Learning Depth and Visual Odometry From Light Fields

School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

Joseph Daniel

May, 2020

Abstract

The meteoric rise of mobile robotics has seen traditionally difficult-to-navigate environments like the home, the road and the ocean, become standard operating environments for autonomous machines. This rapid advancement has culminated in an increasing need for accurate robotic navigation, where the pinnacle of autonomy is the ability to operate adaptively in unconstrained environments. Central to this capability is the ability to perceive motion - a task that has recently drawn considerable attention from those in the computer vision community and given rise to a family of algorithms called *visual odometry*.

The primary contribution of this work is a novel, data-driven pipeline for visual odometry and depth perception, which combines recent successes in plenoptic imaging with the pattern recognition capabilities of convolutional networks. Notably, we formulate our algorithm as an *unsupervised* learning problem, meaning neither ground-truth odometry, nor depth is used to train our pipeline. The self-supervising nature of our algorithm equips it with a robustness to effects like thermal expansion and shock, as it is capable of adapting and learning from experience, even whilst operating in situ.

We trained and validated our algorithm on a dataset collected as part of this thesis project, using a robotic arm with a known kinematic model for ground-truth validation data. Finally, we demonstrate that our algorithm outperforms existing state-of-the-art monocular approaches, demonstrating the capabilities of plenoptic imaging for robotic perception.

Statement of Contribution

- I carried out the literature review in order to evaluate existing algorithms and approaches.
- I authored the data acquisition software and collected the dataset of light field video used in this work.
- With the help of my supervisor Dr. Donald Dansereau, I designed and implemented the novel algorithms described in this work.
- I augmented existing code from Zhou *et al.* [62], authoring custom components for data-loading, loss computations, and performing inference.
- I carried out the experiments described in this work using the proposed algorithms.
- I authored the functions for analysing experimental results, and conducted the analysis myself, with advice provided by my supervisor.
- I carried out the discussion and conclusion, which are my own, influenced by discussion with my supervisor.

Joseph Thomas Daniel

27th May, 2020

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Donald Dansereau, for his guidance throughout this thesis. His intellect and motivation has inspired my interest in robotics and computational imaging, and his perceptiveness has at all times kept me thinking critically. My thanks also to my colleagues in the school of AMME, who have provided consistently valuable feedback in our weekly roundtable discussions and made this thesis all the more enjoyable.

I would like to also acknowledge my family and friends who have supported me during this thesis, whether through their heartening encouragement or simple acts of homemade food delivery. Finally, a special thanks to Josephine for her patience and support in this time of physical isolation - she has consistently shown her compassion and kindness, even though the apartment we share certainly wasn't designed for two to be working from home.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Humans have a remarkable ability to perceive visual stimuli emanating from the world around them. Not only do we identify different objects, scene depth, movement, and colour with ease, but visual scenes might even elicit enjoyment and meaning in our lives. Man-made machines on the other hand are much more easily decomposed into a set of deterministic modules that require explicit, well defined instruction sets. This thesis is concerned with developing a set of algorithms that breaks down, and utilises the wealth of information in image data to produce valuable signals that can be employed in robotic perception.

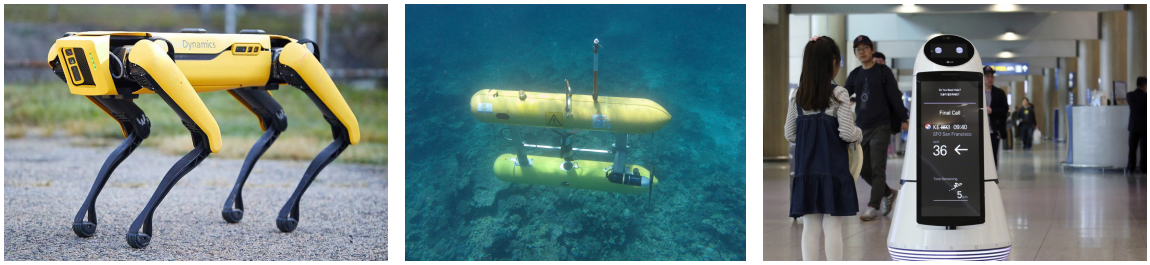
1.1 Motivation

1.1.1 Visual Odometry and Robotic Perception

True autonomy in mobile robotics requires the ability to perceive the world adequately. In particular, understanding the structure and size of the space inhabited by the robot, as well as its own movement through that space are vital to its ability to navigate and interact with the world. Computer vision algorithms in combination with increasingly lower-cost and higher quality imaging hardware offers a popular solution, encompassing enormous diversity in sensing modalities and precipitating the development of powerful perception systems in modern robotics. Plenoptic imaging is one such expression of evolving imaging technologies that continues to yield promising results in tasks such as mapping [33], underwater imaging [52], low light imaging [14], and classification [55].

This work is concerned with the application of **plenoptic imaging** in two intimately related

tasks: **visual odometry** which involves estimating the motion of a camera in 3D space, and **depth reconstruction** which estimates the shape of the scene in an image. We motivate a need for these algorithms by examining one of the most important challenges currently being tackled in robotics: localisation and navigation. A robot's locale is a vital piece of information required to effectively plan paths and navigate through any environment. The challenge however, is that localisation typically requires a map of the space. In an unknown environment where a map doesn't exist, the robot must simultaneous tackle the problems of reconstructing the environment's geometry, and localising itself within that geometry. This is a challenge known as SLAM (Simultaneous Localisation and Mapping), and has proven a challenging robotics problem for at least 30 years [8, 33]. Odometry is a crucial component of even the most sophisticated SLAM algorithms - Figure 1.1 illustrates some of the mobile systems motivating a need for robust navigation. In this work, odometry is treated as a problem which can be tackled with visual perception and computer vision algorithms.



(a) Spot (Boston Dynamics) **(b)** AUV Sirius (ACFR) **(c)** Troika (LG)

Figure 1.1: Autonomous mobile robots are increasingly operating in unconstrained environments, characterised by uneven terrain, weak GPS signal or crowds of moving people. An important challenge for the broader adoption of these types of mobile robots, is navigational capabilities, which rely foundationally on mapping and localisation algorithms.

One might question the practicality of using image data for these tasks when great success has been found with more specialised sensing modalities; motion estimation for example is typically addressed with inertial sensor measurements or Global Positioning System (GPS) receivers. Other robotic systems employ lidar, acoustic range-finding, time-of-flight cameras or structured-light cameras to gain access to 3D models of the environment. Cameras however, can often offer superior tradeoffs in size, weight, cost, power consumption, and they deliver a rich, highly detailed representation of the world. Furthermore, unlike *active* sensor technologies which sense the world by 'illuminating' it, cameras are *passive* sensors, meaning they do not interfere with one another and

can be employed in a more diverse range of environments. Thanks to their ability to adjust their exposure period, aperture size and sensor gain, cameras are also highly capable in a variety of lighting conditions and environments.

Furthermore, the applications of visual odometry extend beyond the context of SLAM algorithms. For example, a multi-rotor Unmanned Aerial Vehicle (UAV) can fuse measurements from inertial and visual measurements to hover in place in GPS denied environments. Outside of the field of robotics, Augmented Reality (AR) applications are employing visual odometry and SLAM to render virtual 3D models in real time on top of camera feeds. Proença [47] envisages an AR application that employs visual odometry to guide an operator through a contaminated nuclear facility, whilst rectifying existing 3D models of the facility. The adversities of installing guidance and navigation infrastructure in such facilities can be extrapolated to cases such as planetary rovers and Remotely Operated Underwater Vehicles (ROVs).

1.1.2 Computational Imaging

Visual perception is attractive for robotics for many reasons, however, as with any kind of system design, the tradeoffs need to be considered. Digital cameras accumulate photons at each pixel, forming an image after a brief period of 'exposure' - the longer the exposure, the brighter the image. However, cameras that move - as robots often do - are likely to accumulate motion blur. Alternatively, a larger aperture accumulates a larger number of photons, but gives rise to shallow depth-of-field - an aesthetically pleasing effect in photography, but a limitation for computer vision algorithms.

Computational photography unifies the design of optics, algorithms, sensors and illumination to capture better photographs. Modern smartphones, as an example, have relatively slow lenses, small sensors, and cheap optics. DSLR and mirrorless cameras on the other hand take advantage of higher grade optics and larger sensors - which might lead us to expect better image quality compared to phone cameras. Modern smartphones however benefit from powerful CPUs, hardware-accelerated graphics, and an entire community of software engineers working on algorithms. Many of the effects that were once unique to expensive imaging devices can now be replicated on smartphones. Low light imaging is enhanced using algorithms such as Google's 'Night Sight' [37] to improve clarity and colour accuracy. Shallow depth-of-field, an effect that is typically difficult to capture on a wide lens with narrow apertures is now replicated using 'Portrait Mode' [53].

In this work, we embrace computational imaging as a modern approach to visual odometry.

Humans are exceptionally well adapted to tasks involving visual motion and depth perception - our two eyes let us process the 3D geometry of a scene, while our learned experiences are often able to fill in the gaps where geometric information is insufficient or unavailable. Image processing algorithms are not equipped with this same kind of human intuition, and so the deceptively complex task of estimating the 3D structure of a scene from a sequence of images continues to attract attention from the computer vision community [13, 19, 45, 61].

The price of camera components is decreasing while image quality continues to improve, not only making cameras an attractive perception module for autonomous robotics, but also spurring the popularity of multiple-view imaging. Embracing this idea, this work adopts vocabulary from the literature on light field imaging, applying this promising imaging technology to the tasks of visual odometry and depth perception. More broadly however, the principle underpinning much of this thesis is that novel imaging devices that break away from the traditional pinhole model of the camera have vast implications in the field of machine vision. We need not look any further than the animal kingdom to see why this is true - the biological eye is estimated to have evolved independently no fewer than 50 times, each variant acutely adapted to the particular set of challenges in their environment. The diversity and evolutionary ingenuity in biological visual perception systems prompts an important question in robotics and computer vision - how best should we equip robots to see the world given a particular set of challenges and environments? In this thesis, a camera array which is a simple yet versatile extension of the stereo camera is used to develop the algorithms that address the visual odometry and depth reconstruction problems.

1.2 Problem Statement

There is a well established library of solutions addressing the visual odometry and depth reconstruction problems. Some use closed form solutions that directly solve for geometry and ego-motion, while others, like this work utilise a data-driven approach to indirectly model visual odometry. Throughout this work, we will refer to these as the *geometric* and *data-driven* approaches, both of which form a crucial part of our survey of existing state-of-the-art methods

This thesis tackles the problem of estimating visual odometry and depth using an *uncalibrated* camera array - a challenge that, to the best of our knowledge, is absent in existing state-of-the-art methods. Robots operating in unconstrained environments are typically exposed to destabilising effects including thermal expansion, vibration and shock - effects which might render the calibrations

of camera arrays invalid. Multiple-view geometry, which is harnessed by the *geometric* family of algorithms, is materially dependent on knowing the orientation and position of the cameras; rotating a camera even a few arc-seconds from its calibrated position results in a disproportionally magnified error in the image being formed.

Existing *data-driven* approaches on the other hand, have typically used monocular or stereo imagery, overlooking the richness of light field imagery as a possibility for improving robustness and accuracy. Within the data-driven family of algorithms, a relatively new *unsupervised* approach has emerged, cleverly piecing together the available information to learn visual odometry from raw footage alone. The unsupervised nature of this approach means that the model calibrates itself in a fashion, learning from raw data to produce ego-motion predictions. Unsupervised machine learning makes 'online-learning' possible, in which models take advantage of new data becoming available in-situ, improving performance and building robustness to different environments. Most importantly however, unsupervised models are able to adapt to errors in the calibration of the equipment without requiring hand-labelled data to supervise the learning process. A self-calibrating and adaptive perception module which is robust to adverse effects is an attractive capability in robotics. Once again however, existing unsupervised algorithms have not been applied to light field data, prompting a natural, yet important next step which is the subject of this work: combining the richness of light field data with the self-calibrating capabilities of unsupervised learning algorithms to perform visual odometry.

1.3 Contributions

This work builds on existing data-driven approaches, suggesting an adaptation of conventional 2D machine learning algorithms to enable unsupervised learning on light field data. Specifically, we make the following contributions:

- We adapt 2D unsupervised algorithms for learning depth and pose, suggesting a generalisation of existing algorithms that enables them to operate on arrays of cameras.
- Through this adaptation, we show that our approach is not only scale-consistent, but also grounded in metric depth and pose measurements.
- We suggest three encodings of the light field which are well suited to 2D convolutions, making them convenient for ingestion using convolutional neural networks.

- We demonstrate that our algorithm outperforms existing state-of-the-art approaches using monocular imagery, reinforcing the benefits of light field technology in robotic imaging.

1.4 Outline

Concepts from computer vision, machine learning and light field imaging are used heavily throughout this thesis, and **Chapter 2** begins by providing an overview of the relevant background information. On the subject of computer vision, it describes the pinhole model of the camera, the fundamentals of multiple view geometry, and develops intuition relevant to light field imaging. Machine learning is presented as an approach to solving computer vision problems, with emphasis directed towards convolutional neural networks, as an algorithm that continues to gain popularity in image based problems.

The existing approaches for performing visual odometry and depth estimation are reviewed in **Chapter 3**, presented in two categories: geometric approaches which rely on directly modeling movement through 3D space, and machine learning approaches which take advantage of massive datasets to build resilient, data-driven solutions.

Chapter 4 introduces a novel light-field based algorithm for simultaneously learning depth and pose. We begin by describing a simple extension of existing monocular approaches, then proceed to layer-in modifications that enforce scale-consistency, and metric pose estimates, taking advantage of 4D light field data. We also describe our experimental setup, including our data acquisition strategy using a robotic manipulator to collect a suitable 4D video dataset.

In **Chapter 5** we presents results from our two proposed algorithmic pipelines, and compare these to state-of-the-art existing methods. We demonstrate that our approach outperforms monocular approaches in both depth and pose estimation. Furthermore, we show results for our experiments with three light field encodings, with the intention of empirically comparing the effectiveness of different input methods.

An in-depth discussion on the successes and challenges in this work is provided in **Chapter 6**, with specific detail on the effectiveness of our two pipelines and three input methods.

Finally, we conclude this work in **Chapter 7**, drawing conclusions from our experimental results on the promise of light field and deep learning technologies, and suggest some potential applications and future research questions to continue the spirit of our work.

Chapter 2

Background

2.1 Geometry in Computer Vision

The pixel is the building block of digital imagery. Thanks to our ability to fabricate advanced circuits on the scale of nanometers, digital camera sensors have become ubiquitous - making their way into all manner of consumer devices. Because the photodiodes in camera sensors are typically arranged on a 2D plane, most of our operations on digital images manipulate the 2D structures present in that pixel data. However, it is important to remember that the 2D shapes and structures that appear on the sensor plane are projected versions of a 3D world.

2.1.1 The Pinhole Camera

The most basic kind of camera, the pinhole camera projects a 3D world point $[X, Y, Z]$ to a homogenous coordinate on a 2D plane $[u, v, w]$ [27]. This relationship is captured in the 3×3 camera intrinsics matrix and is typically expressed as

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{2.1}$$

The intrinsics matrix can alternatively be thought of as an expression that maps ray directions to pixel coordinates, as shown in Figure 2.1:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} tan(\theta) \\ tan(\phi) \\ 1 \end{bmatrix}. \tag{2.2}$$

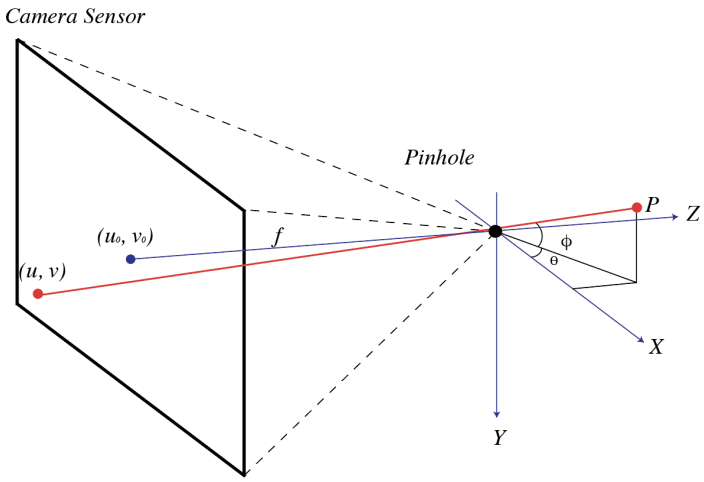


Figure 2.1: The pinhole model describes how a light ray originating from world coordinate P is received at pixel coordinate (u, v). Equivalently, the pinhole model can be thought of as associating the light ray of elevation ϕ and azimuth θ with the specific pixel coordinate (u, v). Since the relationship is a one-to-one mapping between ray directions and pixel coordinates, the inverse of the intrinsics matrix K^{-1} associates each pixel with a corresponding ray direction.

2.1.2 Epipolar Geometry and the Fundamental Matrix

Epipolar geometry describes the relationship between two cameras, and imposes a set of constraints which makes it possible for us to draw meaningful geometric information when we have two views of the same scene. This relationship is encapsulated in a 3×3 matrix called the Fundamental matrix F. The culmination of the epipolar constraint is that any point in 3 dimensional space that appears at pixel $[u_1, v_1, 1]$ in one view and $[u_2, v_2, 1]$ in the second view must satisfy the relationship:

$$\begin{bmatrix} u_1, v_1, 1 \end{bmatrix}^T F \begin{bmatrix} u_2, v_2, 1 \end{bmatrix} = 0. \tag{2.3}$$

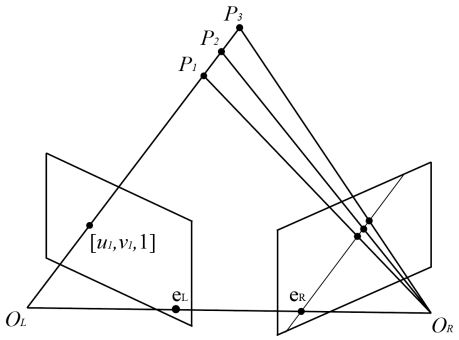


Figure 2.2: Epipolar geometry is the geometry of stereo camera pairs. Without knowing depth, the point $[u, v, 1]$ can be projected to any number of points $P1, P2, P3...$ in 3D space, having many possible projections on the sensor plane of the right camera. The Fundamental matrix however constrains the set of possible projections to lie on a straight line called the epipolar line.

