k_\delta)$, where D_j is a parabolic scaling matrix, $R_{j,l}$ is a rotation matrix, and k_δ is a translation depending on a predetermined fixed parameter δ form a Parseval frame of $L^2(\mathbb{R}^2)$. This notion may be extended to construct a Parseval frame of $L^2(\mathbb{R}^3)$, for more details see [123].

To use a curvelet representation, consider an extended characteristic function $\chi \in L^2(M)$, where M is some rectangular prism in $\mathbb{R}^3$. Since $\chi \in L^2(M)$, it has a curvelet expansion

$$\chi(P) = \sum_{j,k,l} \theta_{j,k,l} \varphi_{j,k,l}(P). \tag{5.7}$$

The parameters of the extended characteristic function χ in this representation are the coefficients

$$\theta_{j,k,l} = \int_M \chi(P) \varphi_{j,k,l}(P) dP. \tag{5.8}$$

In order to update these coefficients given a depth measurement γ_t, we approximate the coefficients $\Delta\hat{\theta}_t$ of $\dot{\chi}_{\hat{\theta}_t}$ by computing the 'update' function

$$v_t(P) := \begin{cases} \operatorname{sgn}({}^{\mathbf{C}}P^z - \gamma_t(\pi_t(P))) & \pi_t(P) \in \mathcal{L} \\ 0 & \text{otherwise.} \end{cases} \tag{5.9}$$

By taking the curvelet transform of (5.9), we obtain a sequence of curvelet coefficients $\Delta\hat{\theta}_t$ that may be added to the coefficients $\hat{\theta}_t$ in order to update them. We exploit the sparsity of the coefficients for real scenes, by keeping only the N most significant coefficients in the state after applying the update, and setting the rest to 0, which

filters the effects of high-frequency noise that may be present in the measurements (see Section 5.4.1).

5.3.3 Neural Network Representation

A feed-forward neural network is a function $\chi : \mathbb{R}^m \to \mathbb{R}^n$ that is a finite iterated composition of functions of the form

$$P \mapsto \sigma_l(A_l P + b_l),$$

where $\{\sigma_l\}_{l=1}^L$ are nonlinear functions known as activation functions, A_l is a matrix and b_l is a vector. Let

$$f_l(P) := \sigma_l(A_l P + b_l),$$

then the neural network function is given by

$$\chi(P) := (f_L \circ \cdots \circ f_1)(P).$$

The number of functions in the composition is L and is known as the depth of the network. Given that the activation functions σ_l are chosen beforehand, the parameters of the neural network function are given by the sequence of matrices and vectors: $\theta := (A_l, b_l)_{l=1}^L$.

The parameters of this representation are updated by taking a random sample S of points in $\pi_t^{-1}(\mathcal{L})$, pairing each $P \in S$ with an ideal value $y(P)$, and performing back-propogation on the training pairs $\{(P, y(P))\}_{P \in S}$ for a small number of training steps using the error

$$\tilde{\epsilon}(\hat{\theta}_t, \gamma_t) := \sum_{P \in S} \left|\left| \chi_{\hat{\theta}_t}(P) - y(P) \right|\right|^2,$$

where $\chi_{\hat{\theta}_t}$ denotes the function that is computed by a neural network with parameters $\hat{\theta}_t$.

In effect, this process will approximate $-\nabla_1 \tilde{\epsilon}(\hat{\theta}_t, \gamma_t)$ and update the parameters in the direction of this gradient. The ideal value $y(P)$ assigned to point P will depend on the choice of activation functions used. For example, if the activation function on the final layer is $\sigma_L(h) = 2\tilde{\sigma}(h) - 1$, where $\tilde{\sigma}$ is a sigmoid function, then the range of the neural network function is $(-1, 1)$, in which case letting

$$y_t(P) = \begin{cases} \operatorname{sgn}({}^{\mathbf{C}}P^z - \gamma_t(\pi_t(P))) & \pi_t(P) \in \mathcal{L} \\ \hat{\chi}_{\theta_t}(P) & \text{otherwise} \end{cases} \tag{5.10}$$

will result in an update that approximates (5.5). The parameters $\hat{\theta}_t$ of the neural

network $\chi_{\hat{\theta}_t}$ are updated by training the neural network on the new target function defined by (5.10) for a small number of steps.

5.4 Experiments

In this section, we provide both simulated and experimental evidence for the correctness of our approach. The simulated scene is a 3D model of a bas-relief obtained from Maier *et al.*[63]. This dataset is chosen in our simulation because it best reflects the theoretical assumptions on the scene given in Section 5.5. This data is provided in a point cloud format, from which a triangle mesh is computed. The domain M is chosen so that the triangle mesh divides the domain into two halves, and from this the true characteristic function can be computed.

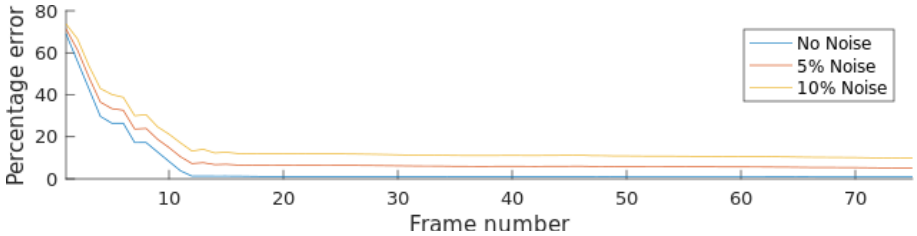
We also test the observer on a real scene using data produced by a Lytro Illum camera. However, since ground truth is not available, error trajectories are not practical to compute for this data. The final 3D reconstructions of the observer are provided instead for visual inspection. A sample central sub-aperture image of the scene is shown in Fig 5.4 for comparison. The scene consists of an object to be reconstructed and a checkerboard. The checkerboard is used to calibrate the camera, providing both estimates of the camera intrinsic parameters ϕ and of the pose of the camera ξ for each frame. A total of 101 frames are used in this experiment.

For each simulated frame, an error comparing the estimated characteristic function with the true characteristic function based on the known scene geometry is reported. This error is defined on the output of the function, not on the parameters of the function. Given a regularly sampled voxel grid G on the input space M, we compute at each time step the approximate error

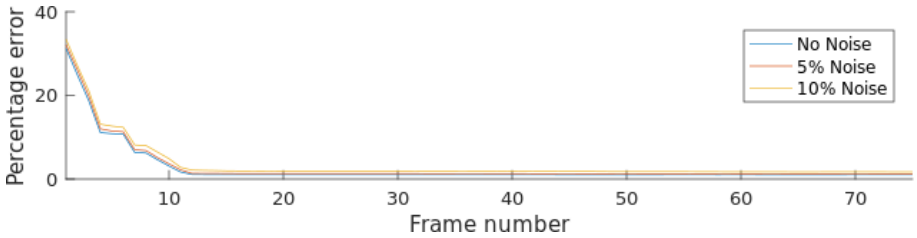
$$\tilde{E}([\hat{\chi}_t], [\chi]) := \frac{1}{2} \sum_{P \in G} ||\text{sgn}(\hat{\chi}_t(P)) - \text{sgn}(\chi(P))||^2 .$$

Graphs of these errors for each of the representation methods are shown in Fig 5.2. Final reconstructions of the simulated scene are shown in Fig 5.3.