The Fundamental matrix thus describes the rotation and translation between two cameras up to scale. Any point projected from pixel coordinate $[u_1, v_1, 1]$ must lie on the epipolar line in the second image, which is formed by the intersection of the epipolar plane and the imaging plane.

2.2 Visual Odometry

An important problem tackled in this thesis, *visual odometry* describes a type of problem in computer vision where we seek to estimate the relative motion of a camera by comparing two photographs taken at two different points in space. We also refer to this challenge as *pose estimation* throughout this text, as the task effectively determines the pose of a rigid body as it moves through space, relative to its start position. Visual odometry is typically described as as a 6 degree-of-freedom problem, because we seek to estimate not only three translational components, but also three rotational components. As we will see in the literature review in Chapter 3, there is a well established library of solutions to the visual odometry problem, however it is far from being a 'solved' problem, and continues to draw attention from the computer vision community. To motivate our design of a visual solution to the pose estimation problem, we first examine the existing non-visual solutions.

One of the simplest methods of odometry estimation, is with the use of wheel motor encoders. In these cases, we use knowledge of the system's geometry to compute the motion of the robot as its wheels are spun. This simple idea is frequently extended to other modes of robotic locomotion

- legged robots, and even hovering quad-copters can use dynamical models to estimate system responses to certain control inputs. These odometry methods are often simple *feedforward* models, they do not make use of *feedback* data to discern motion. Wheels often slip and quad-copters frequently face gusts of wind - scenarios that simple feedforward models cannot predict.

By layering in sensor data such as a GPS, we can employ a *state observer* model to combine our knowledge of system dynamics with feedback data to build an even better picture of the system's state. Even with the use of observers however, we might still find scenarios that challenge our odometry system - for example we might find that GPS accuracy suffers in poor weather conditions. By layering in even more sources of feedback data, such as an inertial sensor, we can combine several sources of truth through *sensor fusion*, to build the best probabilistic picture of the robot's state. Thus, where some sensors may fail, other sensors might take over. Our proposed visual odometry system is therefore designed as one additional source of truth that might be used in robot state estimation.

In Chapter 3, we will describe the current state-of-the-art in visual pose estimation, using two broad families of algorithms - those that directly model the motion of the camera through geometry and the epipolar constraint, and those that fit non-linear approximation models to datasets.

2.3 The 4D Light Field

Light field imaging has emerged as a powerful tool in computer vision for robotics, offering a rich higher-dimensional representation than what can be captured by conventional optics. The underlying principle used to describe the light field is the plenoptic function, a 7-dimensional mapping that assigns radiance values to the light rays at every position in space, in every orientation, at all wavelengths, throughout all of time [1]. This can be formally expressed as $L(x, y, z, \theta, \phi, \lambda, t)$, and measured in $W/m^2/sr/nm/s$.

Levoy et al. [36] shows that with the addition of practical constraints however, the plenoptic function can be expressed more concisely as a parameterisation of 4 variables. Pixels on camera sensors integrate the number of photons arriving at them over a finite period of time removing the temporal dimension, and each colour channel can be thought of as a monochromatic sampling of the light field, removing the spectral dimension. Additionally, and importantly, is the constraint that the radiance of light rays propagating through a vacuum do not change if samples are restricted to the convex hull of the scene, thus reducing the overall dimensionality of the plenoptic function by

one additional parameter [36]. To illustrate this, one could think of the light rays leaving the inside of a bowl sat upright on a table. Many of those rays may only travel a small distance before being blocked by the inside of the bowl itself, meaning those rays will never be registered by any practical measurement device. As shown in Figure 2.3 if we consider only the convex hull of the bowl however, we are required only to represent the value of the plenoptic function on the encapsulating surface of the object [24].

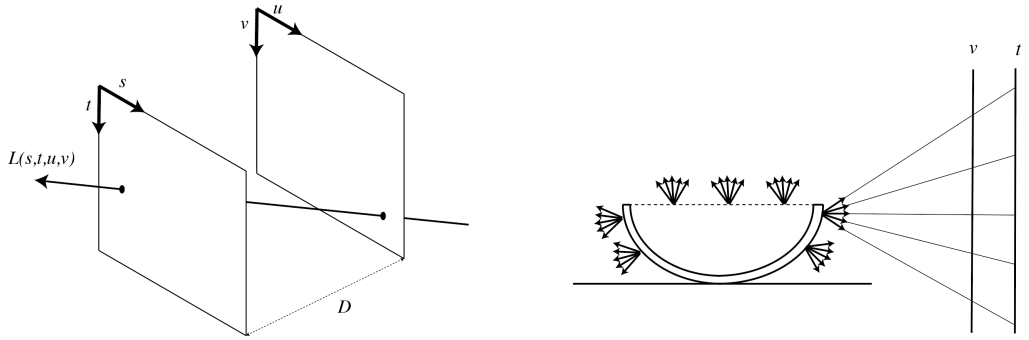


Figure 2.3: Two-plane parameterisation (left): the plenoptic function can be described by the radiance along a ray passing through two parallel planes. The free space assumption (right): if we consider only the bundle of rays leaving from the convex hull of the object at a particular instance in time, in a single colour channel, we can parameterise the light rays as a function of 4 variables rather than 7.

Also illustrated is a common convention for describing light rays in this 4 dimensional space called the two plane parameterisation [24]. In this parameterisation, two parallel planes are used to fix both the position and orientation of each ray by fixing their points of intersection with two parallel planes. By convention, the plane closest to the scene is termed u, v and the plane closest to the camera sensor is the s, t plane.

This 4D realisation of the light field originated as a model of rendering 3D computer graphics, one which shifted emphasis from notions of texture and geometric primitives to modeling the behaviour of light rays permeating space. Since then however, the conceptual framework of the light field has drawn a following of researchers at an intersecting region of signal processing, computer vision and robotics [12].

2.3.1 Visualising the Light Field

Light field imaging finds a foothold in this work through the utilisation of camera arrays, which are devices that sample multiple views of the same scene. Using a camera array is a simple method for acquiring a sparse sample of the light field, where the position of the camera determines *(s, t)* while the location of the pixel determines *(u, v)* [57]. The images captured from a camera array are mapped easily to the 4D light field. An example of a camera array - the device used for collecting the data which enabled much of this work - is shown in Figure 2.4.

Epipolar Plane Images

One way that this geometric information can be easily visualised is by taking slices of the light field image in the *s, u* or *t, v* axes as shown in Figure 2.5. Images captured from camera arrays can be stacked to form a solid volume (analogous to how colour-images form volumes from their RGB channels), from which 2D slices can be sampled. Each of these slices yields an image characterised by sheared straight lines, encoding information about the geometry of the scene, including depth and occlusions [7].

Focal Stacks

The light field can alternatively be visualised by processing the image into a 'focal stack'. Focal stacks closely resemble images with shallow depth of field such as those that can be captured from a commercial DSLR camera. Light field focal stacks differ from focus in the optical sense however in that they are synthetic and can be computed after the image has been taken, allowing control over depth of field and the focal plane in post-processing. Focal stacks can be computed from camera array images by layering images over one another and taking the average value for each pixel. The result is that parts of the scene that closely overlap appear in focus while areas with poor overlap create a 'bokeh' effect.

These representations of the light field play an important role in this thesis project as we experiment with different methods for feeding light field images to the machine learning pipeline. An important consideration in any machine learning algorithm is the feature space - based on what particular inputs will the algorithm be making its decision? Raw images contain millions of measurements and thus represent an incredibly high-dimensional feature space for neural networks to process. Light field images are several times larger, and thus it is important that some form of dimensionality reduction is used to ease the training process. With the goal of investigating effective

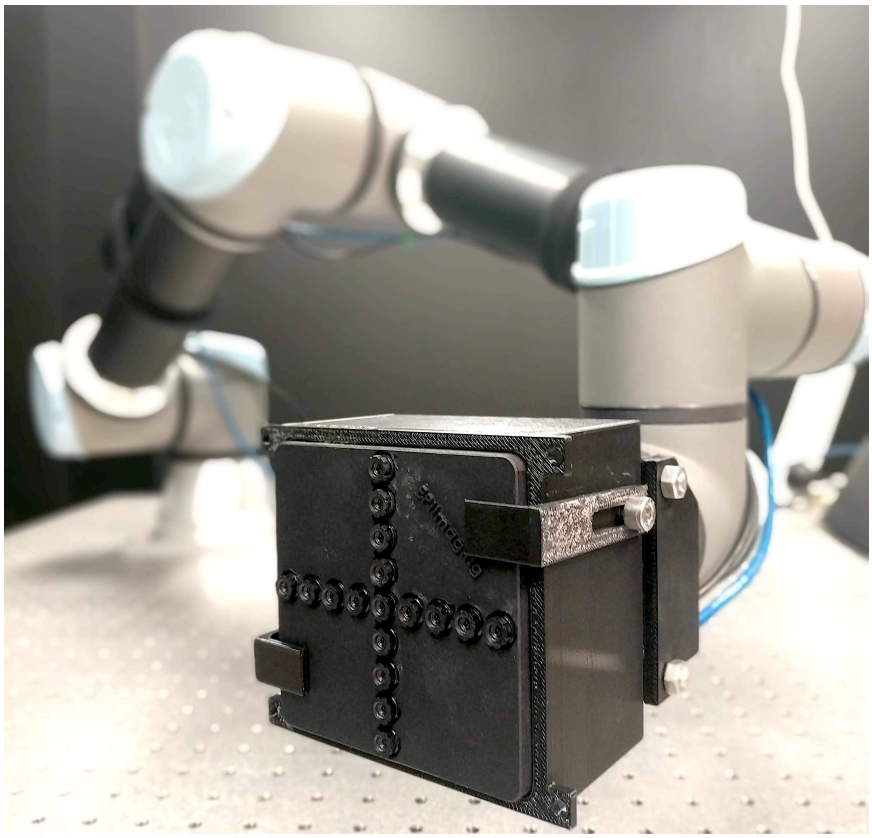


Figure 2.4: An example of a camera array mounted on a robotic arm. This camera array is configured as 17 sub-apertures arranged on a single plane in a cross-hair formation. Camera arrays sample several views of the same scene and are thus capable of acquiring a sparse sample of the light field. This is the camera that will be used throughout this thesis project.

Figure 2.5: Epipolar Plane Images (left): Shown as a slice of a volume, the images formed by dissecting the image in the s,u and t,v planes are characterised by sheared straight lines, with the grade of the slope encoding the amount of parallax experienced by a pixel at that u or v coordinate. Synthetic aperture focusing (right): taking the average of every image from the camera array yields an image where different parts are in focus depending on the alignments of the images.

methods of feeding light fields to neural networks, this thesis will explore the use of three different light field formats as the entry point to the machine learning pipeline. In each of our three proposed input formats, we rearrange the 4D data provided by our camera module as 2D image data - a challenge that forms a substantial part of this works methodology.

2.4 Machine Learning in Computer Vision

An oft-quoted anecdote in computer vision tells of MIT researcher Seymour Papert, who in 1966 assigned a summer project that sounded relatively simple - to construct a 'visual system' that could describe what objects it saw by name [46]. While the regimes of computer vision have evolved substantially since 1966, many of the ideas, and challenges have persisted. This is embodied in the popularity of projects such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [50], drawing researchers from institutions around the world. It was at the ILSVRC annual challenge where in 2012, a convolutional neural network achieved a top-5 error rate of 15.3%, outperforming all previous submissions by 10.8% [32]. Until 'AlexNet' in 2012, the competition consisted of competitors introducing algorithms for marginal improvements year on year. Needless to say, an improvement of over 10% generated noise in the computer vision community, drawing attention to the powerful capabilities of deep learning.

While deep neural networks for computer vision have gained massive popularity since the success of AlexNet, the history of neural architecture models begins with the 'perceptron', as described by Frank Rosenblatt in 1959 [48]. The fundamental building block of neural networks, the perceptron, is a module that accepts several inputs, and produces a single output computed as a weighted sum of each input - allowing complex functions to be approximated when several perceptrons are layered together as a 'multilayer perceptron' [43]. The process of finding the optimal set of weights that produce the desired output given a set of inputs is referred to as training, and in practice is usually found by optimising some cost function using the backpropagation algorithm [49]. Backpropagation uses the chain rule of calculus to calculate partial derivatives of each weight in the network with respect to the cost function. A gradient based optimisation algorithm such as stochastic gradient descent (SGD) can then be employed to minimise the cost function. Thus, it is important that each step in the computation of a neural networks output be differentiable - that is, it must support the backpropagation of gradients or else gradient-based optimisation will fail. The 'fully-differentiable' requirement for neural networks is an important consideration in this work; a novel cost function is

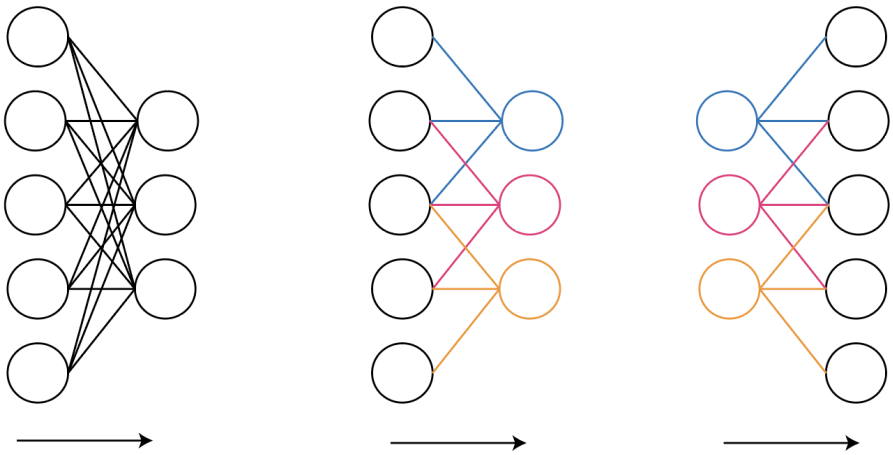


Figure 2.6: Multilayer Perceptron (left): each output from one layer is fed into the inputs of every node in the subsequent layer. Convolutional Neural Networks (middle) on the other hand have a 'receptive field', taking advantage of the spatial coherence of pixels in image data. The convolutional upsampling operation (right) is frequently used to upsample a low-dimensional feature space, to a higher dimensional one. It is often employed as a 'learned' information decompression.

described in Chapter 4, where each step in its computation must be differentiable with respect to the output of the previous step.

Convolutional neural networks (CNNs) are similar to multilayer perceptrons, but introduce a spatial invariance that makes them particularly well suited for extracting high-level features from image data. Where multilayer perceptrons are densely connected, the connectivity pattern of each node in a CNN takes advantage of the hierarchical organisation of patterns in image data by using a 'receptive field'. Biologically inspired, these nodes respond to stimuli only in their receptive field, and so they typically learn to identify salient features in the image - combinations of pixels that represent some kind of underlying structure. As these types of networks grow deeper, the features that they learn typically become more complex [34]. Early layers are provided with a small region of the image, typically a window between 3 and 11 pixels wide, and so will usually learn primitive features such as lines and corners. Deeper layers however may learn to combine low-level geometries, recognising higher-level features such as eyes and mouths, and eventually even human and animal faces.

CNNs are therefore popular 'feature extractors' in computer vision - their ability to learn to respond to different types of stimuli mean that they have been used as a dimensionality downsampling

tool, taking the millions of dimensions present in digital images and compressing them to a feature space with a much smaller number of parameters. Closely related is the convolutional upsampling operation which performs the inverse - taking a feature space and learning to upsample that feature space into something meaningful [39]. This has given rise to a popular CNN topology called the encoder-decoder architecture, which we take advantage of to predict depth (described in detail in Chapter 4). The encoder part of a CNN is composed of a series of convolutional downsampling operations - this can be thought of as finding an efficient compression of the information stored in the image. The decoder subsequently uses this compressed form of the information to extract some meaningful information about it, illustrated in Figure 2.6. One example where this architecture has been used is in [39], which outputs a classification for each pixel in the image. In this thesis, a fully convolutional encoder-decoder architecture will be used for a similar purpose, but rather than discretely classifying each pixel, we regress their depth values.

Chapter 3

Literature Survey

3.1 Geometric Approaches to Visual Odometry

Since the invention of the pinhole camera, most cameras have captured 2D representations of a 3D world, meaning information about physical structure is lost. With the addition of a second viewpoint however we can learn a little more about the shape of the scene as governed by epipolar geometry [27]. Meanwhile, a video is a sequence of images, taken in very quick succession, and so one could think of video footage as a multi-view camera where each image is separated not only spatially but also temporally. In this work, identifying the amount of motion between two temporal frames of the video camera is referred to as 'pose-estimation'. Doing this over several frames produces a trajectory, a process which we call 'visual odometry'.

Due to the diversity of applications, sensors being used, and requirements, the literature on visual odometry has produced a huge variety of approaches, some which utilise features such as SURF [3], SIFT [41], or Harris corners [26], while others take featureless approaches. Some approaches use monocular imagery by studying the epipolar relationship between two views of the same scene [40], while others use Random Sample Consensus (RANSAC) to correspond a known 3D model of a scene to its 2D projection [18], while yet other approaches solve directly for pose using closed-form solutions given two sets of matched 3D points [13]. Naturally, there is a need to distinguish between these different approaches; in this literature survey, we classify approaches to pose-estimation into one of three main categories, described in detail the following section.

3.1.1 A Survey of Geometric Approaches to Pose-Estimation

2D-to-2D: Performing monocular visual odometry is an example of estimating pose using 2D-to-2D projections. In these cases, no 3D information is available and so geometry must be inferred using the epipolar constraint, which was described in Chapter 2. The 2D-to-2D approach is problematic because with a single camera there is often no way of concretely discerning the magnitude of the movement based on pixel data alone, meaning some kind of scale factor needs to be estimated based on characteristics of the image [19, 45, 60, 61]. In fact, this scale ambiguity is often exploited by film makers - what appears as a sweeping shot of a vast landscape or monument on the big screen is often modeled as a miniature film set in the studio. Because the image is monocular, there is no way to ground our measurements of scale in real world units, and so we resort to our imaginations and learned experiences to fill in the gaps. What *is* preserved in these monocular setups however is the overall structure of the scene and motion of the camera - we may not know how large the object is or how far the camera has moved, but we *can* compute the shape of the object as well as the direction of camera motion.