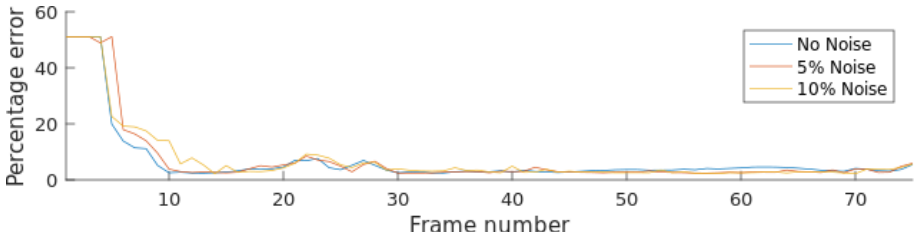
We represent the extended characteristic functions in several different ways in order to demonstrate that our approach is not limited to a specific function class or representation. For the voxel representation, the resolution used is $128 \times 128 \times 128$. For the curvelet representation, we utilize the Curvelab toolbox [14] to implement a discrete curvelet transform. At each timestep, we progressively increase the number of parameters used. The maximum number of parameters used in the curvelet representation is 10000, that is 209 times lower than what is required for the voxel representation. At each timestep, after the update is applied, the most significant



(a) Error for voxel representation



(b) Error for curvelet representation

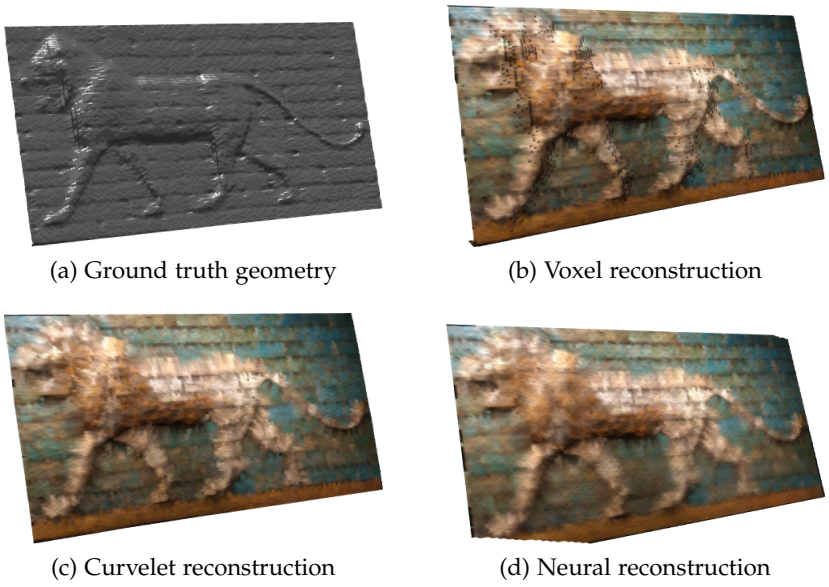


(c) Error for neural representation

Figure 5.2: Error graphs for each representation of the simulated scene.

coefficients are extracted from the current scene estimate using a method described in Starck *et al.*[98]. In our experiments, we do not use curvelets of scaling depth greater than 8. The neural network used in the simulations is a fully-connected neural network with 4 hidden layers, each layer consisting of 100, 50, 20 and 10 neurons, respectively. For the neural network representation, the number of training update steps at each time step is 20, and the activation function used at each layer is a sigmoid function. This update is fast enough to be performed online (on average 0.7 seconds per frame) for a small neural network of 6693 parameters. The parameters for this representation are not initialised to zero, as in the other methods, but assigned randomly.

In the simulations, the camera trajectory consists of 75 frames and is chosen so that every point on the scene is within the field of view of the camera at least once. By frame 20, each portion of the scene has been seen. The trajectory consists of a fast scan of the entire scene with minimal overlapping between frames followed by a slower scan of the scene for the remaining frames.



(a) Ground truth geometry (b) Voxel reconstruction

(c) Curvelet reconstruction (d) Neural reconstruction

Figure 5.3: Comparison of final reconstructions of a simulated scene shown with ground truth.

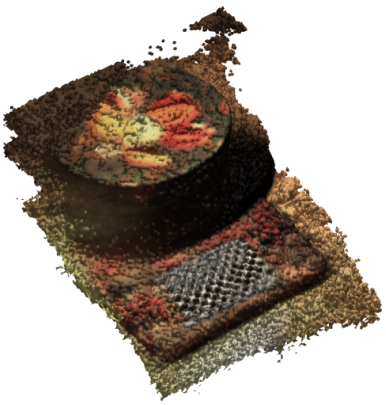
The trajectory for the real data is taken by hand, and is constrained more by keeping the checkerboard in view and in focus in order to achieve good calibration results for the data. These calibration results are used to extract pose estimates for the camera. The focal plane for the light-field camera is set to roughly 30 cm.

5.4.1 Discussion of results

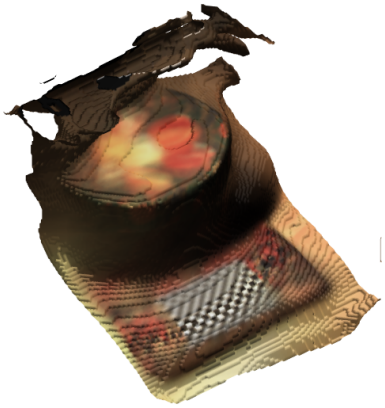
In Fig. 5.2 we show the error trajectories for the simulated data. It can be seen that the neural network method exhibits a steeper initial descent than the other methods, but the voxel and curvelet trajectories converge much faster than the neural network representation and have a lower final noise floor, as well as less variability at this noise floor. It can be seen that for the voxel and curvelet methods, the estimate approaches the noise floor by the time the fast pass over ends at frame number 12. For the neural network method, there is a brief period where the error does not decrease. This is likely due to some time being required before the parameters of the neural network represent a significantly different state to the inital state. Additionally, the neural network method exhibits oversmoothing and much of the finer details of the bas-relief are lost. The difference between the voxel and curvelet representation graphs is striking, and demonstrates that the parameter thresholding in the curvelet representation results in significant noise reduction.

Fig. 5.3 presents the final reconstructions of the simulated scene for each of

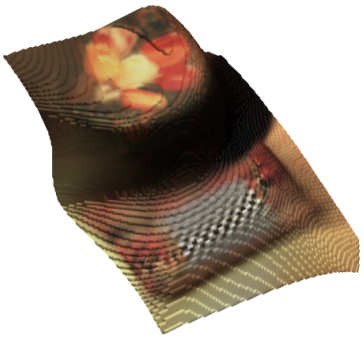
(a) Example subaperture image



(b) Voxel reconstruction



(c) Curvelet reconstruction



(d) Neural reconstruction

Figure 5.4: Final reconstructions using real light-field camera data from a Lytro Illum camera.

the methods together with the ground truth data. The final results for the voxel and curvelet methods are similar despite the latter using 200 times fewer parameters. The neural network method exhibits oversmoothing, but still produces a good approximation of the scene. It is likely that different activation functions and more sophisticated network architectures would produce superior results in this approach.

Fig. 5.4 presents the final reconstructions of the real data together with a subaperture image of the scene. Although ground truth for this scene was not available, the shape and texture of the reconstructed scene is plausible. The curvelet method seems to reduce noise when compared to the voxel method, but can result in artifacts towards the edges of the bounding box. This is a known phenomenon in the curvelet literature [13]. As with the simulated data, the neural network method seems to exhibit excessive smoothing.

5.5 Theoretical Analysis

In this section, we prove that the observer converges point-wise in finite time, despite the fact that the state is infinite-dimensional, if we can update the output values of the characteristic function directly. Further analysis of the behaviour of the observer when only the parameters may be updated is the subject of future work. We also show that the observer can be implemented using light-field measurement data.

Notation

In this section, we use the following notation for line segments. For two distinct points $P_1, P_2 \in \mathbb{R}^3$, let $[P_1, P_2]$ denote the line segment starting at P_1 and ending at P_2, that is

$$[P_1, P_2] := \{P' \in \mathbb{R}^3 : P' = P_1 + \alpha(P_2 - P_1) : \alpha \in [0, 1]\}.$$

We denote the set of positions assumed by the lenslets of the camera by K, so that if $\mathcal{L}_t \subset \mathbb{R}^3$ denotes the embedded lenslet plane at time t, then $K = \bigcup_{t \in \mathbb{R}^+} \mathcal{L}_t$.

Assumptions on the scene

There are several assumptions that are necessary in order to prove convergence of the scene estimate to the true scene. The first assumption we need is that we are estimating the portion X of some larger star-shaped scene X' that is contained within a rectangular prism M. The assumption that X' is star-shaped may be weakened slightly but results in a more complicated proof.

Assumption 10. *The portion X of the scene that is to be estimated is given by $X = X' \cap M$, where X' is star-shaped and M is a rectangular prism.*

The next assumption is necessary in order for several of the maps used in the proof to be differentiable, as well as to guarantee boundedness of the depth map.

Assumption 11. *The total scene surface $\partial X'$ is a manifold that is diffeomorphic to the sphere S^2.*

Since $\partial X'$ is diffeomorphic to the sphere, the Jordan-Brouwer separation theorem says that $\mathbb{R}^3 \setminus \partial X'$ is equal to the disjoint union of two separated sets called the interior I which is bounded, and the exterior E.

Assumptions on the camera trajectory

The following assumption is a persistency of excitation condition, and is a constraint on the camera trajectory.

Assumption 12. *For all $P \in M$ there exists a $t > 0$ and a positive number $\delta > 0$ such that $\pi_s(P) \in \mathcal{L}$ for all $s \in (t, t+\delta)$.*

That is: for each point that point is updated at least once, continuously for some interval of time. We also assume that every point in M is always in front of the camera.

Assumption 13. *The depth of every point $P \in M$ satisfies ${}^{\mathbf{C}}P^z > 0$ for all times $t \geq 0$.*

Finally, we assume that every point of the total scene surface $\partial X'$ is visible from the position of the camera at every time.

Assumption 14. *K is contained in the kernel of the interior I of $\partial X'$.*

Proof of convergence

The proof of convergence uses the following approach. Firstly, we show that any point $P \in M$ can be unambiguously said to be in front of, behind, or on the scene in a way that does not depend on a particular choice of perspective $\ell \in K$ (Proposition 26). Then we note that for points in front of the scene the characteristic value can only decrease, for points behind the scene the value can only increase, and for points on the scene the value is always zero. This leads to our point-wise finite-time convergence result (Proposition 27).

Proposition 26. *Use Assumptions 10, 11, and 14, and let*

1. *M^- be the set of points $P \in M$ such that for all $\ell \in K$ the line segment $[\ell, P]$ does not intersect $\partial X'$ (the* visible set*), and*
2. *M^+ be the set of points $P \in M$ such that for all $\ell \in K$ the line segment $[\ell, P]$ does intersect $\partial X'$ (the* occluded set*).*

Then $M \setminus \partial X = M^- \cup M^+$.

Proof. By Assumptions 10, 11 and 14, the total scene $\partial X'$ is star-shaped and K is within the kernel of the interior I. Let M^- and M^+ be the sets defined in the statement of the proposition.

Let $P \in (M \setminus \partial X) \cap E$ and let $\ell \in K$. Since $\ell \in I$ and $P \in E$, and I and E are separated, then because $[\ell, P]$ is connected, it contains a point that is in neither set. Since $\partial X' = (I \cup E)^c$, we have that there is some $x \in [\ell, P]$ such that $x \in \partial X'$. Therefore, $[\ell, P] \cap \partial X' \neq \emptyset$ for all $\ell \in K$ and all $P \in (M \setminus \partial X) \cap E$. Therefore $(M \setminus \partial X) \cap E \subset M^+$.

Let $P \in (M \setminus \partial X) \cap I$ and let $\ell \in K$. Assume, to arrive at a contradiction, that $[\ell, P] \cap \partial X' \neq \emptyset$. Since $\ell, P \in I$, the line segment $[\ell, P]$ must cross through the boundary $\partial X'$ at least twice, but this is not possible because I is a star-shaped set and ℓ is in the kernel of I. It follows that $[\ell, P] \cap \partial X' = \emptyset$ for all $\ell \in K$ and all $P \in (M \setminus \partial X) \cap I$. Therefore $(M \setminus \partial X) \cap I \subset M^-$.

Note that the sets $(M \setminus \partial X) \cap I$ and $(M \setminus \partial X) \cap E$ partition $M \setminus \partial X$ because $I \cup E = \mathbb{R}^3 \setminus \partial X'$ and $\partial X = M \cap \partial X'$. Also note that $P \in M^+$ implies $P \notin M^-$ and vice versa simply by the definitions of these sets. Therefore, $(M \setminus \partial X) \cap E = M^+$ and $(M \setminus \partial X) \cap I = M^-$. This shows that $M \setminus \partial X = M^- \cup M^+$. □

Proposition 27. *Let $\dot{\chi}_{\hat{\theta}_t}(P)$ be an integrable function satisfying Eqns.* (5.4) *and* (5.5)*. Use Assumptions 10, 11, and 14, and let M^- and M^+ be the sets defined in Proposition 26. Then, under Assumptions 12 and 13 all of the following hold:*

1. *For all $P \in M^-$ there exists a time $T \geq 0$ such that $\chi_{\hat{\theta}_\tau}(P) < 0$ for all $\tau > T$,*
2. *For all $P \in M^+$ there exists a time $T \geq 0$ such that $\chi_{\hat{\theta}_\tau}(P) > 0$ for all $\tau > T$,*
3. *For all $P \in \partial X$, we have that $\chi_{\hat{\theta}_t}(P) = 0$ for all $t \geq 0$.*

Proof. The third statement follows immediately from $\dot{\chi}_{\hat{\theta}_t}(P) = 0$ for all $t \geq 0$ and all $P \in \partial X$, and the fact that $\chi_{\hat{\theta}_0}(P) = 0$.

To show the first statement, let $P \in M^-$. Then there exists a time $t > 0$ and a $\delta > 0$ such that $\pi_s(P) \in \mathcal{L}$ for all $s \in (t, t+\delta)$ (Assumption 12).