3D-to-2D: Known as the Perspective-n-Point (PnP) problem, one 3D-to-2D approach requires solving for pose using a known set of 3D points and their corresponding 2D projections in the image. Fischler & Bolles [7] in their work pioneering the PnP problem [18] applied the RANSAC algorithm to solve for camera pose using $n = 3$ (abbreviated to P3P), producing 4 algebraic solutions for camera pose. They additionally showed that the P6P problem produces a unique algebraic solution.

3D-to-3D: The 3D-to-3D approach is possible when the sensor provides range data, such as in the case of stereo and RGB-D cameras. The task is simplified to finding the rigid alignment of two point clouds. This can be solved for using closed form solutions if point correspondences are known, such as Horn's method using unit-quaternions [29] or Arun *et al.*'s method using the singular value decomposition of a $[3 \times 3]$ matrix [2]. Alternatively, iterative methods such as Iterative Closest Point (ICP) [5, 9, 59] can be used to register the point clouds if point correspondences are not known.

While prior work within each of these approaches has demonstrated impressive results, we observe two limitations of these existing methods. The first limitation applies to the first two categories, which both employ 2D imagery. With increasingly capable compute and potential data throughput, we believe that the barriers to multiple-view imaging are rapidly diminishing, representing the emergence of a promising imaging modality that can provide a more robust mechanism to tackling the pose-estimation problem. Specifically, we believe that multiple-view imaging can and should be used when dealing with problems relating to geometry in computer vision. The second limitation

applies to the 3D-to-3D visual pose-estimation approach, where, to the best of our knowledge, all methods rely on a calibrated camera array, stereo pair or RGB-D camera. While these types of sensors are typically excellent at measuring 3D geometry at short ranges, even small changes to their calibration parameters are a punishing blow to their accuracy. The ability to withstand the accumulation of calibration errors forms an important goal in this work, which is why we turn to learning based methods. The remainder of this section will summarise some of the key learnings in pose-estimation presented in the works mentioned above.

3.1.2 Iterative Methods

Iterative methods are typically formulated as non-linear least-squares minimisation problems, which can be solved by linearising the problem and iteratively stepping to a minimum. Local minima represent a challenge in these minimisation problems as the global minimum may be overlooked. In overcoming this, we typically make the reasonable assumption that a standard video camera operating with real-time constraints on a mobile robotic platform generally produces frames several times per second, meaning inter-frame motion is relatively small. Iterative methods may therefore initialise their parameters to initially estimate zero movement, meaning there is a high likelihood that the iterative solution converges to the global minimum. Alternatively, a closed-form solution may be used to initialise the estimate, followed by iterative refinement.

Another iterative approach pioneered by Fischler & Bolles [18] uses random sampling to iteratively fit a model to observed data-points through a guess-and-check procedure which is robust to outliers and noisy data.

Reprojection error is an important concept in both iterative and closed-form methods of pose-estimation. It describes the photometric error when a 3D point (obtained either through estimation or range sensing) is projected onto the sensor plane of a camera. The distance between the actual observed point and the projected point is the photometric error. As a motivating example, consider an RGB-D camera which has provided a range measurement for each pixel, allowing the reconstruction of a point-cloud representation of the scene. After the camera has moved a small amount, we are interested in finding the translation ${}^2t_1$ and rotation ${}^2R_1$ from the first frame to the second. A point P from the point-cloud should be seen by the camera in the second frame at the coordinate $K({}^2R_1P + {}^2t_1)$. Comparing this with the actual observed coordinate of that point, we compute the reprojection error. More generally, given the homogenous transformation from the first pose to the second $[R|t]$, for every 3D point observed in the first frame, P_i, and their (actual) observed 2D

projections in the second frame, p_i, the reprojection error is computed as

$$\sum_{i=0}^{n} ||p_i - K[R|t]P_i||^2.$$

Hartley & Zisserman [27] formulate a pose-estimation pipeline that uses the *8-point-algorithm* to estimate an initial pose, from which point correspondences are triangulated to form a 3D representation of the scene. This 3D representation is used to iteratively refine both the 3D reconstruction and the pose estimate by minimising the reprojection error.

While these methods have been shown to perform well, doing so monocularly is an unnecessarily complex task when multiple-view imaging is now so pervasive. In this thesis, we embrace multiple-view cameras for geometry based tasks, as a family of techniques that greatly simplifies algorithm design and reduces complexity.

3.1.3 Closed-form Methods

Treating the pose-estimation problem with a closed-form solution has many advantages over iterative methods. In particular, closed-form solutions attempt to estimate pose without any iterative refinement by solving directly for the global minimum. Furthermore, the computation-time required by closed-form solutions is independent of the scenery itself, running in constant-time - an attractive characteristic in real-time systems, where consistent run-time can greatly simplify system design.

In Section 2.1.2 we studied the relationship between multiple cameras, where we described the Fundamental Matrix as the rotation and translation between two cameras, up to scale. Closely related is the Essential Matrix, which similarly describes this scale-ambiguous relationship between two cameras, with the key difference being that while the Fundamental Matrix is defined in the pixel-space of the camera, the Essential Matrix is defined instead in terms of the normalized coordinate frame. If the intrinsic parameters for the camera is known, it is a straightforward conversion between the two representations. The essential matrix can be decomposed to find the rotation and translation of the two cameras up to scale, yielding four possible solutions. The interested reader is encouraged to consult with [27] for the full derivation.

Plenoptic Flow is one technique suggested by Dansereau *et al.* [13] that takes a modular approach to visual odometry, exploiting the richness of the signals in the light-field when extended into the time-domain. The modules include a *depth estimation* component, a 4D analogue to the traditionally 2D *optical flow* problem, and a closed-form point-cloud registration algorithm [29].

The first step exploits the *gradient-depth* constraint, fixing each pixel's depth in the scene and allowing the reconstruction of a point-cloud from the first frame of video. Now, a naive approach is to subsequently perform the same operation using a second frame of video, resulting in two point-clouds that can be registered to compute the relative pose. Unfortunately, this approach produces two independent point-clouds lacking known point-to-point correspondences, meaning an iterative algorithm is required to find the rigid alignment between them. Given the approach's emphasis on closed-form solutions, an alternative idea is proposed that exploits the time-domain behaviour of light-fields. The *plenoptic flow* algorithm estimates a 3D velocity for each point from light field derivatives, producing known point-to-point correspondences, which allows registration using Horn's method for absolute orientation [29].

This approach is shown experimentally to perform well [15], however the algorithm requires prior knowledge of the camera's parameters - in other words the camera requires calibration and rectified light field imagery. In this thesis, we seek to develop a pipeline that takes advantage of plenoptic imagery, but without necessitating prior knowledge of the camera's parameters.

3.2 Data-driven Approaches to Visual Odometry

A relatively new approach that has driven a large body of research is the use machine learning and statistical modeling to perform pose-estimation, utilising the image-processing power of convolutional neural networks to learn a non-linear mapping directly from a pair of images to their relative pose [16, 20, 22, 38, 62]. Combining the spatial awareness of the convolutional down sampling operation with a neural network's ability to learn accurate approximations for complex, non linear functions, these approaches have found success in both supervised [16, 38], and unsupervised configurations [20, 22, 62].

We first establish a taxonomy of deep-learning methods that have emerged for approaching the pose-estimation problem in Section 3.2.1. In Section 3.2.2 a more extensive discussion of existing methods is provided, with emphasis on the methods the most closely align with our proposed approach, and the limitations of current methods which have guided this work's hypotheses.

3.2.1 A Taxonomy of Data-Driven Methods for Visual Odometry

All data-driven methods covered in this section relate specifically to *deep-learning* approaches for pose-estimation. In image-processing algorithms of late, deep-learning almost invariably includes

the use of a convolutional neural network, due to the powerful regularising characteristics of the convolutional down-sampling operation and their hierarchical feature-extraction capabilities.

Similarly to the geometric approaches described in Section 3.1, the very closely related task of depth-estimation and 3D scene reconstruction is an important step in many of the methods described here. Thus, this section will provide an overview of methods including both depth-estimation and pose-estimation.

Datasets: Data-driven algorithms require data, and their surge has been helpfully facilitated by the availability of massive online datasets such as KITTI [21], NYU-v2 [44], and CityScapes [10]. Such datasets are typically accompanied with suites of validation data collected from a variety of sensor sources, such as RGB-D cameras, stereo camera pairs, lidar, inertial sensors and GPS. The availability of such datasets has been crucial to the training and validation of many of the works described in this section.

Supervised Methods for Depth: In the supervised family of algorithms, Eigen et al. [16] and Liu et al. [38] take advantage of the ground truth depth maps available in the KITTI dataset to learn depth monocularly; this is an enigmatic challenge for non-learning-based methods as there is insufficient information in monocular imagery to infer depth.

Supervised Methods for Pose-Estimation: On the problem of pose-estimation and visual odometry, both Wang *et al.* [54] and Jiao *et al.* [31] combine the image-processing capabilities of CNNs with the time-domain strengths of Recurrent Neural Networks (RNNs) to estimate 6-degree-of-freedom poses over short sequences of video. In these approaches, a Long-Short-Term-Memory network (LSTM) is used to output temporally cohesive pose estimates, recognising that the inertial stability of a moving vehicle means its velocity and direction is unlikely to quickly change direction over the course of a few frames of video. These approaches have demonstrated strong results in visual odometry, however they require ground-truth pose data to supervise their networks.

Unsupervised Methods: Unsupervised methods have arisen from the recongition of two main facts. Firstly, that ground truth data is typically expensive, and time-consuming to collect. Secondly, that a visual-perception module that is capable of improving performance in-situ without direct supervision, is an attractive capability in robotics. Unsupervised approaches [20, 62] have taken advantage of the epipolar constraints imposed on moving cameras, cleverly piecing together information to simultaneously estimate both a 3D scene reconstruction, and the motion of the camera. Many of these unsupervised methods suffer from a scale ambiguity due to insufficient information, which we have discussed in Section 3.1.1. Furthermore, some methods may even suffer

from scale *inconsistency*, a more concerning issue where there are no constraints enforcing pose is estimated consistently at the same scale. For example, the pose estimation module may estimate a motion of only a few centimeters in one pair of frames, while the next pair produces an estimate of a few meters, despite the actual inter-frame motion being very similar. A more extensive overview of these types of methods is provided in Section 3.2.2, with some discussion around how the scale-ambiguity has been resolved. In this work, we suggest a method that uses plenoptic imagery, not only enforcing scale-consistency, but generating metric depth and pose estimates.

Semi- and Weakly-Supervised Methods: A more recent paradigm has been to treat the problem as one that can be approached using *some*, but not *all* of the ground-truth information that is available. In general, the same approach is used as unsupervised methods, simultaneously estimating depth and pose, with the addition of certain supervision signals. The thinking behind such approaches is that while sometimes equipment such as lidar might be too expensive or infeasible to attach to a mobile robotic platform, certain parameters about the equipment is known ahead-of-time. For example, Godard *et al.* [22] and Zhan *et al.* [58] both propose using the known distance between a stereo-pair of cameras to enforce scale-consistency and scale-awareness in their predictions. In this thesis, we apply a similar character of thinking to enforce scale awareness, by generalising to arrays of cameras. Light field imagery however, needs to be suitably formatted before insertion into 2D convolutional networks - forming an important part of our methodology in Chapter 4.

3.2.2 Unsupervised Depth and Relative Pose Estimation

This section closely follows the derivation of simultaneous depth and pose estimation proposed by Zhou *et al.* [62], although many works have adopted the same pipeline. The general pipeline utilises two CNNs as seen in Figure 3.1, one whose purpose is to predict dense per-pixel depth maps given a single image, and the other whose job is to estimate the 6 degree-of-freedom pose from the first frame to the second given both images.

While the training pipeline is unsupervised in the sense that labelled data is not required, some form of supervision signal is nevertheless required to optimise the parameters of the pair of networks. These papers take advantage of the fact that if the physical structure of a scene is known, then a novel rendering of that scene from a different viewpoint is achievable. Thus, [20] suggested using photometric reconstruction as a supervision signal - from I_{t2}, reconstruct I_{t1}. The difference between the reconstruction and the real image can form a supervisory signal for the pair of networks. This work similarly uses photometric reconstruction as the principal supervisory signal in training a pair

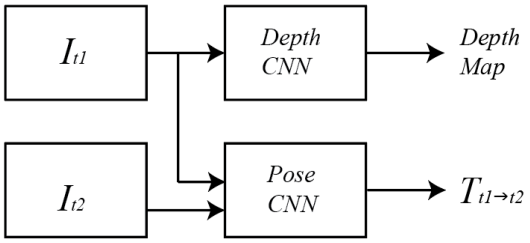


Figure 3.1: Machine learning approaches have demonstrated strong results in simultaneously estimating depth maps and relative poses between images. Typically, a pair of CNNs is used, one for each task. The depth CNN is provided with the first view at $t = 1$ and the pose CNN is provided both views.

of networks to perform visual odometry and depth estimation. The photometric reconstruction loss is

$$L_{photometric} = \sum_{n}^{W} \sum_{m}^{H} |I_{t1}(m,n) - \hat{I_{t1}}(m,n)|. \tag{3.1}$$

Humans can easily imagine what a scene will look like if we move our heads slightly, but only if we know the overall shape of the object we are looking at. Similarly, given two images separated by a small interval of time I_{t1} and I_{t2}, one could synthesise what I_{t1} would look like by sampling the pixels from I_{t2}, if the relative pose $T_{t1 \to t2}$ between the two cameras and a pixel-wise depth of the scene $D_{t1}(p)$ is known. In practice, this can be done by obtaining the coordinates of the pixels in the first image p_{t1} projected on to I_{t2}'s camera sensor. Assuming a pinhole model, the complete expression for p_{t1}'s projected location on the second camera, $\hat{p_{t2}}$ is

$$\hat{p_{t2}} = K T_{t1 \to t2} D(p_{t1}) K^{-1} p_{t1}. \tag{3.2}$$

Here K represents the intrinsics matrix. In right-to-left order, this transform first maps each pixel to a ray direction using the inverse camera intrinsics K^{-1}. Each ray is then assigned a z-coordinate using the per-pixel depth map $D(p_{t1})$, producing a 3D point cloud. The transform $T_{t1 \to t2}$ transforms each point to the coordinate frame of the second camera, then the matrix K projects each of those points onto the camera sensor of the second camera. The obtained coordinate $\hat{p_{t2}}$ is continuous, while pixel coordinates are discrete. A pixel value thus needs to be interpolated, keeping in mind that each step of a neural network pipeline must be differentiable to support the backpropagation of gradients.

It was suggested by Zhou et al. [62] that adopting bilinear sampling could be used as a fully differentiable sampling mechanism. The use of bilinear interpolation as a differentiable sampling pipeline was first proposed Jaderberg et al. [30], and adapted by Zhou et al. [62] to perform the differentiable image warp. A bilinear sampling kernel is described by

$$V = \sum_{n}^{W} \sum_{m}^{H} U_{nm} max(0, 1 - |x - n|) max(0, 1 - |y - m|). \tag{3.3}$$

V is the output of the sampling kernel, U is the source image being sampled, n and m index over the columns and rows of the kernel respectively, H and W are the height and width of the sampling kernel, and x and y are the local coordinates of the sampling location. The 2×2 sampling kernel used to perform the photometric reconstruction is, in essence a weighted sum of the 4 nearest neighbour pixels, based on its proximity to those pixels. The equation is differentiable with respect to both x and y:

$$\frac{\partial V}{\partial x} = \sum_{n}^{W} \sum_{m}^{H} U_{nm} max(0, 1 - |y - m|). \tag{3.4}$$

$$\frac{\partial V}{\partial y} = \sum_{n}^{W} \sum_{m}^{H} U_{nm} max(0, 1 - |x - n|). \tag{3.5}$$

Using this bilinear sampling kernel, we can thus sample pixels from I_{t2} to reconstruct I_{t1} in a fully differentiable manner, allowing the backpropagation of gradients through the networks. The pipeline with photometric reconstruction as the supervision signal can be illustrated as shown in Figure 3.2. Bilinear interpolation is similarly employed in this work to compute the photometric reconstruction loss of the estimated depth and pose. However, because this work operates on light fields, the output of photometric reconstruction won't be a single image, but a whole array of images.

In addition to the photometric reconstruction loss providing the main supervision signal to the network, Zhou et al. [62] employs a smoothness loss to ensure that the produced depth map is globally smooth. The smoothness penalises the second order gradient of the image - i.e. the depth network is encouraged to produce a depth map characterised mostly by low-frequency components, and penalised for high-frequency components. Recognising that depth discontinuities frequently occur in parts of the image where a strong edge appears, Godard et al. [22] on the other hand suggested the use of an edge-aware smoothness loss that also penalises large gradients in the depth map ∂d, but lowers the weight of the loss in regions where the image gradient ∂I is large.

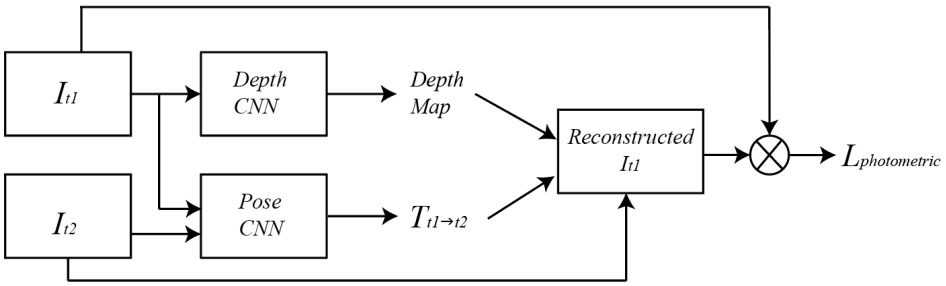


Figure 3.2: To provide a supervisory signal to the depth and pose CNNs, Garg et al. [20] suggested using photometric reconstruction. The loss function is formulated by taking the difference between the reconstructed image $\hat{I_{t1}}$ and the actual image I_{t1}, shown in equation 3.1.

$$L_{smooth} = \sum_{n}^{W} \sum_{m}^{H} |\partial_x d_{nm} e^{-|\delta_x I_{nm}|}| + \partial_y d_{nm} e^{-|\delta_y I_{nm}|}| \tag{3.6}$$

There have been several iterations [6, 22, 23, 58] of this pipeline, most of which have focussed on introducing novel loss functions to improve the quality of the depth and pose estimates. To the best of our knowledge however, this pipeline has only been applied to monocular and stereo camera imagery. While monocular and stereo cameras are ubiquitous in modern digital devices, this thesis drives the investigation further, studying the potential for the use of similar pipelines to achieve improved results using a broader, more generalised family of imaging devices.

Breaking free of the pinhole principle from which human eyes have developed and which commercial cameras have adopted, has wide implications in the field of machine vision. Reinforcing this idea is the rise in popularity of imaging techniques such as multi-spectral imaging, plenoptic cameras and polydioptric cameras, however interpreting ray directions and geometry for new cameras may not always be apparent, giving rise to the data-driven approach. Working on the principle that machine vision does not necessitate mimicking the human eye to capture imagery, this thesis investigates the capabilities of machine learning for querying depth and pose using one such novel imaging device - a camera array.

3.3 Convolutional Neural Networks and Light Fields

Crafting a data-driven pipeline for visual odometry and depth requires a neural architecture that is capable of ingesting light field data. Importantly, the shape of the data must be considered. Our

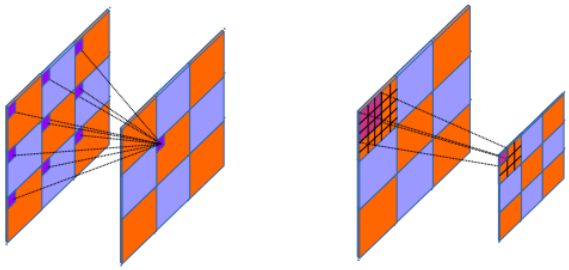


Figure 3.3: Adapted from [55], this figure shows convolutions in the angular and spatial dimensions of the light field. Angular Filters (left): by fixing the coordinate *(u, v)* and varying *(s, t)*, the 2D grid of images can be presented to a neural network in the form of orthographic images. Spatial Filters (right): instead of fixing *(u, v)* coordinates, *(s, t)* is fixed while *(u, v)* is varied. This presents information to the neural network as a grid of perspective images.

parameterisation of the light field has four dimensions, denoted by *S, U, T, V*. Convolutional neural networks normally operate on 2D image data, meaning the search for a suitable neural architecture is an important step in testing this work's hypotheses (convolutions in 3 and 4 dimensions are also possible, however they incur significant computational overhead, and for reasons that will become apparent in the next chapter, are difficult to apply to our specific image data).

Wang et al. [55] exploit the rich textural information contained in the light field, applying a convolutional network to the task of material classification using light field imagery. In addition to reporting significantly better accuracy compared to 2D imagery (with gains of over 7%), [55] suggests 5 novel methods for interpreting light field data for convolutional networks. One of Wang's most promising architectures, which is referred to in their paper as using interleaved 'spatial' and 'angular' kernels, exploits the fact that presenting light field data in a different order exposes new kinds of information.

As discussed in [12], a 4D light field image may be tiled as a 2D grid of 2D projective images (called *(u,v)* slices). Alternatively, the light field may be tiled as a 2D grid of 2D *orthographic* images, as seen in Figure 3.4. By presenting the image data to their network in the two different forms as shown in Figure 3.3, Wang et al. approximates convolution in 4 dimensions, allowing their network to learn a 4D signal processing function without the computational overhead of convolving in 4 dimensions.

There are some hurdles in this approach that make it challenging to apply in our depth and visual odometry pipeline. Most importantly, arranging the light field as orthographic images is impossible

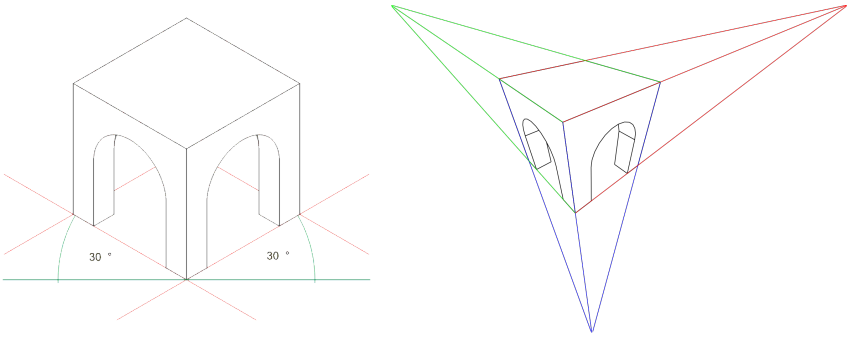


Figure 3.4: Orthographic projection (left) [4]: the lines of sight to all points in the scene are parallel to one other, and perpendicular to the projective plane. Perspective projection (right) [25]: lines of sight converge to a point giving the effect that distant objects appear smaller than nearby objects.

with our specific camera without introducing significant redundant data. The challenge arises from the shape of the camera array. As shown in Figure 2.4, our camera has sub-apertures arranged in a cross-hair formation, meaning the complete 2D grid of 2D images is unavailable to us. In the following chapter, we will describe a novel technique that, similar to Wang et al. approximates convolution in 4 dimensions by presenting light field data in a different order to the ordinary *(u, v)* slices that we are familiar with.

Wang et al. suggests another suitable architecture for application in this work, which involves stacking the images along their colour-channels, feeding the data to a convolutional network as a cuboidal volume. This approach is a simple and intuitive way of arranging the light field data, however neural networks may not be able to take full advantage of the 4D signals in light field data this way. Nevertheless, it is a useful way of representing the light field, especially as a format which is easily prepared for ingestion by CNNs, and is thus also explored as an input format in this work's experiments.

3.4 Learning Depth From Light Fields

In Section 3.2.1 we briefly touched on many of the experiments that have been conducted using the KITTI dataset, including two approaches for the supervised learning of depth from monocular imagery. We extend on that discussion in this section, describing in more detail some of the approaches for learning depth from light fields, rather than monocular imagery. This is a relevant and important part of the literature to understand, as depth estimation forms an integral part of our

visual odometry pipeline.

Almost all learning based approaches employ convolutional neural networks - they have been shown time and again to perform well on image data. Within the CNN family of learning-based depth estimation, we distinguish between work that uses 2D convolutional neural networks, and those using convolutions in 3 or more dimensions.

Within the 2D family of algorithms, Heber *et al.* [28] was the first to suggest a deep learning based method for depth estimation from light fields. Their approach used a CNN to learn an end-to-end mapping between the 4D light field and 2D hyperplane orientations - which translate directly to depth.

Epipolar plane images (EPIs) are employed frequently in 2D methods - EPIs directly encode depth as simple 2D features, making them an attractive input format for depth estimation. For example, Sun *et al.* [56] suggest an enhanced EPI feature that encodes scene depth at each point in the light field, which they use to estimate depth using a supervised CNN. Leistner *et al.* [35] takes this one step further, generalising the use of EPIs for learning depth from wide-baseline light fields. Using a strategy which they describe as *EPI-shift*, their technique reduces the size of the convolutional receptive-field needed to detect especially shallow slopes, which are characteristic of wide-baseline light field cameras. Shin *et al.* [51] extend the use of EPIs for depth, by also incorporating the diagonally constructed EPIs for a total of 4 individual EPI feature maps. Each of these is encoded by a separate feature extractor, where features are subsequently concatenated and fed to a secondary CNN which outputs disparity predictions.

Light fields are typically represented as a 2D array of 2D images, begging the question, why are higher-dimensional convolutions not frequently used? This approach has indeed been suggested, by Faluvégi *et al.* [17] who use a fully-convolutional 3D CNN to predict disparity from light field images, demonstrating results that are competitive with the state-of-the-art. On the other hand, 3D convolutions are computationally expensive, and many of the benefits of 2D CNNs are lost - such as the ability to use pre-trained networks or the ability to employ existing 2D architectures.

3.5 Summary

In this chapter, we've introduced the idea of odometry and depth perception as two challenging tasks which might be addressed with visual sensors. We followed this discussion with a survey of current state-of-the-art methods - which we have broken into two categories - the *geometric* approach

which emphasises mathematically modelling the geometry of a moving camera, and the *data-driven* approach which instead fits a non-linear approximation model to large datasets. This work addresses a challenging problem, that is not addressed in any of the methods we have surveyed - the ability to learn depth and odometry from an uncalibrated camera array. We approach this problem in this work, by suggesting a flexible algorithm that, in principle, operates on arbitrarily large arrays of cameras. Furthermore, our algorithm combines successes from both geometric and data-driven families of algorithms, occupying a niche that draws on the success of plenoptic imaging, and the self-supervising features of machine learning approaches. Previous work had not suggested any methods for estimating depth and pose using uncalibrated camera arrays - in Chapter 4, we will outline in detail the operation of our suggested algorithm to overcome this shortcoming using unsupervised learning. This algorithm will bring robotics one step closer towards autonomy in environments where adverse effects such as shock and vibration are punishing for camera calibrations. In Chapter 5 we apply both qualitative and quantitative analysis to the results of our proposed pipeline.

Chapter 4

Methodology

In Chapter 3, we saw that previous work has produced a variety of approaches to depth estimation and visual odometry, using both data-driven and hand crafted solutions. In this chapter, we propose a novel pipeline that behaves similarly to the pipelines described in Chapter 3, with the key distinction of operating on 4D, as opposed to 2D data. The contributions of previous work focus on the use of monocular and stereo image data, while this chapter focuses on developing a pipeline that employs the full breadth of geometric information in the light field. The first section of this chapter describes the tools and methodologies of acquiring a suitable dataset, while the second part derives the pipeline used for our experiments.

4.1 Data Acquisition

4.1.1 Ground Truth Pose Data

Important to this work is a strong evaluation framework. Existing datasets such as KITTI and CityScapes benefit from state-of-the-art sensor suites including inertial sensors and lidar, allowing researchers to effectively benchmark their results. Similarly, this thesis places a strong emphasis on validation using ground truth data. One of the primary tasks for evaluation is visual odometry, and so ground-truth pose data for each image is a valuable resource. In this project, pose is collected by attaching the camera to a Universal Robots UR5E robotic manipulator, which is capable of sensing to a high degree of accuracy the pose of the end effector. Furthermore, the UR5E is able to perform accurate movements and trajectories with payloads of up to 5 kilograms, making it a suitable robotic

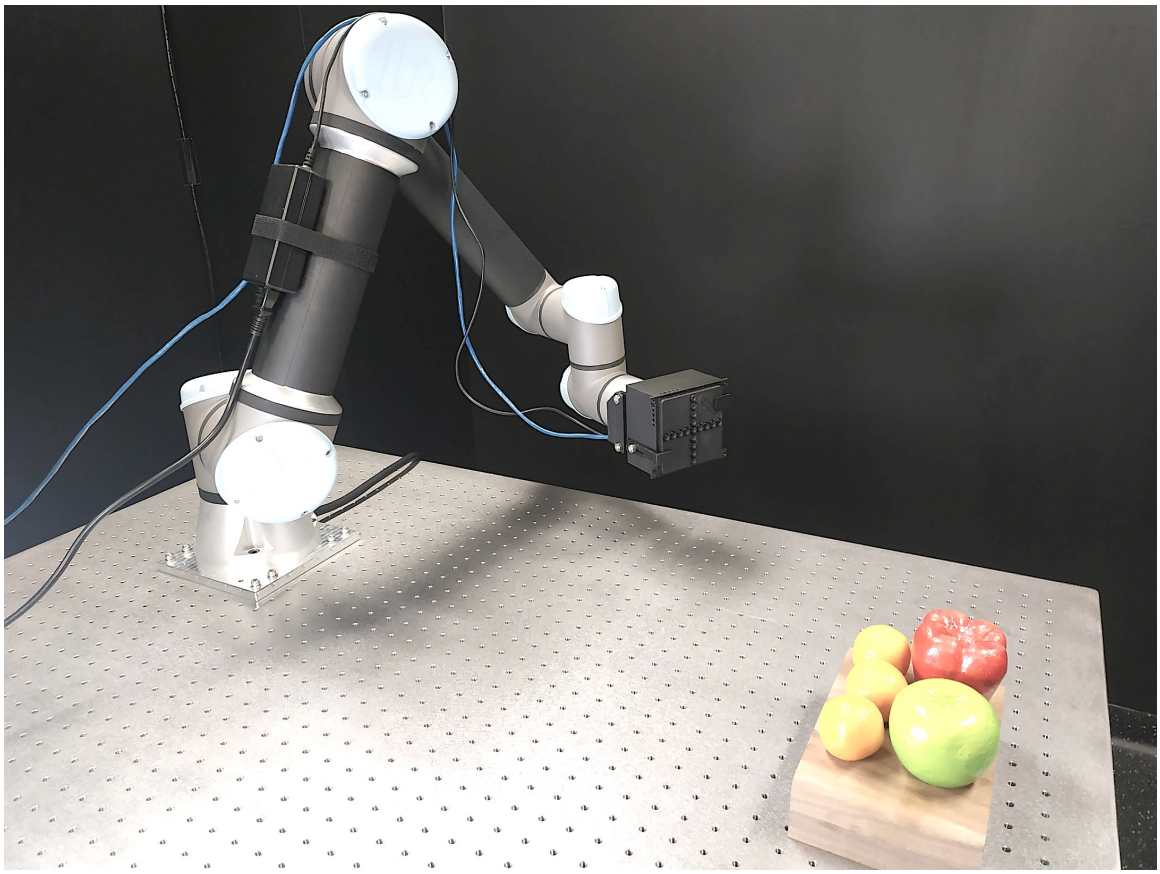


Figure 4.1: The experimental setup utilises a Universal Robots UR5E robotic arm to precisely measure ground-truth pose to allow effective evaluation of the project's visual odometry results.

mount for the camera.

Manipulator-Camera Interface Adapter

The UR5E uses a standard interface plate for attaching end-effectors such as gripping tools. The camera on the other hand, is an early prototype provided by EPIImaging, without a physical adapter interface for the UR5E manipulator. We therefore fabricated our own mount for the device, taking into consideration the camera's ventilation and cabling requirements, shown in Figure 4.1. A two-piece adapter was designed in SolidWorks and fabricated using a fused deposition modelling (FDM) platform with polylactic acid (PLA) filament.

Manipulator Control

A client library for communicating with the UR5E was developed in python, establishing a TCP/IP connection with the robot over the local network, allowing both movement commands to be sent to the robot, and feedback data about the end-effector's pose to be streamed back to the client PC. The client library also provides functionality for computing trajectories and corresponding joint angles, using the PyBullet [11] physics engine and the known kinematic model. One useful trajectory function from our library procedurally generates waypoints while keeping the camera's field-of-view trained on the scene, using a stringent collision-checking mechanism to enable autonomous data-capture.

4.1.2 Imagery

The specific imaging device being used is manufactured by EPIImaging, and consists of 17 sub-apertures. The communication interface with the camera is a network connection serving image data using the HTTP protocol via a REST API. Both rectified and un-rectified imagery can be retrieved, and the format of the returned data is a $17 \times 3 \times 1280 \times 960$ stream of raw bytes representing 8-bit pixel values from each of the 17 sensors. A simple client library was developed in python to automate API requests, and perform decoding of the pixel data.

4.2 Architectural Overview

In this section, we derive a pipeline for simultaneous depth and visual odometry using light field data. Our pipeline is constructed modularly, so to build a precise intuition for how our pipeline operates, we begin by providing an architectural overview, followed by detailed descriptions of each individual module. We introduce the CNN modules used to estimate pose and depth. Subsequently we'll discuss strategies for forming meaningful loss-functions to train our pair of networks using a photometric-warp loss-module. Next, we launch a discussion on the input formats, altering how light field data is fed to our networks, discussing the comparable advantages and disadvantages of each. Finally, we introduce some modifications to the photometric-warp loss-module that sacrifices some flexibility in exchange for more robust depth and pose estimates that more-fully exploit information in the light field.

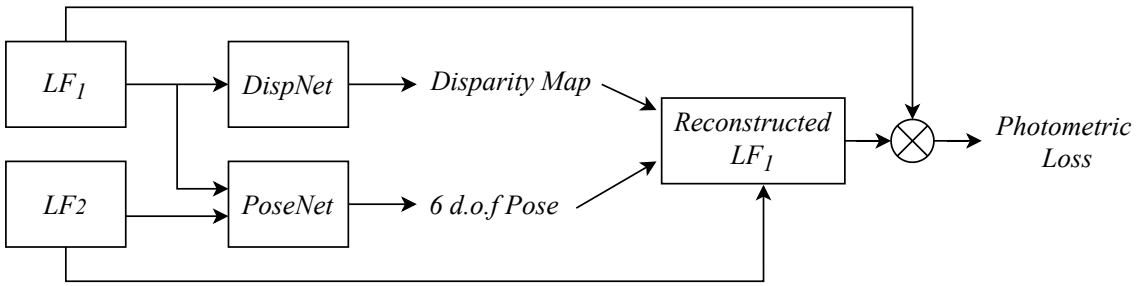


Figure 4.2: Our three main modules work together to estimate what the next frame of video looks like. Using our actual knowledge of what that frame looks like, we compute a photometric loss, which is back-propagated and used to apply gradient updates for the disparity and the pose networks.

4.2.1 Learning Pipeline Overview

Like [62], we exploit the properties of a moving camera - often dubbed 'shape-from-motion' - to learn depth and visual odometry in an unsupervised manner. Our learning pipeline is composed of three central modules - depth estimation, pose estimation, and a bilinear interpolation photometric warp module. These three modules work cohesively to estimate a novel view of the scene - i.e. the next frame of video. Using our actual knowledge of what the next frame of video looks like, we compute a fully differentiable loss function which allows the application of a gradient update to the parameters of the depth and pose estimation modules. The architecture is illustrated in Figure 4.2.

4.2.2 Pose Estimation

We treat pose estimation as a regression problem, estimating three translational components [X, Y, Z] and three rotational components [Rx, Ry, Rz]. We perform this regressions with a convolutional network, which we call 'PoseNet', in line with previous work [58, 6, 62]. This network architecture is fully convolutional, meaning that in principle, it operates on arbitrarily large or small images. The network architecture is shown in Figure 4.3.