For a given $s \in (t, t+\delta)$ let $\ell \in \mathcal{L}_s \subset K$ denote the embedded location of $\pi_s(P) \in \mathcal{L}$ and let $Q \in \partial X'$ be the point that lies on the half-line starting at ℓ and passing through P (Assumptions 10 and 14). Since $P \in M^-$ the line segment $[\ell, P]$ does not intersect $\partial X'$ hence the distance of P from ℓ is less than the distance of Q from ℓ. Now, due to Assumption 13, we have that the depth ${}^{\mathbf{C}}P^z$ of point P is also less than the depth ${}^{\mathbf{C}}Q^z = \gamma_s(\pi_s(P))$ of point Q.

Therefore, the value of ${}^{\mathbf{C}}P^z - \gamma_s(\pi_s(P))$ is negative on the interval $(t, t+\delta)$. Now, since $\pi_s(P) \in \mathcal{L}$ for $s \in (t, t+\delta)$, the derivative $\dot{\chi}_{\hat{\theta}_s}(P)$ of $\chi_{\hat{\theta}_s}(P)$ is negative on this interval. Therefore, we have that $\chi_{\hat{\theta}_{t+\delta}}(P) = \chi_{\hat{\theta}_t}(P) + \int_t^{t+\delta} \dot{\chi}_{\hat{\theta}_s}(P) ds < \chi_{\hat{\theta}_t}(P)$ because the integral is negative.

But since the derivative of $\chi_{\hat{\theta}_t}$ at P is always either negative or zero, and $\chi_{\hat{\theta}_0}(P) = 0$, we have that $\chi_{\hat{\theta}_t}(P) \leq 0$ to begin with. Therefore, $\chi_{\hat{\theta}_{t+\delta}}(P) < 0$. Let $T := t + \delta$ and note that for all future times $\tau > T$, we have that $\dot{\chi}_{\hat{\theta}_\tau}(P) \leq 0$ and hence $\chi_{\hat{\theta}_\tau} < 0$.

The statement for $P \in M^+$ follows along the same lines. □

A careful inspection of the previous proof shows that if Assumption 12 is strengthened to require the existence of a finite time $T_{\max} > 0$ such that for all $P \in M$ the

corresponding $t + \delta < T_{\max}$ then the entire scene estimate converges in finite time $T_{\max}$. More generally, any portion of the scene is reconstructed as soon as it has been seen continuously for some time.

5.6 Conclusion

In this chapter, we represent a scene as the superlevel set of an extended characteristic function. We then use dense measurements of the scene that are known to correlate with depth, such as those obtained from a light-field camera or laser range finder, in order to update the parameters of the extended characteristic function. We prove that using ideal light-field data, we may in principle perfectly reconstruct a scene under certain mild assumptions using this observer. Regardless of the dimensionality of the representation, the observer estimate converges to the true scene in finite time under the same assumptions.

Chapter 6

Conclusion

This thesis has adopted the perspective of systems theory to develop observers for scene reconstruction using the novel imaging technology of light-field cameras. Several contributions to the understanding of light-field cameras were developed throughout the course of this study. This included the development of a novel point-projection model for light-field cameras that was subsequently used to obtain state-of-the-art results for a camera calibration technique that exploited this model through use of robust feature-extraction of plenoptic discs. The relationship between this point-projection model and a ray-projection model was found, providing an explicit translation between the two models. The relationship between plenoptic disc radii and disparity were uncovered, and further investigations led to the derivation of a partial differential equation that all disparity fields obey. An examination into the limitations of depth and disparity estimation were conducted and concluded necessary and sufficient conditions that a coloured scene must satisfy in order for depth estimation from light-field gradients to be possible.

These investigations into pure light-field geometry were applied to develop new observers for scene reconstruction. The first of these techniques involved the derivation of a photometric error function based upon the idea of plenoptic discs introduced in earlier chapters. The gradient of this error function, together with the pose of the camera, determined the velocities of point estimates and the trajectories of these point estimates were proven to have the true scene as a limit set. In this way, the state estimate – which took the form of an explicit point-cloud representation – was shown to converge to a subset of the true scene without knowing which subset that was. A different approach was taken in the subsequent chapter in which an implicit scene representation was used. An implicit representation was shown to simplify theoretical analysis of the state estimate significantly by avoiding complex topological considerations and by using the fact that such a state converges if the extended characteristic values for points not on the scene *diverge* from 0. This is a far weaker statement to prove logically, and allows us to conclude finite convergence of

the scene estimate through the additional fact that if the scene is compact, then the entirety of it can be viewed in finite-time because it has a finite subcover.

6.1 Future Work

This thesis has spawned several dozen additional ideas that were not fully realised. In this section, some of these ideas will be discussed.

6.1.1 Observers for Sparse SLAM Using Light-Field Video

One avenue for future work is to extend the results of Chapter 4 to include an estimate of the camera pose through the application of recent work in equivariant observer design [62]. As a preliminary survey, an gradient-based observer for estimating visual odometry is provided here.

While light-field cameras are able to reconstruct depth maps from a single image, the reliability of these depth maps may be low in regions of little texture. Nonetheless, this suggests two methods of visual odometry: 1) we track the 3D positions of only the points that have high texture and use these estimated positions to recover the camera pose, 2) we estimate the entire depth map, generate a point cloud from that and use the iterative closest point algorithm to perform point-set registration in order to localise the camera. In this section we describe an approach to light-field SLAM adopting the former method.

Section 2.3.1 gives a precise equation relating plenoptic disc features with 3D points in the body-fixed frame of the camera. One application of this is pose estimation. For some specified points ${}^{\mathbf{O}}P_i$ where $i = 1, ..., 4$, let

$$\mathbf{P} := \begin{pmatrix} \overline{{}^{\mathbf{O}}P_1} & \overline{{}^{\mathbf{O}}P_2} & \overline{{}^{\mathbf{O}}P_3} & \overline{{}^{\mathbf{O}}P_4} \end{pmatrix}$$

and for the plenoptic discs θ_i corresponding to these points when the camera is is some unknown pose ξ, let

$$\mathbf{\Theta} := \begin{pmatrix} \lambda_1\overline{\theta}_1 & \lambda_2\overline{\theta}_2 & \lambda_3\overline{\theta}_3 & \lambda_4\overline{\theta}_4 \end{pmatrix}$$

where the constants λ_i are equal to $(\xi {}^{\mathbf{O}}P_i)^z$, and therefore

$$\lambda_i = -\frac{rK_2}{rK_1 + R_i}. \tag{6.1}$$

Since equation (2.11) holds for any P, we have that

$$\boldsymbol{\Theta} = H\xi\mathbf{P} \tag{6.2}$$

Therefore, given known essential matrix H, known points ${}^{\mathrm{O}}P_i$ and plenoptic discs θ_i corresponding to these points, we have that:

$$H^{-1}\boldsymbol{\Theta}\mathbf{P}^{-1} = \xi. \tag{6.3}$$

The matrix consisting of the four points ${}^{\mathrm{O}}P_i$ in homogenous coordinates as column vectors is invertible if and only if

$$\det \mathbf{P} \neq 0$$

which means that each of the ${}^{\mathrm{O}}P_i$ do not all lie on the same plane.