4.2.3 Depth Estimation

Depth estimation is similarly modeled as a regression problem, with two output layers - regressing half- and full-scale depth map predictions. The depth estimation module uses the fully-convolutional encoder-decoder 'DispNet' architecture [42]. Skip connections from the encoder to the decoder mean

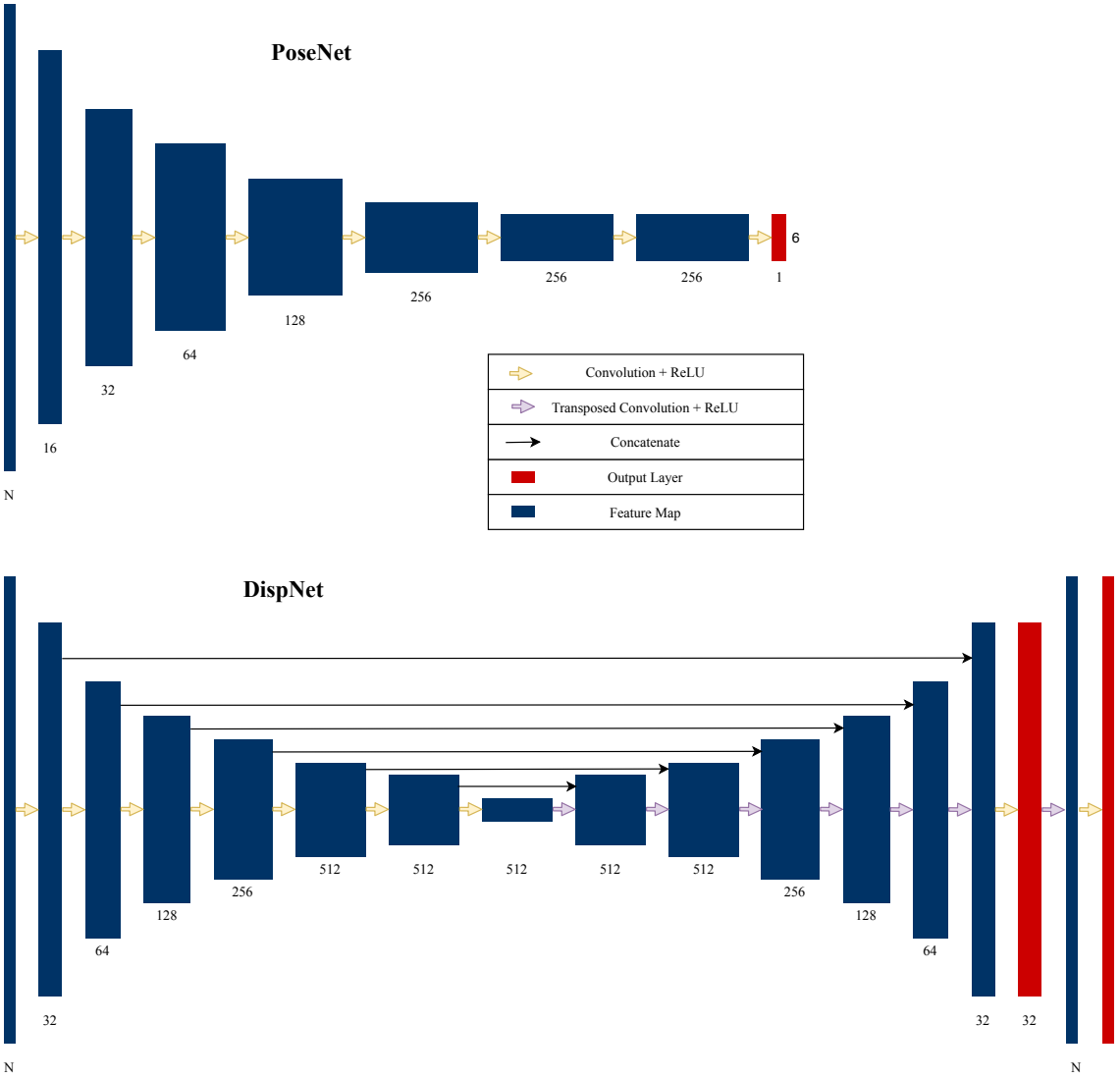


Figure 4.3: The PoseNet architecture is a fully convolutional neural network. The input is of shape $H \times W \times N$, and the output shape is $6 \times 1 \times 1$ - one output for each of the 6 degrees of freedom predicted. Similarly, the DispNet architecture is a fully convolutional network that outputs disparity maps at two different scales (shown in red). Skip connections allows the incorporation of both high and low level features in estimating disparity. In the first variant of our algorithm, there is a single output channel, predicting a disparity map aligned with the center view of the camera array. In the second variant, there are N output channels - one for each sub-aperture. These layers are displayed for illustrative purposes only, and are not to scale.

Table 4.1: PoseNet Convolutional Network Architecture

Layer	Kernel Size	Output Channels	Stride	Padding	Activation
Conv1	7	16	2	3	ReLU
Conv2	5	32	2	2	ReLU
Conv3	3	64	2	1	ReLU
Conv4	3	128	2	1	ReLU
Conv5	3	256	2	1	ReLU
Conv6	3	256	2	1	ReLU
Conv7	3	256	2	1	ReLU
Output	1	1	1	1	-

the decoding layers are able to access both high level features and low level features on either side of the latent space. Prior work [62] has demonstrated that predicting depth maps at multiple scales improves photometric reconstruction error by enforcing some level of local smoothness. In practice, this helps the network learn to predict depth in challenging, textureless regions of the image. The network architecture is shown in Figure 4.3.

4.2.4 Photometric Warp Loss

We have now described one CNN tasked with estimating a per-pixel depth map of the scene, and another which estimates the 6-degree-of-freedom pose between two frames of video. The challenge is now to provide a supervision signal that sufficiently constrains these networks to actually perform the tasks of depth and pose estimation.

A Differentiable Loss Function with Image Based Rendering

As described in Chapter 3, given an estimate for both depth and pose, we can synthesise approximate images of the scene as seen from nearby viewpoints. We take advantage of this fact to form a photometric-warp loss, similar to [62].

As described in Chapter 3, the procedure for this is to first project each pixel from the first light field LF_1 to a 3D point cloud using the known camera intrinsics and estimated depth values. Recall that the matrix K is the intrinsics matrix introduced in Chapter 2, which maps pixel coordinates to ray directions for a pinhole camera. Using the intrinsics and the regressed depth map $D(s_0, t_0, u, v)$,

Table 4.2: DispNet Convolutional Network Architecture.

	Layer	Kernel Size	Concatenate	Output Channels	Stride	Padding	Activation
Encoder	Conv1	7	-	32	2	3	ReLU
	Conv2	5	-	64	2	2	ReLU
	Conv3	3	-	128	2	1	ReLU
	Conv4	3	-	256	2	1	ReLU
	Conv5	3	-	512	2	1	ReLU
	Conv6	3	-	512	2	1	ReLU
	Conv7	3	-	512	2	1	ReLU
Decoder	UpConv7	3	Conv6	512	2	1	ReLU
	UpConv6	3	Conv5	512	2	1	ReLU
	UpConv5	3	Conv4	256	2	1	ReLU
	UpConv4	3	Conv3	128	2	1	ReLU
	UpConv3	3	Conv2	64	2	1	ReLU
	UpConv2	3	Conv1	32	2	1	ReLU
	Output2	3	-	1	1	1	Sigmoid
	UpConv1	3	-	32	2	1	ReLU
	Output1	3	-	1	1	1	Sigmoid

Layers prefixed with 'Conv' or 'Output' are convolutional layers, and those prefixed with 'UpConv' use the transposed-2D-convolution to upsample the preceeding layer.

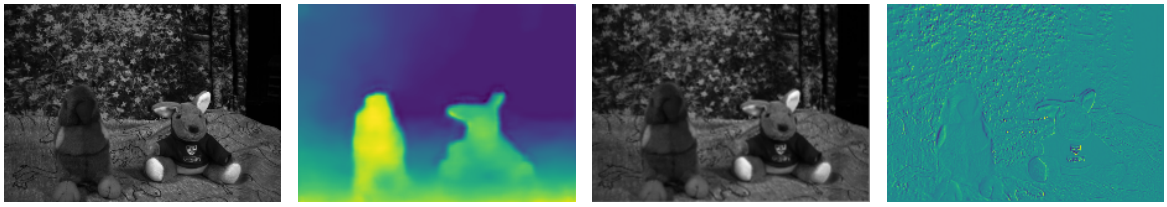
we can project each pixel $[u, v]^T$ to a 3D point cloud Q

$$Q = D(s_0, t_0, u, v)K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \tag{4.1}$$

The 3D coordinate Q is then projected onto the sensor plane of LF_2, at the pixel coordinate $[s_0, t_0, \hat{u}, \hat{v}]$. This relies on our estimate of the relative pose $[R|t]$ from the first to the second camera origin

$$[s_0, t_0, \hat{u}, \hat{v}] = K[R|t]Q. \tag{4.2}$$

The pixels $[s_0, t_0, \hat{u}, \hat{v}]$ are subsequently sampled to populate a new array, forming our synthesised image. This transform is equivalent to constructing a 3D model of the scene using our depth values, moving an imaginary camera by $[R|t]$ and taking a virtual snapshot of the scene from this new angle, the snapshot being the resulting photometric warp. The steps are shown in Figures 4.4 and 4.5.



(a) Target Image **(b)** Predicted Depth **(c)** Synthesised Image **(d)** Photometric Error

Figure 4.4: To recreate the target image, we use the depth map to calculate the pixel coordinates of the source image from which to sample. The source image is then sampled using bilinear interpolation to reconstruct the synthesised image. The reconstructed and target image are then compared to calculate the photometric error.

More broadly, the practice we have described here is referred to as 'image based rendering', which is a related, but slightly different paradigm from the rasterization, shaders and textures typically used in geometry-based rendering engines. In image based rendering, we instead synthesise viewpoints using existing image data. In a sense, image based rendering simply shuffles the pixels of an existing image around, according to some pre-defined rules. In our case, we derive the shuffling-rules using 3D data inferred by our depth and pose networks.

The perceptive reader may have noticed that the pixel value $p_2 = [s_0, t_0, \hat{u}, \hat{v}]$ is continuous in u and v, while pixel coordinates are, by necessity, discretely valued. A naive solution is to discretise the coordinate p_2 by sampling its nearest-neighbouring pixel. We must remember however that neural networks rely on the chain rule of calculus to perform back-propagation, and so whatever sampling regime we choose must be differentiable. We therefore turn to bilinear sampling, as described in Chapter 3 as a differentiable sampling mechanism. Its operation and derivatives are described in equations 3.3-3.5.

This sampling-based approach is limited in regions surrounding occluders. When the motion of the camera dis-occludes part of the scene in the target image, the sampler is tasked with recreating an image using information that it does not have. Thus, we are most likely to see 3D boundaries as

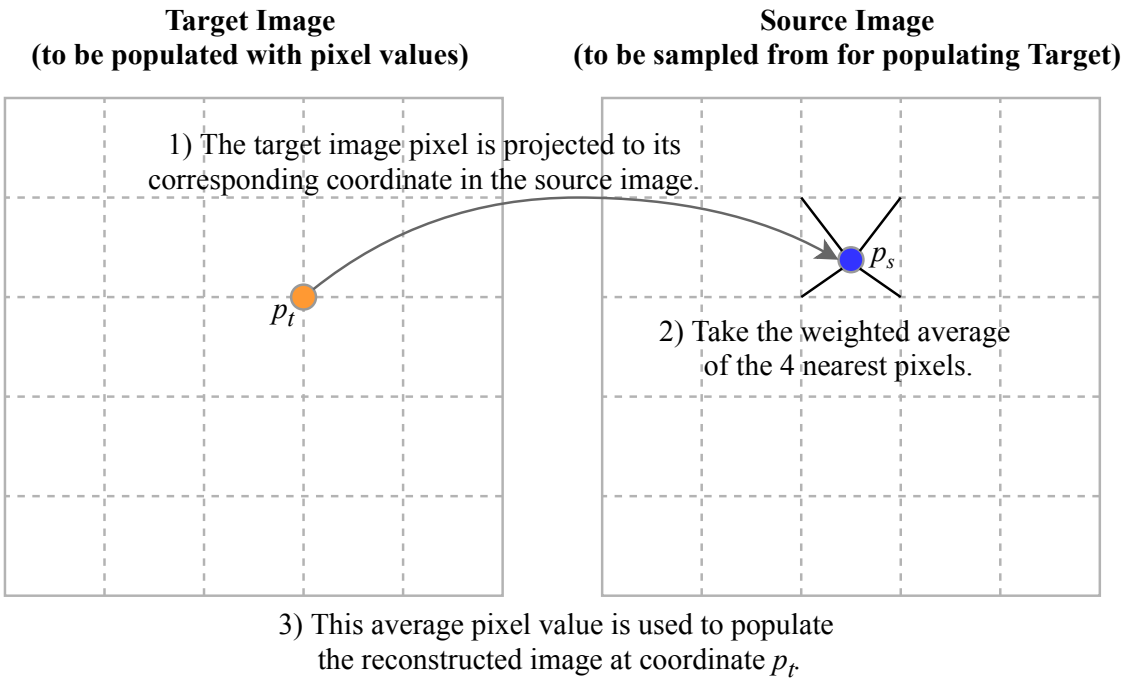


Figure 4.5: We use bilinear interpolation as the differentiable sampling process, just as suggested in [62]. To populate each pixel value p_t in the target image, each pixel is projected to its corresponding coordinate in the source image, and sampled using bilinear interpolation.

one of the main sources of error. Furthermore, this formation of a loss function assumes a texturally rich image which can be used to compute the error. These effects are apparent in Figure 4.4 (d), where we observe heavy errors around 3D boundaries, as well as texturally rich regions of the image.

4.2.5 Input Methods

This work tackles an interesting and unique challenge, of how to feed sparse light field data to convolutional neural networks. We summarise this challenge as:

Given sparse 4 dimensional data, how best can we arrange that data as 2 dimensional slices in a valuable and informative way to a convolutional network for the purposes of visual odometry and depth estimation.

As we have seen in Chapter 3, this problem has been approached in prior work, usually with the assumption that a complete 2D grid of 2D images is available. In this work however, the specific camera configuration prohibits arranging our data as a 2D grid of images without introducing

significant redundant data. In a way, this forces us to consider more flexible and generalisable methods for processing light fields into constructive forms of 2 dimensional data. In this section, we suggest three experimental methods for ingesting light field data through neural networks. In the next chapter, we will present visual odometry and depth results using each of these methods.

Volumetric Images

The most straightforward of our three methods, this technique involves taking *U, V* slices (perspective-projection images) of the light field, and stacking them along a 3rd axis (usually reserved for the colour-channels). The resulting image is an $H \times W \times N$ volume, where H and W are the height and width of the *U, V* slice, and N is the number of sub-apertures being sampled. This method is suggested, and tested by Wang et al. [55] for material classification, who report improved results over conventional 2D imagery. Internally, we expect a convolutional network to learn features relating to parallax, occlusion and depth when exposed to data of this format.

Focal Stacks

Focal stacks are constructed as a superposition of 2D images from several sub-apertures, creating interference in regions of the image that are 'out-of-focus', while keeping 'in-focus' elements of the image crisp. A focal stack specifically encodes depth in the form of interference at each region of the image. Thus, we might expect a CNN trained on this type of imagery to learn to decode in- and out-of-focus parts of the image, using this information to estimate depth and pose. Importantly, we must consider that our spatial sampling rate (distance between sub-apertures) is significantly lower than our angular sampling rate (distance between pixels). As a result our focal stacks exhibit significant aliasing effects which may impact training.

Tiled Epipolar Plane Images

In Chapter 3 we saw that Wang et al. [55] sliced the 4D light field in two different ways, first as perspective-projection images, and secondly as orthographic-projection images. By showing slices in *S, T* and *U, V*, Wang et al. suggests that a convolutional network is able to learn to approximate 4D signal processing functions without the computational overhead of convolving in 4 dimensions. Due to the specific constraints of the camera array being used (and indeed of any camera array not arranged as a regular grid of sub-apertures), there are challenges associated with slicing our light field data as orthographic images. We suggest that there is indeed another meaningful way of slicing

Figure 4.6: While synthetically changing the focal plane brings different parts of the scene into focus, the trade-off is the aliasing and artifacts that arise. In this example, we show two synthetically refocussed images, performed using 5 of the 17 available light field samples. The artifacts are especially clear around the boundaries of the fruit shown here - not only are they blurry, but distinct artifacts and ghostlike edges are clearly visible.

the 4D light field into 2D arrays - namely as epipolar plane images. As we discussed in Chapter 2, epipolar plane images are *S, U* or *T, V* slices of the light field, encoding depth and occlusion in the slope of their characteristic sheared lines. With our input strategy, we tile *T, V* slices side-by-side, and *S, U* slices top-to-bottom.

One challenge we face in ingesting this data, is that both previously described methods have input shapes $H \times W$. The tiled images from Figure 4.7 on the other hand have shape $H \times (W \times N)$ where N is the number of sampled sub-apertures. In practice, these images are much wider than either of our previous input formats. We therefore suggest the use of an additional convolutional encoding module that down-samples the tiled EPIs to the shape expected by our depth and pose networks.

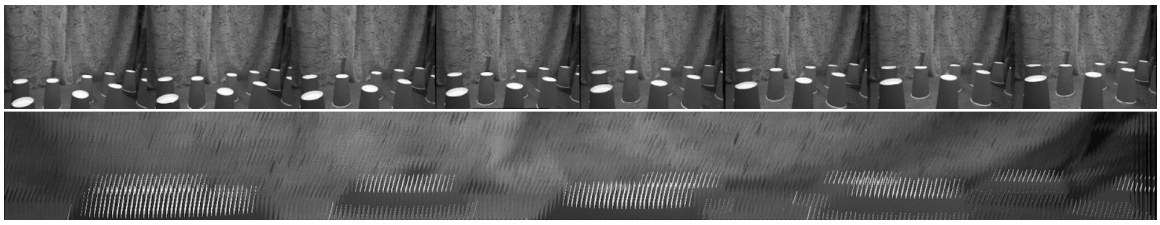


Figure 4.7: Tiled *U, V* slices (top): this is an intuitive way of slicing the light field, which tiles images from each sub-aperture side-by-side. Tiled *T, V* slices (bottom): somewhat less intuitively, this method instead tiles the vertical epipolar-plane images side-by-side. We call these *wide* EPIs. As shown, the overall dimensionality and shape of the light field is unchanged, we simply present the information in a different order, revealing different encodings of the captured scene. We can also tile the light field as *S, U* slices, stacking them vertically, which we refer to as *tall* EPIs.

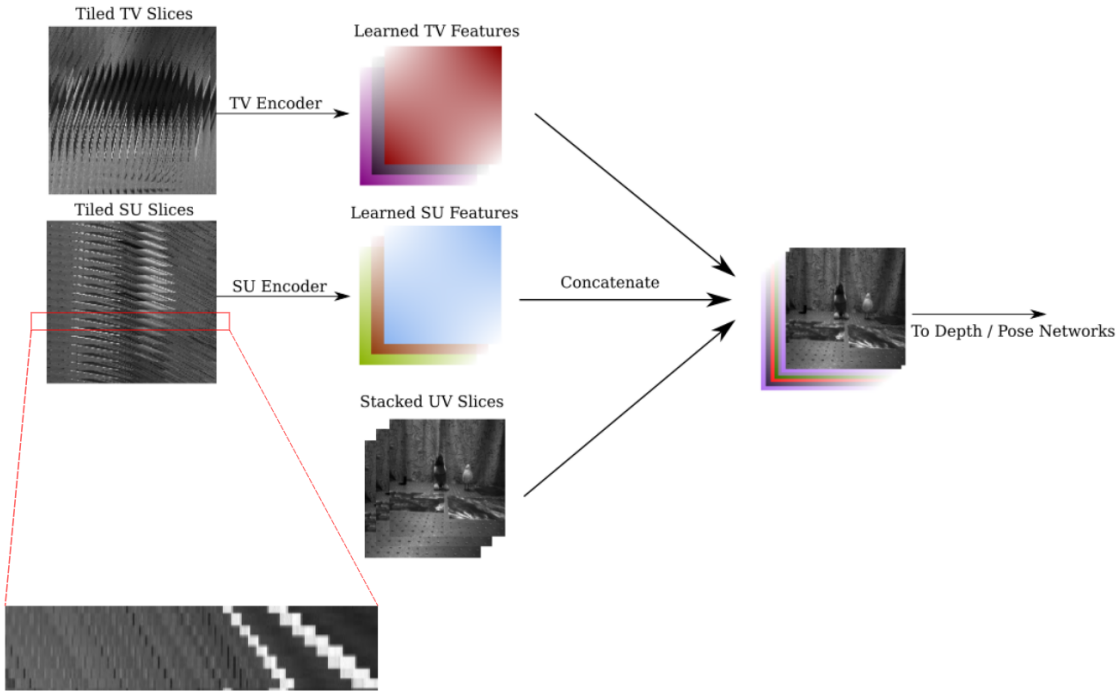


Figure 4.8: Our proposed encoder module which encodes the tiled EPIs, downsampling them to the shape expected by our depth and pose networks. Notably, features which have been learned in *T, V* as well as *S, U* are concatenated with the stacked *U, V* slices (exactly as seen in our first input method.) For brevity, we have only shown a small patch of the tall and wide EPIs, but the reader is reminded that these tilings are in fact very tall, and very wide.

The encoding module consists of one convolutional layer followed by a ReLU activation layer. For the *wide* EPIs we convolve with a kernel of size $N \times N$, and a horizontal stride of N. Zero-padding is applied to the top and bottom of the tiled EPI. The effect of this convolution is that the wide EPI is down-sampled with the same output size as a single *U, V* slice. Similarly, we apply an $N \times N$ filter with a vertical stride of N to the tall EPIs, also reducing their shape to that of a *U, V* slice. The resulting feature maps are combined with the *U, V* slices, and provided to our depth and pose networks. Thus, our networks are able to learn features in three different feature spaces. We hypothesise that with this ingestion strategy, our networks are able to learn an approximation for 4D signal processing functions, without the computational overhead of convolving in 4 dimensions, and without the requirement for a complete 2D grid of images.

4.2.6 Single-View vs. Light Field Reconstruction

In this section, we modify our pipeline to more-fully take advantage of the information in the light field. One important weakness of the pipeline we have described so far, is that there are no constraints on our networks to estimate scale with the same magnitude between video frames. Given three frames of video of the same scene, and stable camera motion, the depth network will happily estimate a room that is twice as large in the second pair of frames than the first, if of course the pose network also estimates a twice-as-large translation. This is the scale ambiguity that we have discussed in Chapter 2. With the modifications suggested in this section, we take advantage of the fact that we have multiple sub-apertures capturing the scene at the same time, using this knowledge to enforce scale consistency across video frames.

The modification described here is straightforward, but it sacrifices some flexibility. The pipeline as we have described it so far requires only one vital piece of information to learn depth and visual odometry - the focal length of the camera being used (assuming all sub-apertures have very similar focal lengths). With this modification, we are required to at least know the arrangement of the sub-apertures in relation to one-another (although the absolute scale of their layout is not required). In our case, we know that our sub-apertures are arranged as a cross-hair.

Instead of using photometric warp to reconstruct the image of a single sub-aperture, this modification adjusts the pipeline such that instead, we now reconstruct the entire light field. The camera array is a rigid body, meaning that as it rotates and translates through space, the sub-apertures also rotate and translate relative to each other, but in a very well defined way. Say we know the homogenous transform $T_{a \to b}$ of sub-aperture $\mathcal{N}$ between two frames of video, we can compute the

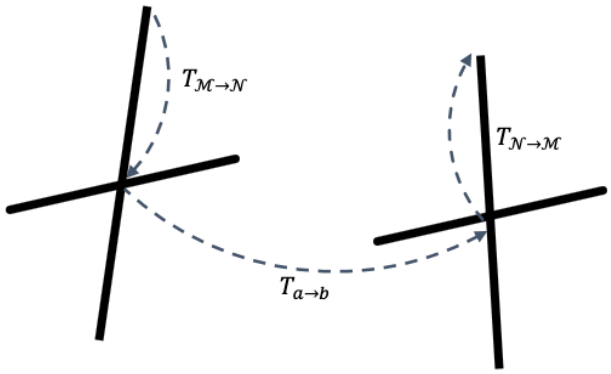


Figure 4.9: Given knowledge of how one sub-aperture has moved through space, we can also compute how any other sub-aperture has moved by chaining the transforms in the order shown.

global transform of any other sub-aperture $\mathcal{M}$ by chaining together the homogenous transforms as shown in Figure 4.9.

To exploit this knowledge, we modify the depth prediction network to now output N depth maps. Each depth map should be aligned with exactly one sub-aperture, so we can reconstruct its corresponding image. The pose network continues to estimate only the pose of the central sub-aperture, but the frame-to-frame pose for each pinhole is computed from the chained transform shown in Figure 4.9 (where $T_{a\to b}$ is the PoseNet output). For each sub-aperture we perform the same photometric warp with bilinear interpolation to synthesise N viewpoints.

In addition to imposing a stronger supervision signal, this pipeline guarantees to a certain extent that our pose and depth modules predict depth at a consistent scale between video frames. We have enforced a photometric warp that requires prior knowledge of the shape of the camera, and with this information now making up a part of the photometric-warp pipeline, our neural networks are forced to learn to use this information to estimate depth and pose at the correct scale. To see why this is true, we should consider the effect on the loss function for an incorrectly estimated scale-factor. We may find that estimating a twice-as-large room for a twice-as-large translation works to photometrically warp the center-view well, but this will be disastrous for each of the other $N-1$ sub-apertures being warped. In short, our networks are penalised harshly for incorrect scale estimates.

Chapter 5

Results

In this chapter, we describe the results from our pose and depth experiments. In Chapter 3 we described an unsupervised pipeline for learning depth and visual odometry from monocular imagery. Similarly, in Chapter 4 we adapted conventional 2D machine learning techniques to apply 4D image data to learning these tasks. Specifically, we recommended two variants of an algorithm for learning depth and visual odometry from light fields - the first uses all the information from the light field to generate a novel rendering of a single U, V slice, while the second method uses the known configuration of the camera array to create a rendering of the complete light field. In this chapter, we contrast results for three algorithms - each of the two novel pipelines described in Chapter 4, and the existing monocular approach described in Chapter 3. For the two algorithms that we have suggested, we also evaluate three ingestion methods: focal-stacks, volumetric images, and tiled EPI images.

First, we evaluate each algorithm's performance in pose-estimation, and demonstrate results for cumulated trajectories for several input-sequences. Subsequently, we'll evaluate the algorithms' performance in depth-prediction and photometric reconstruction.

5.1 Experimental Setup Details

In each of our experiments, we use the same training dataset of 9000 grayscale input images consisting of 48 short video-sequences. The testing set, which is also the same across all experiments, consists of 400 input images over 4 video-sequences. For training we use a batch size of 2 images, and train for a total of 300 epochs.

5.2 Pose Estimation and Visual Odometry

An important metric of error is the *absolute instantaneous error* for each of the 6 degrees-of-freedom that are estimated. This metric describes the magnitude of the error between the predicted and the true component. As described in Chapter 4, we have collected ground-truth pose data using a robotic manipulator platform with a repeatability of ±0.01 millimeters. Meanwhile, we can evaluate each pipeline's effectiveness in estimating the *average cumulative error* by comparing the true trajectory with the predicted trajectory.

5.2.1 Single-View Reconstruction Approach

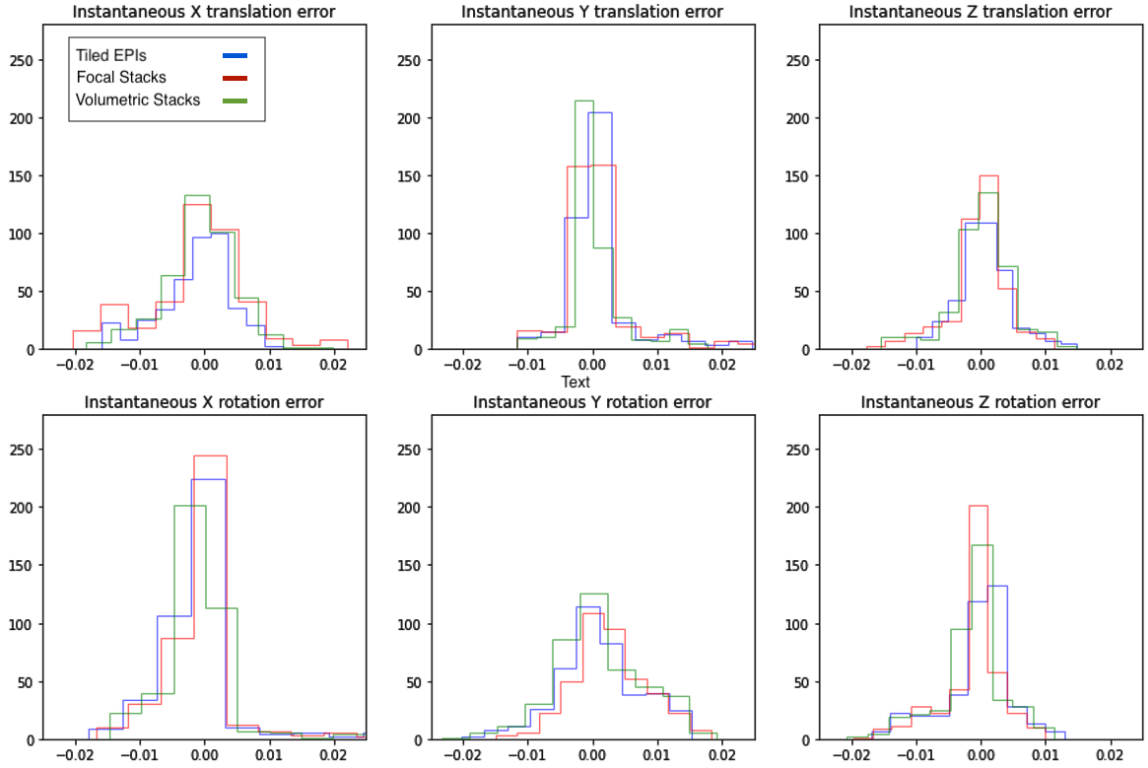
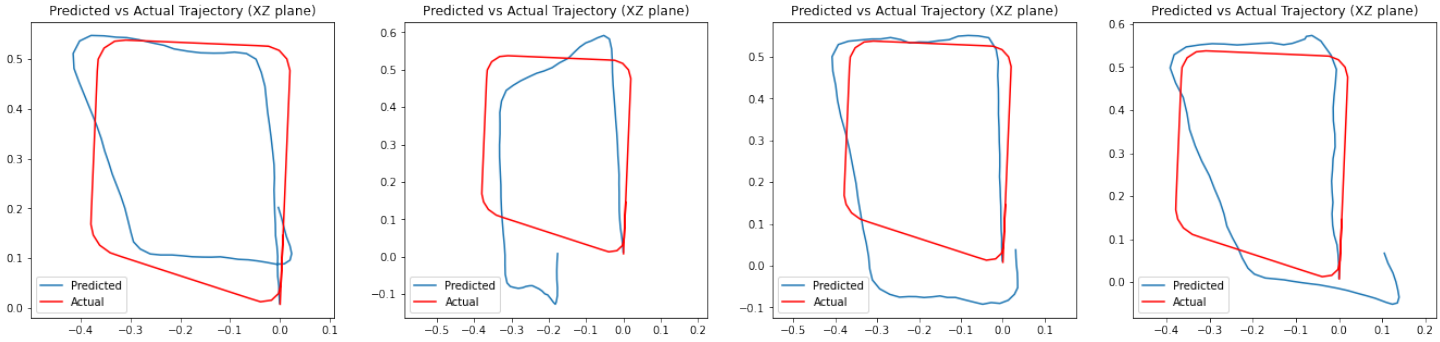


Figure 5.1: Histograms of instantaneous absolute errors, comparing our three input methods. Importantly, while the error histograms exhibit reasonably tight distributions, some outliers exist. Outliers such as these are likely to impact the cumulative error over the complete trajectory, due to the cumulative nature of odometry errors. Comparing these histograms, a decisive winner is not immediately apparent. Table 5.1 summarises the mean errors, from which we might conclude that the tiled EPI format has performed best.

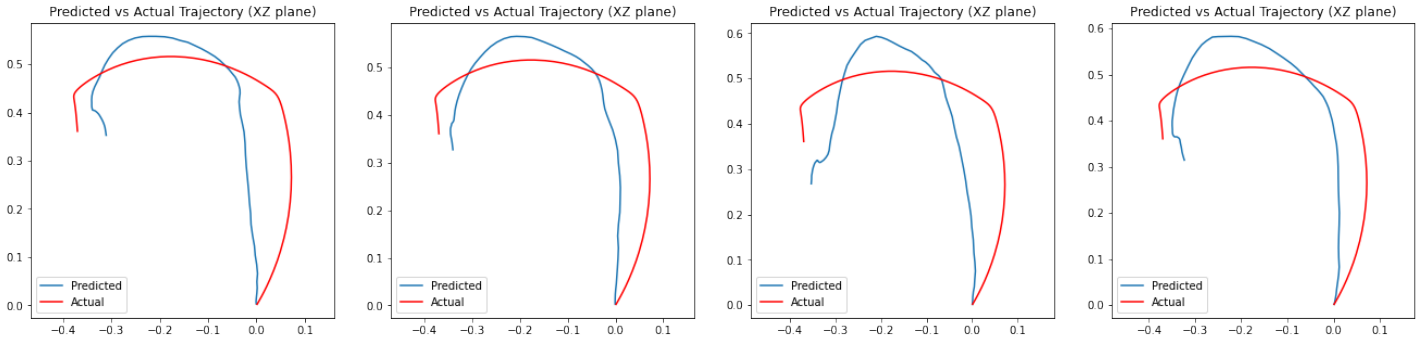
The histogram in Figure 5.1 summarises 400 observations of the absolute instantaneous error for each of the 6 degrees of freedom (translation and rotation in X, Y and Z), over three input methods (tiled epipolar images, focal stacks, and volumetric stacks).

Figures 5.2 and 5.3 show two examples of true and estimated trajectories for the three input methods, as well as the existing monocular method.



(a) Tiled EPIs **(b)** Focal Stacks **(c)** Volumetric Images **(d)** Monocular

Figure 5.2: Estimated and true trajectories for input sequence 16. Inspecting these traces, there is no clear consensus over which input format best reconstructs the path. Qualitatively, we suggest that (b) - Focal Stacks, exhibits the poorest fit for the true path.



(a) Tiled EPIs **(b)** Focal Stacks **(c)** Volumetric Images **(d)** Monocular

Figure 5.3: Estimated and true trajectories for input sequence 44. Qualitatively, these results are competitive with existing approaches. Once again, there is no qualitative consensus over which input format reconstructs the paths best. Inspecting Table 5.1 however we see that quantitatively, on average, tiled EPIs exhibit the smallest error.

Importantly, in Figure 5.2 (d) and 5.3 (d), we have shown trajectories estimated by the existing

monocular approach. Our results using light field imagery are qualitatively competitive with existing monocular approaches, and we demonstrate quantitatively in Section 5.2.4 that our approach using light fields in fact outperforms the monocular approach.

Table 5.1 summarises the instantaneous and cumulative errors for the three input methods.

Table 5.1: Mean Absolute Instantaneous Error and Cumulative Error

Input Method	Absolute Instantaneous Error (mm)	Cumulative Error (mm)
Tiled EPIs	**9.64**	**71.38**
Focal Stacks	9.79	104.03
Volumetric Stacks	12.17	103.47
Monocular	12.90	122.23

The fact that using light field imagery produces improved trajectory results compared to monocular imagery is unsurprising. Multiple view imaging offers rich geometric information about the scene, and our three suggested input methods are all arranged in a way such that simple 2D features relating to parallax and depth may be easily learned by a 2D convolutional filter. Monocular imagery on the other hand forces a convolutional network to learn semantic features relating to depth - we might expect a network trained on monocular imagery to learn roughly what the size of a stuffed animal is, and use the size of the image it forms on the sensor plane to estimate its distance from the camera. These extra steps are not necessary when using multiple view imaging because the depth of the stuffed animal is encoded directly as 2D features that can be learnt, circumventing any of the guesswork that a monocular approach requires. We elaborate more on the difference between monocular and light field imagery in Chapter 6. In the next section, we once again investigate how these input methods perform, after modifying the reconstruction pipeline to reconstruct the complete light field, rather than a single viewpoint from the center of the camera module.