A first formulation of a pose observer using these equations can be derived by using the known dynamics of the camera. We can use (6.3) to construct a best estimate of ξ_t by finding the least-squares solution to (at time t using the feature estimates $\boldsymbol{\Theta}_t$):

$$\xi_t^* = \underset{\xi}{\operatorname{argmin}} \left|\left|\xi^{-1}\mathbf{P} - H^{-1}\boldsymbol{\Theta}_t\right|\right|^2. \tag{6.4}$$

We could also turn this into an innovation term used in an observer framework. Defining

$$\epsilon(\hat{\xi}_t, \boldsymbol{\Theta}_t; \mathbf{P}, H) = \left|\left|\xi^{-1}\mathbf{P} - H^{-1}\boldsymbol{\Theta}_t\right|\right|^2$$

and assuming that we have right-invariant system dynamics

$$\dot{\xi}_t = U_t\xi_t,$$

we get the observer dynamics:

$$\dot{\hat{\xi}}_t = U_t\hat{\xi}_t - \nabla_1\epsilon(\hat{\xi}_t, \boldsymbol{\Theta}_t; \mathbf{P}, H). \tag{6.5}$$

Some preliminary results for visual odometry produced by simulating the static estimation approach (6.4) and the observer approach (6.5) are shown in Figs. 6.1 - 6.2, respectively.

This implementation currently requires known world features, although pose relative to an initial estimate of the features from the first frame of camera data may

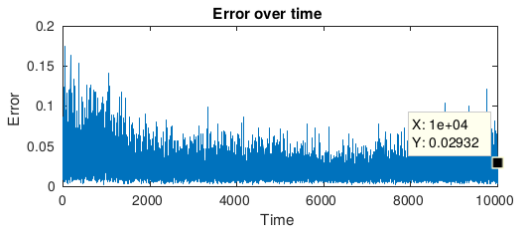


Figure 6.1: Error of pose estimates using static estimator given by (6.4).

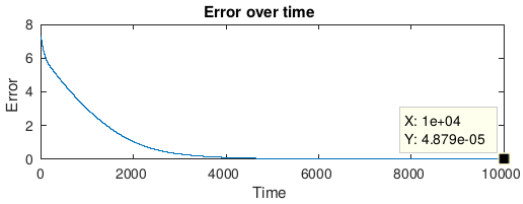


Figure 6.2: Error of pose estimates using observer given by (6.5).

be performed, better results in practice will result from updating the estimate of these world feature locations simultaneously. The path to extending these results to sparse visual SLAM requires several additional developments. Firstly use of a robust feature-matching technique for plenoptic cameras. Feature extraction from light-field data is a topic that has produced promising recent results [21] that may prove useful to plenoptic SLAM. Recent developments of equivariant observers have introduced the *SLAM manifold* [62], which formulates the state-space of the SLAM problem geometrically. The combined use of light-field feature extraction and equivariant observers is a clear path towards future publication.

6.1.2 Equivariant Observers for Implicit Dense SLAM

Chapter 5 proposed an observer for estimating dense scenes represented implicitly. There are many exciting questions that arise from the use of observers for estimating implicit representations of scenes. Suppose that we have some function class $\mathcal{F}$ inducing a scene class $\mathbf{X}$ through an implicit representation χ, so that $\mathbf{X}$ consists of the set of scenes X where there is a $\theta \in \mathbf{\Theta}$ such that $X = \chi_\theta^{-1}(\mathbb{R}_0^+)$. As observed in Section 5.2, there is an equivalence relation $\sim$ on $\mathcal{F}$. What was not noted in that chapter, however, is that this induces an equivalence relation on the parameter space $\mathbf{\Theta}$ as well. If $\chi : \mathbf{\Theta} \to \mathcal{F}$ is a representation of the function space $\mathcal{F}$ then there is an induced equivalence relation $\sim$ on $\mathbf{\Theta}$ defined by $\theta_1 \sim \theta_2$ if $\chi_{\theta_1} \sim \chi_{\theta_2}$. One problem, then, is to investigate the structure $\mathbf{\Theta}/\sim$ in order to provide unique coordinates to each scene in the scene class $\mathbf{X}$, and determine whether there is some natural way

of updating these coordinates directly given a measurement of the scene. Because this clearly depends on how the function space is represented, this also leads to the question of what kind of representation is best suited for scene reconstruction.

A second extension of Chapter 5 is a method for estimating the combined state of the scene together with the pose of the sensor imaging that scene. If we represent a scene implicitly by the function $\chi_\theta \in \mathcal{F}$ for some function space $\mathcal{F}$ and some representation χ and the pose as an element $\xi \in SE(3)$, then the total space of the dense slam problem is the set $\mathcal{T} = \mathbf{\Theta} \times SE(3)$. It is attractive, then, to consider the equivalence relation $\simeq$ defined as $(\theta_1, \xi_1) \simeq (\theta_2, \xi_2)$ if there is a $H \in SE(3)$ such that $\chi_{\theta_1}^{-1}(\mathbb{R}^+) = H\chi_{\theta_2}^{-1}(\mathbb{R}^+)$, $\chi_{\theta_1}^{-1}(0) = H\chi_{\theta_2}^{-1}(0)$, and $\xi_1 = H\xi_2$, meaning: the scene-pose pair represented by (θ_1, ξ_1) is simply a rigid body transformation of the scene-pose pair represented by (θ_2, ξ_2). Following the approach of Mahony and Tamel [62], one aveneue of future work is to determine whether the structure $\mathcal{T}/\simeq$ has a nice geometry, how this structure depends on the choice of representation, whether there is a symmetry group acting on this structure, and whether the dense SLAM problem can be posed on this group instead. Clearly such a structure will depend on the choice of representation used, and it is conceivable that some representations are better suited to such an approach than others, and since the function space is typically infinite-dimensional, techniques from infinite-dimensional analysis and geometry are likely to prove useful. Such a project would be a longer-term endeavour, but many smaller preliminary results could also result from such a study.

6.1.3 Necessary and Sufficient Conditions on Scenes for Optic Flow and Structure-from-Motion

Sections 2.5.3 and 2.5.4 provide a proof of the necessary and sufficient conditions required for depth estimation from first-order light-field properties. This theoretical result may be extended in several ways. Firstly, the proof provided relies on several assumptions which are likely either not necessary or consequences of the other assumptions. The first of these is the assumption that the scene class contains all of the fronto-parallel planar scenes. This assumption is likely not necessary because the embedded tangent planes of the scenes in the scene class already allow us to perform a similar analysis. It is noted that some of the current assumptions and definitions may constrain the topology of the scene significantly. Extending this work to additional dimensions of the scene and colour space may resolve some of these issues. It is also acknowledged that this result may be extended to optic-flow based structure-from-motion. Light-field geometry extends to the data produced by monocular video and the level-sets in a monocular video also correspond to points in a Lambertian

scene. This property is exploited in optic-flow algorithms to associate points between frames. It is likely the case that a scene being textured and Lambertian is not only sufficient to obtain structure-from-motion using optic flow, but also necessary.