5.2.2 Light Field Reconstruction Approach

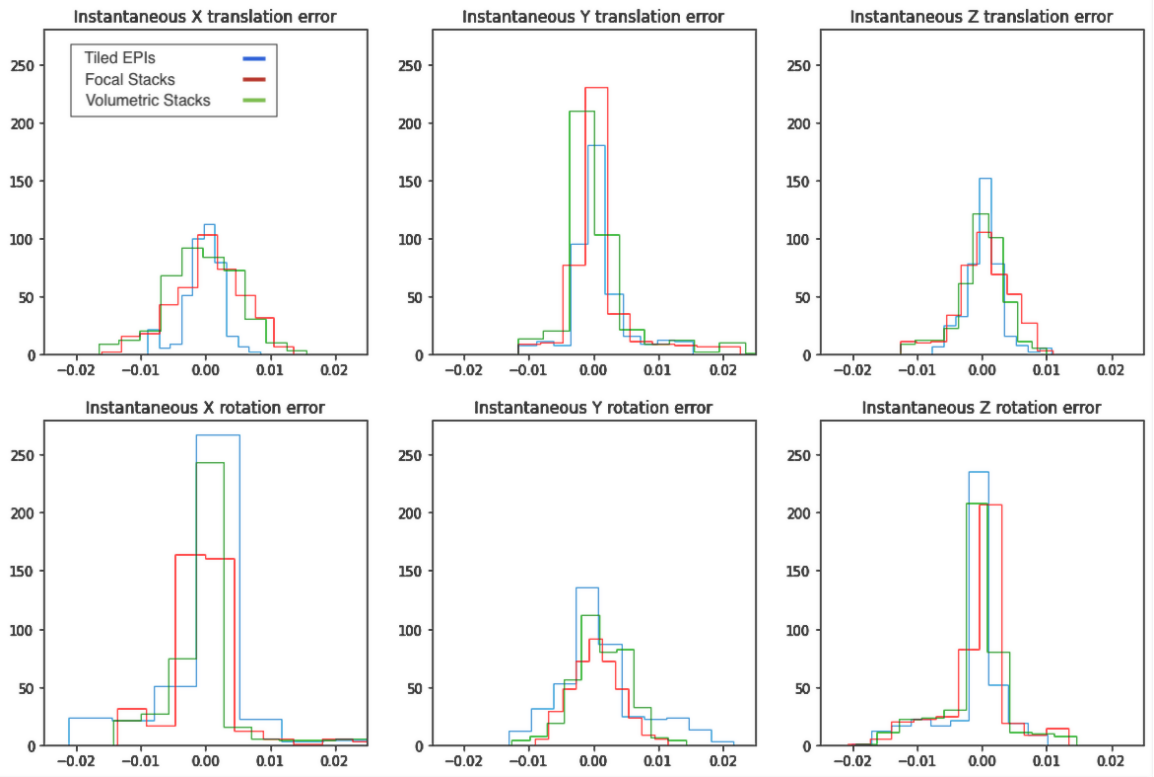


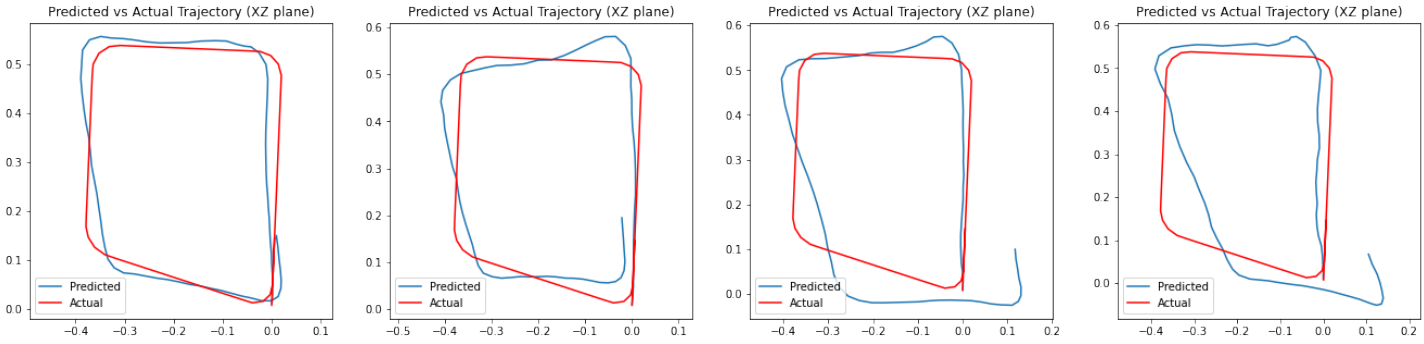
Figure 5.4: Histograms of instantaneous absolute errors comparing our three input methods. In general, we observe that the error distributions using the light field reconstruction approach are generally tighter with shorter tails, compared with the single-view reconstruction approach. We also observe that the distributions of our three input methods indicate that the tiled EPI format performs best - consistently exhbiting the highest peaks around 0.

Here we describe the performance of the second variant of our suggested pipeline, using the known camera configuration to photometrically warp the complete light field. Because this pipeline incorporates known information about the relationship between individual sub-apertures on the camera module, we expect this pipeline to demonstrate improved performance in pose estimation and scale-awareness.

We carry this expectation because this variant of the pipeline generates a stronger supervision signal - the larger number of pixels being warped results in a significantly heavier loss for the same error, compared with the single-view reconstruction approach. Additionally, this variant carries a

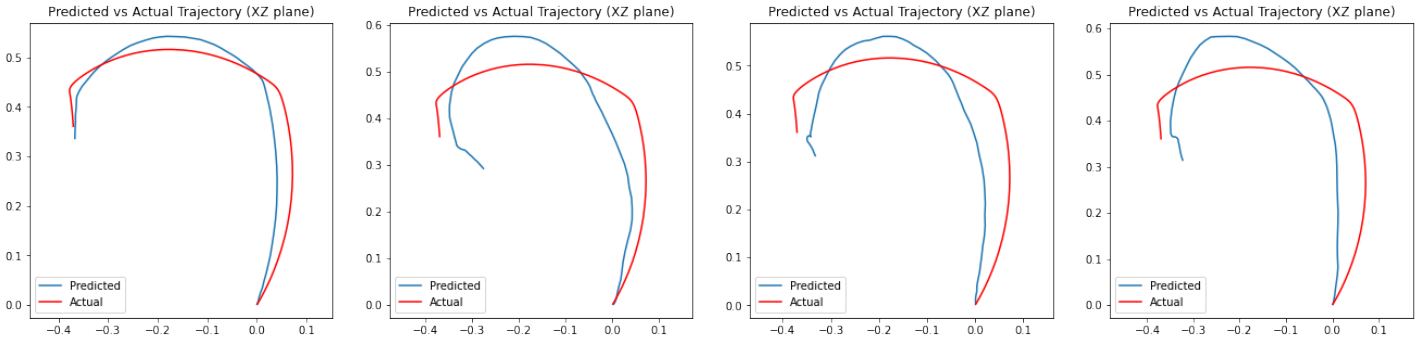
much smaller tolerance to scale inconsistencies due to the constraints imposed when incorporating multiple sub-apertures and their known relative spacing.

Once again, we attach our analysis to a comparison to the monocular approach. Figures 5.5 and 5.6 both show a comparison to the trajectories generated by the monocular approach, as well as our three input methods.



(a) Tiled EPIs **(b)** Focal Stacks **(c)** Volumetric Images **(d)** Monocular

Figure 5.5: Estimated and true trajectories for input sequence 16. Here, we see that tiled EPIs qualitatively outperform the other methods. In general, each of the three input methods using light field imagery are competitive with the existing monocular approach, with Table 5.2 indicating that these methods all outperform monocular imagery.



(a) Tiled EPIs **(b)** Focal Stacks **(c)** Volumetric Images **(d)** Monocular

Figure 5.6: Estimated and true trajectories for input sequence 44. Once again, tiled EPIs exhibit the most visually accurate trajectory, even picking up on the subtle corners that were missed by each of our other input methods.

Table 5.2: Mean Absolute Instantaneous Error and Cumulative Error

Input Method	Absolute Instantaneous Error (mm)	Cumulative Error (mm)
Tiled EPIs	**7.61**	**50.61**
Focal Stacks	9.71	73.40
Volumetric Stacks	10.50	110.04
Monocular	12.90	122.23

5.2.3 Comparison Between Approaches and Summary

For ease of comparison, here we present graphical results for our three input formats, and two reconstruction approaches. Figures 5.7 and 5.8 show instantaneous and cumulative translational errors respectively, while Figure 5.9 shows instantaneous rotational errors. In all cases, tiled EPIs produce the smallest errors.

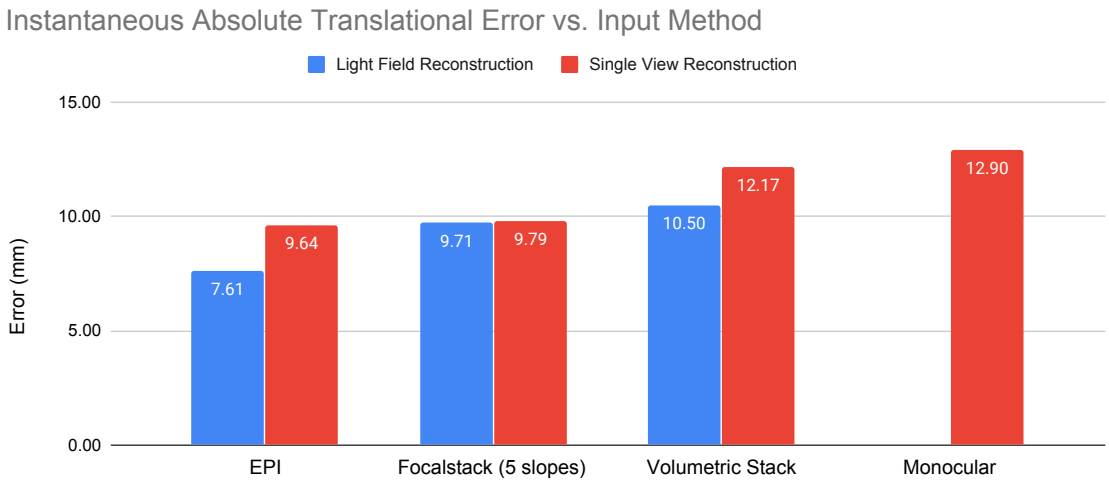


Figure 5.7: We find that in terms of translational error, the light field reconstruction pipeline has an edge over the single-view warp variant in all three cases. Additionally, using tiled EPI's demonstrates the smallest translational error of the three input methods.

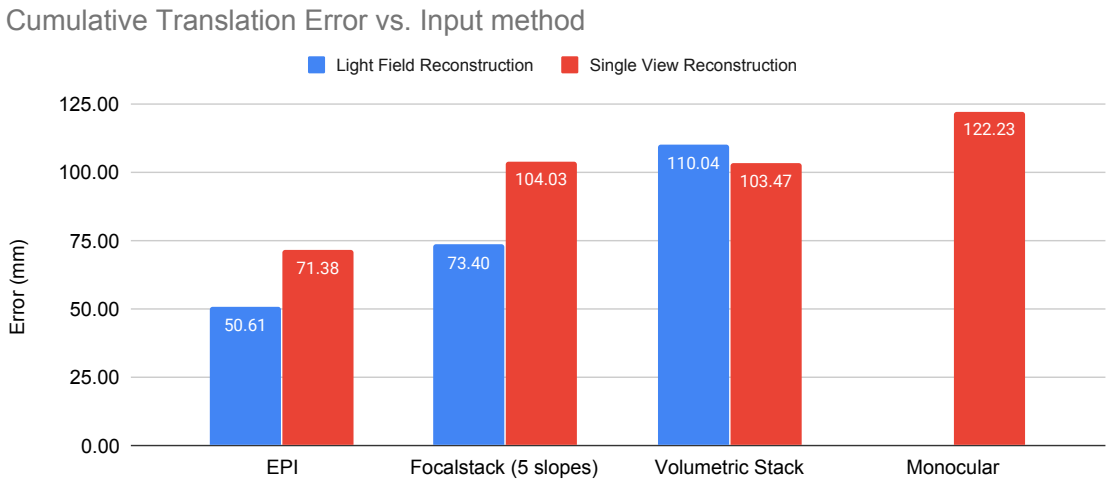


Figure 5.8: Similarly, we find that over the course of an entire video-sequence, tiled EPIs once again outperform all other input strategies. Over an entire video-sequence, the light field reconstruction pipeline generally outperforms the single-warp pipeline.

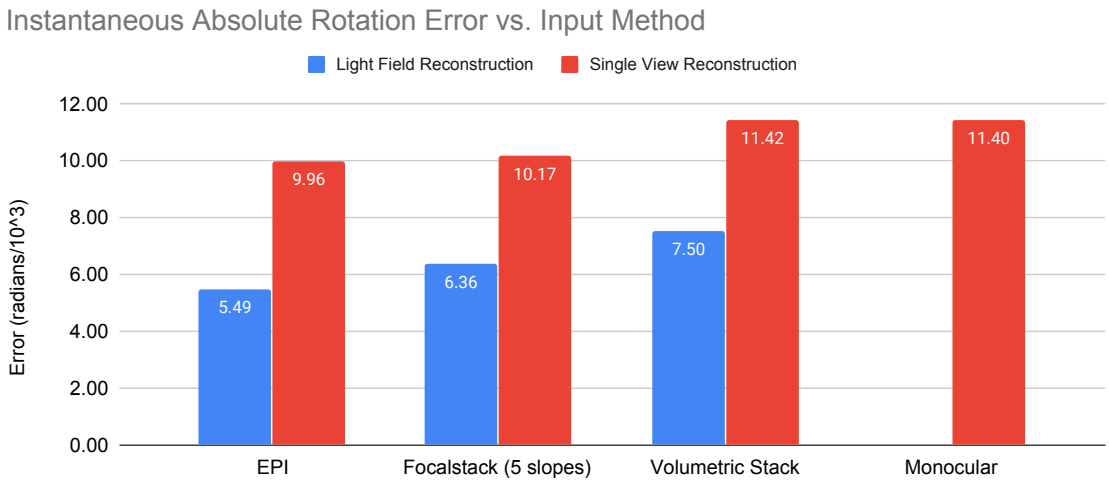


Figure 5.9: We find that in rotational terms, light-field-reconstruction significantly outperforms the single-view-reconstruction pipeline. Similarly, we find that once again, the tiled EPI insertion strategy exhibits the smallest error in terms of rotation on our validation dataset.

5.3 Depth and Photometric Reconstruction

Aside from pose predictions and trajectories, our pipeline also generates depth-map predictions. In this section, we evaluate these outputs. Because our data acquisition methodology did not include the collection of ground-truth depth data, we must find more creative ways of evaluating depth. One way we can do this is by inspecting the visual quality of the depth maps, observing the artifacts and details resolved. For example, in the depth maps shown, we include a particularly challenging scenario that includes a wire-frame model of a flamingo.

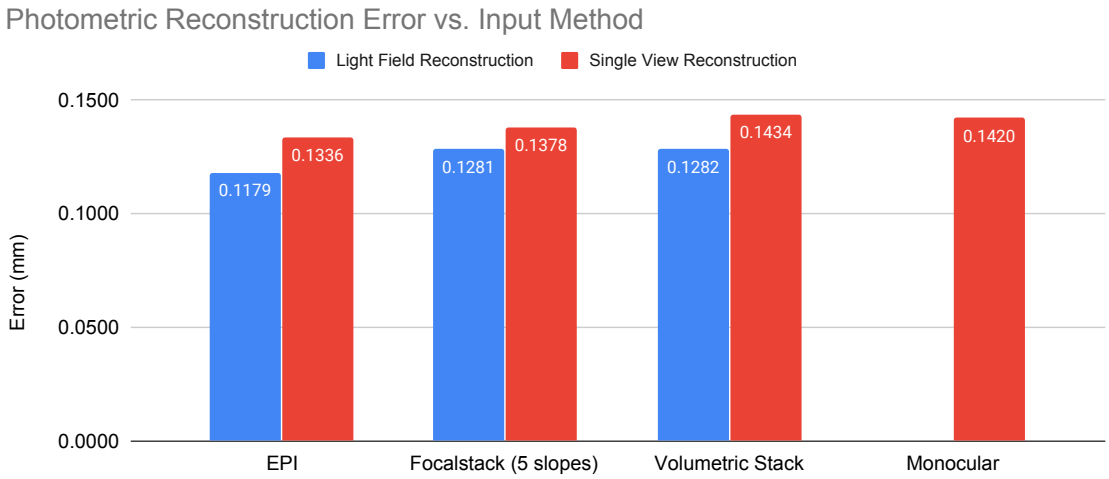


Figure 5.10: The use of light-field-reconstruction to supervise the learning process consistently outperforms single-view reconstruction in terms of photometric error, for all of our tested input methods.

Another way we can evaluate the quality of our depth maps is to compare the photometric loss from image synthesis. Comparing the synthesised image with the ground-truth image, we can compute a difference-image which shows the magnitude of the error in rendering the synthesised image. Photometric error is measured as the difference between the ground truth pixel intensity, and the pixel intensity in the synthesised image, meaning that we can also summarise photometric loss as a single number, which is shown in Figure 5.10 - comparing different input methods.

5.3.1 Single-View Reconstruction Approach

Figure 5.11 shows depth map predictions for our three input methods, over a series of example frames from our testing dataset. Of the three input methods, using tiled EPIs exhibits the smoothest, most visually accurate frames, even picking up on holes in objects, and parts of a wire-frame flamingo.