References

1. E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20. MIT Press, 1991. (cited on page 15)
2. E. H. Adelson and J. Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):99–106, Feb 1992. (cited on pages xiii, 14, 15, and 38)
3. N. Aghannan and P. Rouchon. An intrinsic observer for a class of lagrangian systems. *IEEE Transactions on Automatic Control*, 48(6):936–945, June 2003. (cited on page 10)
4. A. A. Alatan, Y. Yemez, U. Gudukbay, X. Zabulis, K. Muller, C. E. Erdem, C. Weigel, and A. Smolic. Scene representation technologies for 3dtv—a survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(11):1587–1605, Nov 2007. (cited on page 16)
5. K. Andersen. *The Geometry of an Art: The History of the Mathematical Theory of Perspective from Alberti to Monge*. Sources and Studies in the History of Mathematics and Physical Sciences. Springer-Verlag New York, 1 edition, 2008. (cited on page 15)
6. F. Becker, F. Lenzen, J. Kappes, and C. Schnörr. Variational recursive joint estimation of dense scene structure and camera motion from monocular high speed traffic sequences. *International Journal of Computer Vision*, 105:269–297, 07 2013. (cited on pages 2 and 11)
7. F. Bergamasco, A. Albarelli, L. Cosmo, A. Torsello, E. Rodolà, and D. Cremers. Adopting an unconstrained ray model in light-field cameras for 3d shape reconstruction. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3003–3012, June 2015. (cited on page 65)
8. J. Berger. *Second Order Minimum Energy Filtering of Joint Variational Camera Motion and Depth Map Reconstructions*. PhD thesis, 01 2017. (cited on pages 2 and 11)
9. Y. Bok, H. G. Jeon, and I. S. Kweon. Geometric calibration of micro-lens-based light field cameras using line features. *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, 39(2):287–300, Feb 2017. (cited on pages 4, 41, 43, 47, 65, 67, 68, 69, 72, 73, 74, 75, and 76)

10. S. Bonnabel, P. Martin, and P. Rouchon. Non-linear symmetry-preserving observers on lie groups. *IEEE Transactions on Automatic Control*, 54(7):1709–1713, July 2009. (cited on page 10)
11. C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, Dec 2016. (cited on pages 6, 11, and 14)
12. R. J. Campbell and P. J. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81(2):166 – 210, 2001. (cited on page 16)
13. E. Candès, L. Demanet, D. Donoho, and L. Ying. Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, 5(3):861–899, 2006. (cited on pages 109 and 114)
14. E. Candes, L. Demanet, D. Donoho, and L. Ying. Curvelab. http://www.curvelet.org/, 2015. Accessed: 2019-11-01. (cited on page 111)
15. X. Chen and H. Kano. A new state observer for perspective systems. *IEEE Transactions on Automatic Control*, 47(4):658–663, Apr 2002. (cited on page 10)
16. X. Chen and H. Kano. State observer for a class of nonlinear systems and its application to machine vision. *IEEE Transactions on Automatic Control*, 49(11):2085–2091, Nov 2004. (cited on page 10)
17. O. Christensen. *An Introduction to Frames and Riesz Bases*. Birkhauser Verlag GmbH, Springer, 2016. (cited on page 109)
18. B. Cyganek and J. P. Siebert. *An Introduction to 3D Computer Vision Techniques and Algorithms*. Wiley, 1 edition, 2009. (cited on page 6)
19. O. Dahl, F. Nyberg, and A. Heyden. Nonlinear and adaptive observers for perspective dynamic systems. In *2007 American Control Conference*, pages 966–971, July 2007. (cited on page 10)
20. D. Dansereau and L. Bruton. Gradient-based depth estimation from 4d light fields. In *IEEE Proceedings of the International Symposium on Circuits and Systems*, volume 3, pages 549–552, 2004. (cited on pages 38 and 82)
21. D. G. Dansereau, B. Girod, and G. Wetzstein. Liff: Light field features in scale and depth. *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8034–8043, 2019. (cited on page 122)

22. D. G. Dansereau, O. Pizarro, and S. B. Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1027–1034, June 2013. (cited on pages 4, 38, 47, 66, 67, 73, 74, 75, and 76)

23. L. C. Evans. *Partial Differential Equations*. Graduate Studies in Mathematics 19. American Mathematical Society, 1998. (cited on page 62)

24. A. Filippov. *Differential Equations with Discontinuous Righthand Sides*. Springer, 1 edition, 1988. (cited on page 89)

25. J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *The Artificial Intelligence Review*, 43(1):55–81, 01 2015. Copyright - Springer Science+Business Media Dordrecht 2015; Last updated - 2018-10-06. (cited on page 6)

26. T. Georgiev and A. Lumsdaine. Depth of field in plenoptic cameras. In *Eurographics*, 2009. (cited on pages 3 and 15)

27. A. Gershun. The light field. *Journal of Mathematics and Physics*, 18(1-4):51–151, 1939. (cited on pages xiii, 14, and 15)

28. I. Grave and Y. Tang. A new observer for perspective vision systems under noisy measurements. *IEEE Transactions on Automatic Control*, 60(2):503–508, Feb 2015. (cited on pages 2 and 10)

29. C. Hahne. The standard plenoptic camera. http://www.plenoptic.info/, 2018. (cited on page 68)

30. X.-F. Han, H. Laga, and M. Bennamoun. Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era. 06 2019. (cited on page 9)

31. R. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22:125–140, 03 1997. (cited on page 13)

32. R. Hartley and R. Gupta. Linear pushbroom cameras. In *Proceedings of the Third European Conference on Computer Vision*, volume 800, pages 555–566, 05 1994. (cited on page 13)

33. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision, 2nd Edition*. 2 edition, 2004. (cited on pages 12 and 38)

34. R. I. Hartley. Minimizing algebraic error. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 356:1175 – 1192, 1998. (cited on page 7)

35. C. Heinze, S. Spyropoulos, S. Hussmann, and C. Perwass. Automated robust metric calibration algorithm for multifocus plenoptic cameras. *IEEE Transactions*

on Instrumentation and Measurement, 65(5):1197–1205, May 2016. (cited on pages 65 and 68)

36. M. Hirsch. On imbedding differentiable manifolds in Euclidean space. *Annals of Mathematics*, 73(3):566–571, 1961. (cited on page 52)
37. K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. pages 19–34, 03 2017. (cited on pages 5, 38, and 39)
38. B. K. Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. Technical report, USA, 1970. (cited on page 8)
39. D. P. Huttenlocher, G. A. Klanderman, and W. A. Rucklidge. Comparing images using the hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(9):850–863, Sept. 1993. (cited on page 25)
40. F. E. Ives. Parallax stereogram and process of making same., 1903. (cited on pages xiii, 14, and 15)
41. R. E. Jacobson, N. Axford, S. Ray, and G. G. Attridge. *Manual of Photography: Photographic and Digital Imaging*. Butterworth-Heinemann, Newton, MA, USA, 9th edition, 2001. (cited on page 42)
42. H. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y. Tai, and I. S. Kweon. Accurate depth map estimation from a lenslet light field camera. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1547–1555, June 2015. (cited on page 38)
43. Z. Ji, C. Zhang, and Q. Wang. Light field camera self-calibration and registration. *SPIE Photonics Asia*, 10020:10, 2016. (cited on pages 65 and 67)
44. O. Johannsen, C. Heinze, B. Goldluecke, and C. Perwaß. *On the Calibration of Focused Plenoptic Cameras*, pages 302–317. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. (cited on pages 65 and 67)
45. O. Johannsen, K. Honauer, B. Goldluecke, A. Alperovich, F. Battisti, Y. Bok, M. Brizzi, M. Carli, G. Choe, M. Diebold, M. Gutsche, H.-G. Jeon, I. Kweon, J. Park, J. Park, H. Schilling, H. Sheng, L. Si, M. Strecke, and H. Zhu. A taxonomy and evaluation of dense light field depth estimation algorithms. pages 1795–1812, 07 2017. (cited on pages 5 and 38)
46. O. Johannsen, A. Sulc, and B. Goldluecke. What sparse light field coding reveals about scene structure. pages 3262–3270, 06 2016. (cited on page 38)
47. M. Kazhdan. Reconstruction of solid models from oriented point sets. pages 73–82, 01 2005. (cited on pages 2 and 103)

48. M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. (cited on pages 2, 3, and 103)
49. J. J. Kazimierz Kuratowski. *Topology*, volume vol.1. Academic Pr / PWN, revised edition, 1966. (cited on page 26)
50. J. Keshavan, H. Escobar-Alvarez, and J. S. Humbert. An adaptive observer framework for accurate feature depth estimation using an uncalibrated monocular camera. *Control Engineering Practice*, 46:59 – 65, 2016. (cited on pages 2 and 10)
51. A. Khatamian and H. Arabnia. Survey on 3d surface reconstruction. *Journal of Information Processing Systems*, 12:338–357, 01 2016. (cited on page 16)
52. Kowa. *LM35SC Data Sheet*, 2018. (cited on page 73)
53. K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 307–314 vol.1, Sep. 1999. (cited on pages 9 and 18)
54. C. Lageman, J. Trumpf, and R. Mahony. Gradient-like observers for invariant dynamics on a lie group. *IEEE Transactions on Automatic Control*, 55(2):367–377, Feb 2010. (cited on page 10)
55. A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, Feb 1994. (cited on page 9)
56. M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 31–42, New York, NY, USA, 1996. ACM. (cited on pages xiii, 4, 13, 14, 15, and 38)
57. G. Lippmann. Épreuves réversibles donnant la sensation du relief. *J. Phys. Theor. Appl.*, 7(1):821–825, 1908. (cited on pages xiii, 14, and 15)
58. H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828), 1981. (cited on pages 7 and 12)
59. W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, Aug. 1987. (cited on page 17)
60. P. Lourenco, B. Guerreiro, P. Batista, P. Oliveira, and C. Silvestre. Simultaneous localization and mapping for aerial vehicles: a 3-d sensor-based gas filter. *Autonomous Robots*, 40, 09 2015. (cited on pages 2 and 10)

61. A. Lumsdaine and T. Georgiev. The focused plenoptic camera. In *2009 IEEE International Conference on Computational Photography*, pages 1–8, 2009. (cited on pages xiii, 14, and 66)

62. R. Mahony and T. Hamel. A geometric nonlinear observer for simultaneous localisation and mapping. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2408–2415, Dec 2017. (cited on pages 11, 120, 122, and 123)

63. R. Maier, K. Kim, D. Cremers, J. Kautz, and M. Nießner. Intrinsic3d: High-quality 3d reconstruction by joint appearance and geometry optimization with spatially-varying lighting. 10 2017. (cited on page 111)

64. R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, Feb 1995. (cited on page 18)

65. J. Manson, G. Petrova, and S. Schaefer. Streaming surface reconstruction using wavelets. *Computer Graphics Forum*, 27(5):1411–1420, 2008. (cited on page 103)

66. D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, 1976. (cited on page 7)

67. MathWorks. *When the Solver Fails*, 2018. (cited on page 75)

68. L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–238, 1989. (cited on page 11)

69. P. McManamon. *Field Guide to Lidar*. SPIE Field Guide FG36. SPIE Press, 2015. (cited on page 14)

70. L. M. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2018. (cited on pages 9 and 103)

71. A. Meydenbauer. Die photometrographie. *Wochenblatt des Architektenvereins zu Berlin*, 1867. (cited on page 7)

72. R. A. Morano, C. Ozturk, R. Conn, S. Dubin, S. Zietz, and J. Nissano. Structured light using pseudorandom codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):322–327, March 1998. (cited on page 13)

73. A. Mustafa, H. Kim, J. Guillemaut, and A. Hilton. Temporally coherent 4d reconstruction of complex dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4660–4669, June 2016. (cited on page 104)

74. 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*, 2(11):144–162, 2005. (cited on pages xiii, 3, 14, 15, and 68)

75. S. Nousias, F. Chadebecq, J. Pichat, P. Keane, S. Ourselin, and C. Bergeles. Corner-based geometric calibration of multi-focus plenoptic cameras. In *IEEE International Conference on Computer Vision*, 2017. (cited on pages 4, 41, 65, 66, 67, 68, 69, 73, 74, 75, and 76)

76. S. G. P. O'Brien, K. Ashton, and J. Trumpf. An observer for infinite dimensional 3d surface reconstruction that converges in finite time. In *21st IFAC World Congress , Year = 2020*. (cited on pages 6, 33, and 103)

77. S. G. P. O'Brien, J. Trumpf, V. Ila, and R. Mahony. A geometric observer for scene reconstruction using plenoptic cameras. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 557–564, Dec 2018. (cited on pages 6, 32, and 81)

78. S. G. P. O'Brien, J. Trumpf, V. Ila, and R. Mahony. Estimation and geometry of disparity-fields from raw light-fields. (Preprint), 2019. (cited on pages 6, 32, 37, and 39)

79. S. G. P. O'Brien, J. Trumpf, V. Ila, and R. E. Mahony. Calibrating light-field cameras using plenoptic disc features. *2018 International Conference on 3D Vision (3DV)*, pages 286–294, 2018. (cited on pages 6, 32, 37, 39, and 65)

80. S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Applied Mathematical Sciences 153. Springer-Verlag New York, 1 edition, 2003. (cited on pages 3 and 18)

81. O. Ozyesil, V. Voroninski, R. Basri, and A. Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017. (cited on page 6)

82. J. Peng, Z. Xiong, D. Liu, and X. Chen. Unsupervised depth estimation from light field using a convolutional neural network. *2018 International Conference on 3D Vision (3DV)*, pages 295–303, 2018. (cited on page 3)

83. C. Perwass and L. Wietzke. Single lens 3d-camera with extended depth-of-field. 8291:4–, 02 2012. (cited on pages 3 and 38)

84. H. Pfister, M. Zwicker, J. Baar, and M. Gross. Surfels: Surface elements as rendering primitives. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 05 2000. (cited on page 17)

85. L. Piegl and W. Tiller. *The NURBS Book (2nd Ed.)*. Springer-Verlag, Berlin, Heidelberg, 1997. (cited on page 17)

86. J. W. Polderman and J. C. Willems. *Introduction to Mathematical Systems Theory: A Behavioral Approach*. Springer-Verlag, Berlin, Heidelberg, 1997. (cited on page 29)

87. M. Potmesil and I. Chakravarty. Synthetic image generation with a lens and aperture camera model. *ACM Trans. Graph.*, 1(2):85–108, Apr. 1982. (cited on

page 42)

88. E. Prados and O. Faugeras. *Shape From Shading*, pages 375–388. Springer US, Boston, MA, 2006. (cited on page 8)

89. F. Ramos and L. Ott. Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent. *The International Journal of Robotics Research*, 35:1717–1730, 12 2016. (cited on page 104)

90. P. Rander, P. J. Narayanan, and T. Kanade. Virtualized reality: constructing time-varying virtual worlds from real world events. In *Proceedings. Visualization '97 (Cat. No. 97CB36155)*, pages 277–283, Oct 1997. (cited on page 13)

91. A. Saccon, J. Trumpf, R. Mahony, and A. P. Aguiar. Second-order-optimal minimum-energy filters on lie groups. *IEEE Transactions on Automatic Control*, 61(10):2906–2919, Oct 2016. (cited on page 11)

92. M. Salzmann and P. Fua. *Deformable surface 3D reconstruction from monocular images*. Synthesis Lectures on Computer Vision. Morgan, 2010. (cited on page 6)

93. M. R. U. Saputra, A. Markham, and N. Trigoni. Visual slam and structure from motion in dynamic environments: A survey. *ACM Comput. Surv.*, 51(2), Feb. 2018. (cited on page 6)

94. R. Senanayake and F. Ramos. Building continuous occupancy maps with moving robots. In *AAAI*, 02 2018. (cited on page 104)

95. L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, 2001. (cited on page 16)

96. C. Shin, H.-G. Jeon, Y. Yoon, I. Kweon, and S. Joo Kim. Epinet: A fully-convolutional neural network using epipolar geometry for depth from light field images. 04 2018. (cited on pages 3, 38, and 39)

97. L. Si and Q. Wang. Dense depth-map estimation and geometry inference from light fields via global optimization. In *Computer Vision – ACCV 2016*, pages 83–98. Springer International Publishing, 2017. (cited on page 38)

98. J.-L. Starck, E. J. Candes, and D. L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):670–684, 2002. (cited on page 112)

99. M. Strecke, A. Alperovich, and B. Goldluecke. Accurate depth and normal maps from occlusion-aware focal stack symmetry. pages 2529–2537, 07 2017. (cited on page 38)

100. K. H. Strobl and M. Lingenauber. Stepwise calibration of focused plenoptic cameras. *Computer Vision and Image Understanding*, 145:140 – 147, 2016. (cited on pages 65 and 67)

101. J. Sun, C. Xu, B. Zhang, S. Wang, M. M. Hossain, H. Qi, and H. Tan. Geometric calibration of focused light field camera for 3-d flame temperature measurement. In *IEEE International Instrumentation and Measurement Technology Conference*, pages 1–6, 2016. (cited on pages 65 and 67)

102. R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010. (cited on page 17)

103. T. Taketomi, H. Uchiyama, and S. Ikeda. Visual slam algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):16, Jun 2017. (cited on page 6)

104. S. Thrun. Learning occupancy grid maps with forward sensor models. *Auton. Robots*, 15(2):111–127, 2003. (cited on page 13)

105. S. Thrun and A. Bü. Integrating grid-based and topological maps for mobile robot navigation. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, pages 944–950. AAAI Press, 1996. (cited on pages 3, 9, 13, 18, and 104)

106. C. Toth and G. Jóźków. Remote sensing platforms and sensors: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115:22 – 36, 2016. Theme issue 'State-of-the-art in photogrammetry, remote sensing and spatial information science'. (cited on page 11)

107. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment — a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, pages 298–372, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. (cited on page 8)

108. J. Trumpf, H. L. Trentelman, and J. C. Willems. Internal model principles for observers. *IEEE Transactions on Automatic Control*, 59(7):1737–1749, July 2014. (cited on page 30)

109. P. van Goor, R. Mahony, T. Hamel, and J. Trumpf. A geometric observer design for visual localisation and mapping. *Proceedings of the 58th IEEE Conference on Decision and Control (CDC)*, pages 2543–2549, 2019. (cited on page 11)

110. T. Wang, A. A. Efros, and R. Ramamoorthi. Occlusion-aware depth estimation using light-field cameras. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3487–3495, Dec 2015. (cited on page 38)

111. Y. Wang and A. F. Lynch. A block triangular form for nonlinear observer design. *IEEE Transactions on Automatic Control*, 51(11):1803–1808, Nov 2006. (cited on page 10)

112. S. Wanner and B. Goldlücke. Globally consistent depth labeling of 4d light fields. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 41–48, 2012. (cited on page 38)

113. S. Wanner, S. Meister, and B. Goldluecke. Datasets and benchmarks for densely sampled 4d light fields. In M. M. Bronstein, J. Favre, and K. Hormann, editors, *VMV*, pages 225–226. Eurographics Association, 2013. (cited on page 39)

114. R. A. Wijsman. Convergence of sequences of convex sets, cones and functions. *Bull. Amer. Math. Soc.*, 70(1):186–188, 01 1964. (cited on page 25)

115. B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM Trans. Graph.*, 24(3):765–776, July 2005. (cited on page 13)

116. B. S. Wilburn, M. Smulski, H.-H. K. Lee, and M. A. Horowitz. The light field video camera. In S. Panchanathan, V. M. B. Jr., and S. I. Sudharsanan, editors, *Media Processors 2002*, volume 4674, pages 29 – 36. International Society for Optics and Photonics, SPIE, 2001. (cited on page 13)

117. Williem, I. K. Park, and K. M. Lee. Robust light field depth estimation using occlusion-noise aware data costs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2484–2497, Oct 2018. (cited on page 38)

118. A. P. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17(1):17 – 45, 1981. (cited on page 9)

119. C. Wöhler. *3D computer vision: efficient methods and applications*. X.media.publishing. Springer Berlin Heidelberg, 1 edition, 2009. (cited on page 16)

120. R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139 – 144, 1980. (cited on page 8)

121. G. Wright. Radial basis functions for scientific computing, 2014. (cited on page 88)

122. J. C. Yang, M. Everett, C. Buehler, and L. McMillan. A real-time distributed light field camera. In *Proceedings of the 13th Eurographics Workshop on Rendering*, EGRW '02, page 77–86, Goslar, DEU, 2002. Eurographics Association. (cited on page 13)

123. L. Ying, L. Demanet, and E. Candes. 3d discrete curvelet transform. pages 351 – 361. International Society for Optics and Photonics, SPIE, 2005. (cited on page

109)

124. N. Zarrouati-Vissiere, E. Aldea, and P. Rouchon. So(3)-invariant asymptotic observers for dense depth field estimation based on visual data and known camera motion. *2012 American Control Conference (ACC)*, pages 4116–4123, 2012. (cited on pages 2 and 11)
125. N. Zeller, F. Quint, and U. Stilla. Depth estimation and camera calibration of a focused plenoptic camera for visual odometry. *Journal of Photogrammetry and Remote Sensing*, 118:83 – 100, 2016. (cited on pages 65 and 67)
126. N. Zeller, F. Quint, and U. Stilla. From the calibration of a light-field camera to direct plenoptic odometry. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):1004–1019, 2017. (cited on pages 65 and 67)
127. C. Zhang and T. Chen. A self-reconfigurable camera array. In *ACM SIGGRAPH 2004 Sketches*, SIGGRAPH '04, page 151, New York, NY, USA, 2004. Association for Computing Machinery. (cited on page 13)
128. C. Zhang, Z. Ji, and Q. Wang. Decoding and calibration method on focused plenoptic camera. *Computational Visual Media*, 2(1):57–69, 2016. (cited on pages 65 and 68)
129. S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong. Robust depth estimation for light field via spinning parallelogram operator. *Comput. Vis. Image Underst.*, 145(C):148–159, Apr. 2016. (cited on page 38)
130. D. Zlotnik and J. Forbes. Gradient-based observer for simultaneous localization and mapping. *IEEE Transactions on Automatic Control*, PP:1–1, 04 2018. (cited on page 10)