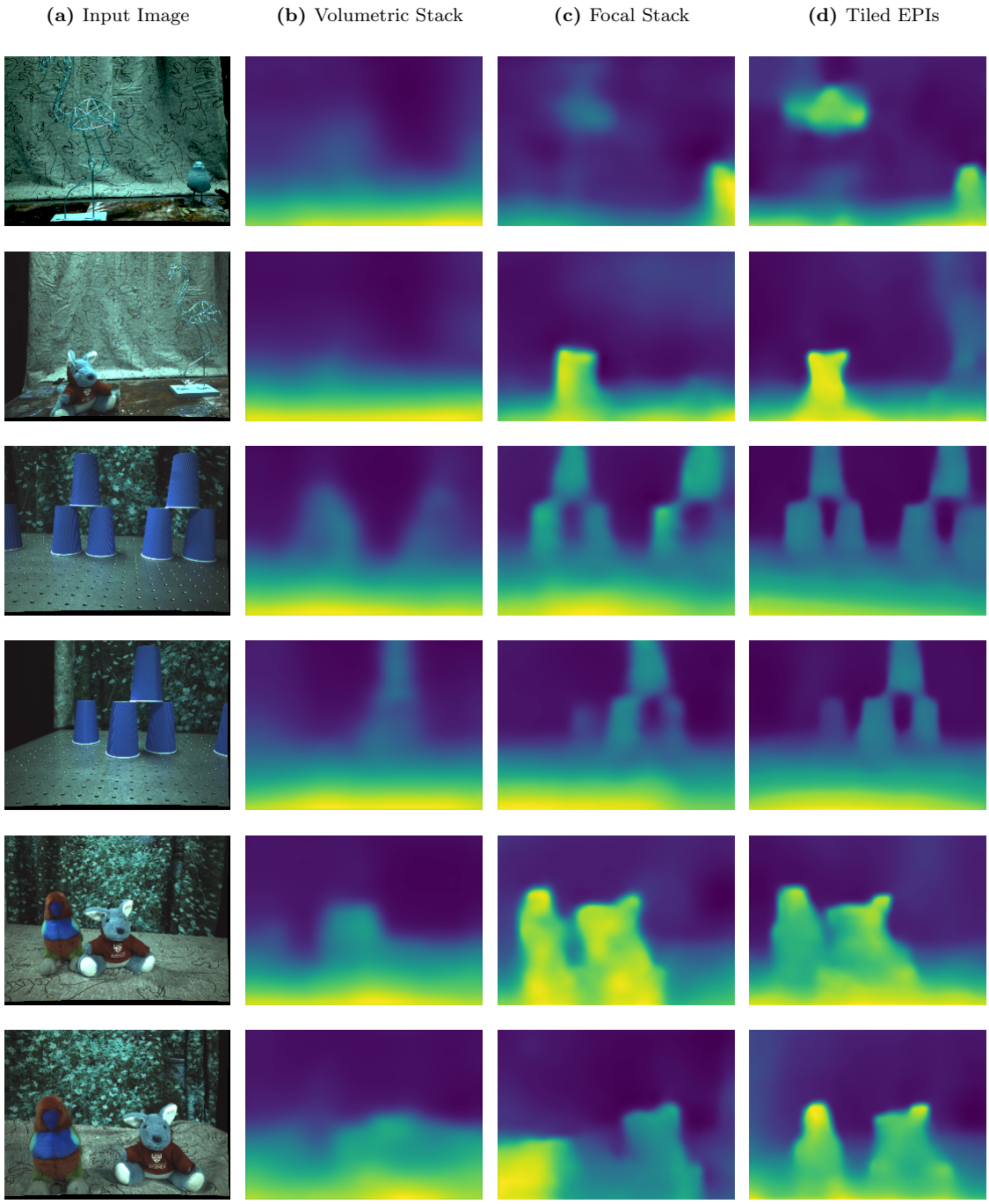


Figure 5.11: Comparing depth maps estimated using each of our three input methods using single-view reconstruction, the tiled EPI input format appears to give the most visually accurate predictions, with edges resolved clearly, even capturing the appearance of the wireframe flamingo (first row) and some of the holes in between objects.

5.3.2 Light Field Reconstruction Approach

Similarly, we now show depth predictions when light field reconstruction is used for training the networks. Notably, only the output layer of the disparity network has changed - in other words, the only major difference is the formation of a supervising loss function. We see from Figure 5.12 that the depth maps are qualitatively more convincing with details resolved far more clearly.

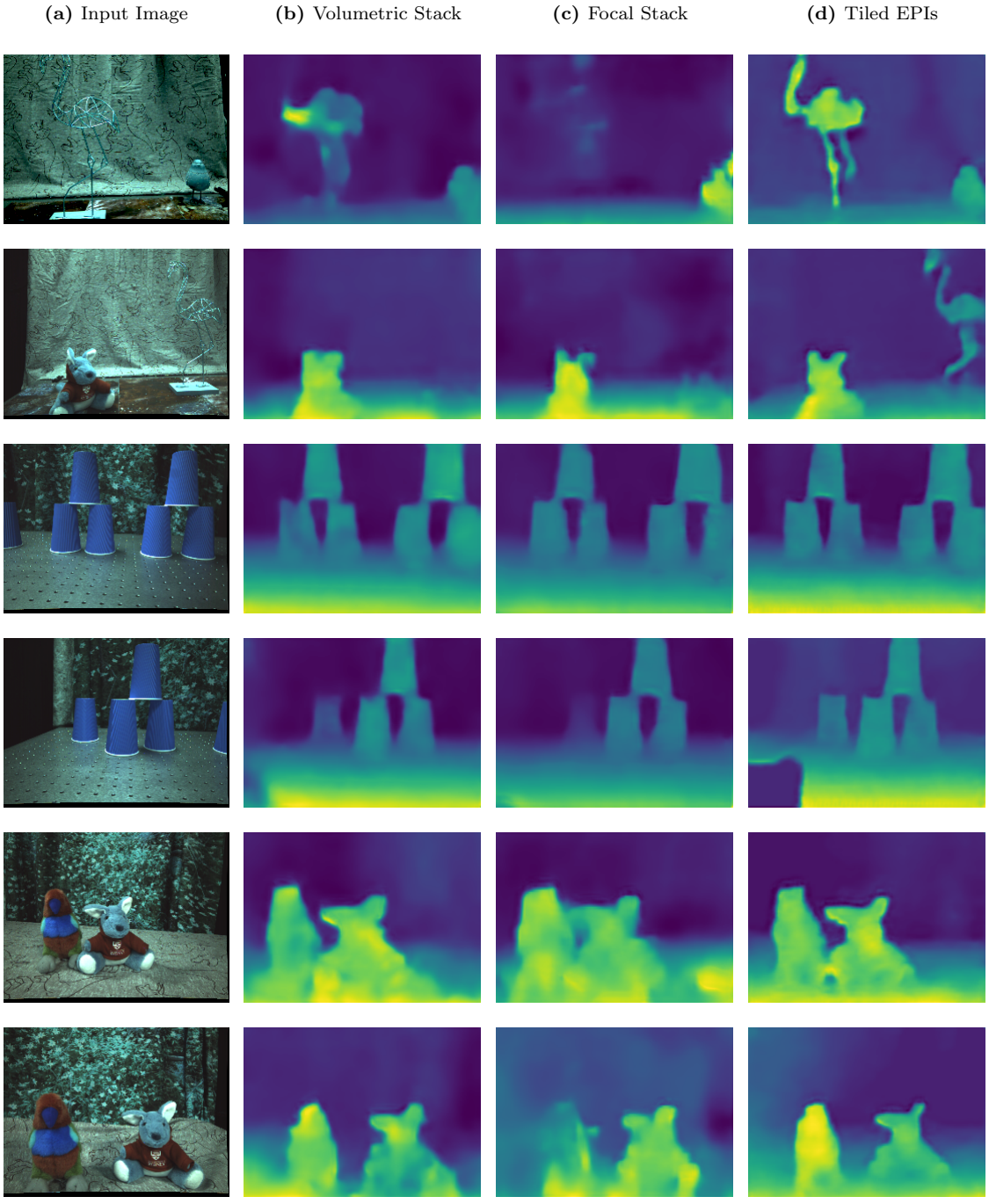


Figure 5.12: Compared to the single-view reconstruction pipeline, these depth maps using the light field reconstruction pipeline resolve edges and details far more clearly. The tiled EPI input format once again captures the details of the wireframe flamingo most convincingly.

5.3.3 Monocular Approach

We observe from Figure 5.13 that depth estimates using monocular imagery show far less detail and that the depth maps appear much smoother. This is to be expected, as no 3D information is preserved in a 2D image, meaning that only learned 2D features can be used to infer depth.

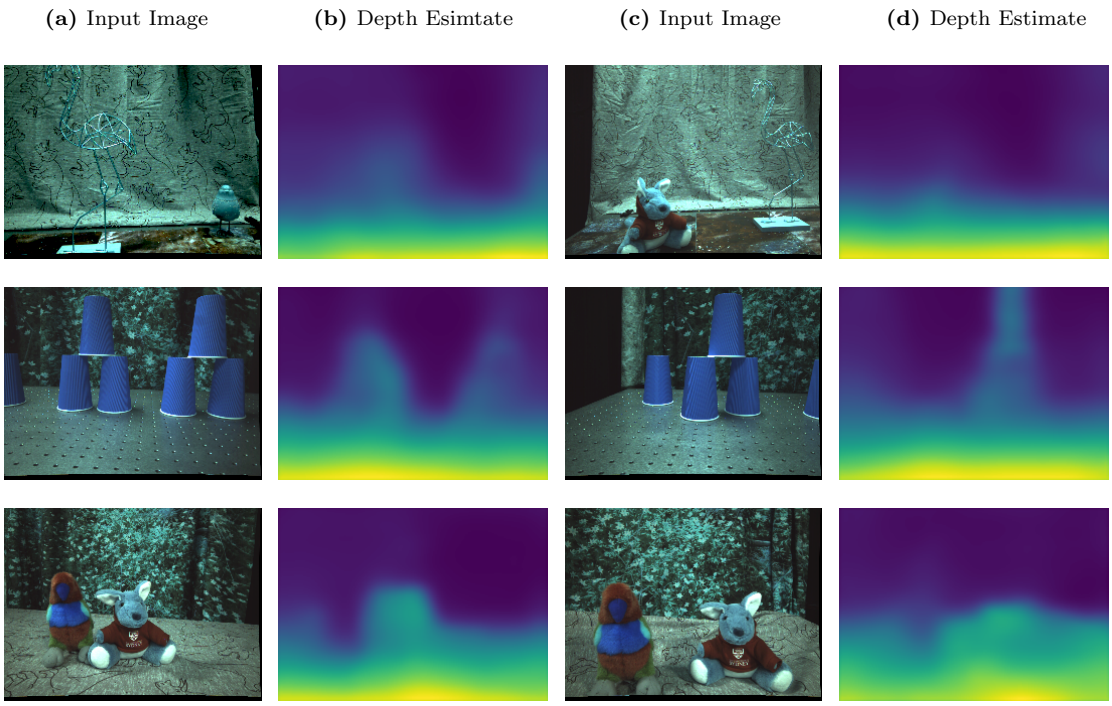


Figure 5.13: Unsurprisingly, monocular imagery produces the least visually convincing depth maps. Simple features like the flatness of the table and blobs of objects in front of the camera appear, but edges and discontinuities are not captured well.

5.3.4 Qualitative Analysis

Another useful result for inspection is the *difference image* - this shows the pixel-wise difference between the photometrically warped output of our algorithm, and the target image. Inspecting the two images side-by-side and discerning the difference is a challenging task, and so with the aid of the difference image we make the difference between the two images clearer.

Below, we show some examples of the target image (the image to be reconstructed), the output from photometric reconstruction, and the difference image. We show examples for the best-

performing version of our algorithm - using epipolar plane images as the input format with light-field-reconstruction used as the training pipeline.

(a) Target Image **(b)** Reconstructed Image **(c)** Difference

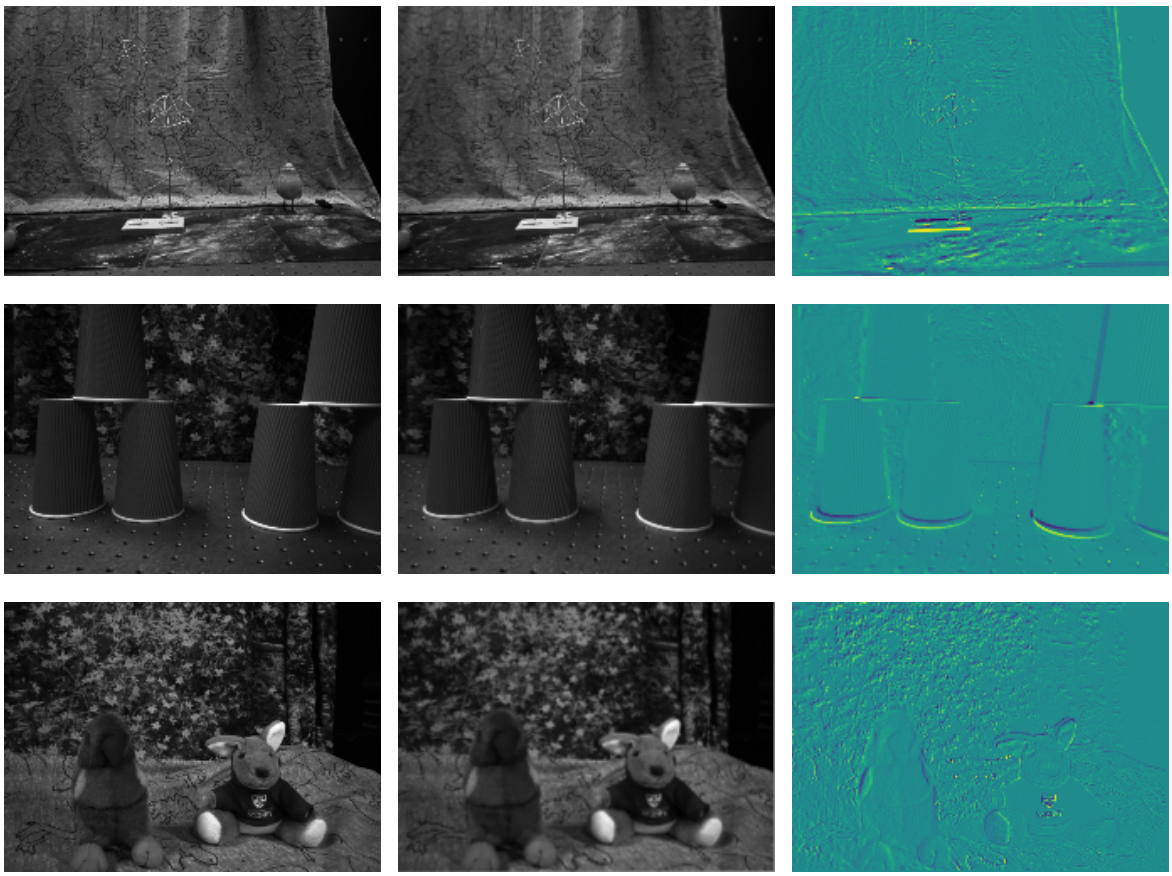


Figure 5.14: Difference images reveal where the largest errors occurred in photometric synthesis. The most apparent errors occur around 3D discontinuities and high-texture parts of the image.

Notably, we observe that the reconstructed image exhibits reduced clarity compared to the target image. Not unexpectedly, we find that the largest errors occur around 3D depth discontinuities - where occluders coming into and out-of view affect the correctness of our image-based sampling strategy. Similarly, textured regions of the image also exhibit heavy losses, as even small errors in the pose and depth estimate exacerbate the photometric warp error around these regions.

5.4 Summary of Results

Table 5.3 is a summary of results from this chapter, including odometry and photometric errors. Algorithmic bandwidth is an important consideration in real-time systems such as robotics, and so we also report the time required for both inference and training. We report the execution time for Python code running on an Intel i5 4460 CPU at 3.2 GHz, with tensor computations taking place on a GeForce GTX 1050Ti GPU. Inference is near-realtime for full framerate video, supporting up to 5 frames per second, which is competitive with the existing monocular approach (which we measure at just under 10 frames per second). The following acronyms are used in the table headings.

- AITE: Absolute instantaneous translational error
- AIRE: Absolute instantaneous rotational error
- CE: Cumulative translational error
- PWE: Photometric warp error

Table 5.3: Summary of Odometry and Photometric Warp Errors, and Execution Times

Input Method	AITE (mm)	AIRE (mm)	CE (mm)	PWE	Infer (ms)	Train (hrs)
Single View Reconstruction Pipeline						
Tiled EPIs	9.64	9.96	71.38	0.1336	154	17.9
Focal Stacks	9.79	10.17	104.03	0.1378	205	25.0
Volumetric Stacks	12.17	11.42	103.47	0.1434	112	17.6
Light Field Reconstruction Pipeline						
Tiled EPIs	**7.61**	**5.49**	**50.61**	**0.1179**	199	29.5
Focal Stacks	9.71	6.36	73.40	0.1281	264	30.4
Volumetric Stacks	10.50	7.50	110.04	0.1282	121	25.4
Monocular Approach						
Monocular Image	12.90	11.40	122.23	0.1420	**110**	**17.6**

Chapter 6

Discussion

In Chapter 4, we hypothesised that of our two suggested learning pipelines, the variant that photometrically reconstructs the complete light field is likely to out-perform either the monocular or single-view-reconstruction approach. Our experimental results in Chapter 5 largely confirmed this hypothesis, and in this chapter we investigate these results in more detail, drawing conclusions from our data and relating these to our hypothesis.

Not only were our suspicions with respect to the learning pipeline confirmed, but we also showed that of the three insertion methods, using tiled epipolar plane images produced the best pose-estimates and the smallest photometric warp error. We present a discussion in this chapter around the impact that different insertion methods have.

We recognise that deep neural networks are opaque in nature - their millions of parameters makes any meaningful understanding of their internal mechanisms challenging to understand. However, we do know from decades of machine learning research that convolutional networks act as feature-extractors - and so our methodology has focused on providing data to these networks with the most informative features presented first. Regardless, what we present in this chapter is only a reasoned understanding of the most likely impacts of the different input formats and learning pipelines.

The first part of this chapter discusses the compromises between our three unsupervised learning variants. The second part compares the different insertion methods, investigating the likely causes for the disparate results between them. Finally, we compare our results to state-of-the-art monocular approaches.

6.1 Comparison Between Unsupervised Learning Pipelines

In this section, we present a discussion on the different impacts of the two unsupervised learning pipelines (monocular, single-view-reconstruction and light-field-reconstruction). Specifically, we address four trade-offs which impact the performance of our convolutional networks - the strength of the supervision signal, the enforcement of scale consistency, pipeline flexibility, and the computational cost.

6.1.1 Strengthening the Supervision Signal

One question that remained, prior to the analysis of our results, was whether the photometric reconstruction of the entire light-field might lead to the unstable propagation of errors through our network. As the number of pixels being reconstructed increases, so to does the magnitude of the loss function. As the loss signal grows, there remains a possibility that the partial derivatives of the network's parameters with respect to the loss grow large, causing the optimisation algorithm to apply unstable updates to the network's parameters. We had prepared strategies for mitigating this instability in the case that it did occur, such as reducing the learning-rate and gradient-clipping, however we found that training progressed smoothly using the light-field reconstruction approach. We conclude from this that not only is photometrically reconstructing the complete light-field a stable formation of a loss function, but it is indeed beneficial to the learning of depth and visual odometry. Supporting this conclusion, we observe that the light-field-reconstruction pipeline supports our networks in locating a lower local minimum, with an average of a 10.3% reduction in photometric error compared to single-view-reconstruction across all our experiments.

Importantly, we distinguish between simply magnifying the supervision signal, and using light-field-reconstruction to obtain a heavier loss function. If we relied on the former strategy, we might simply increase the learning-rate to penalise errors more harshly - which leads to instability in training. We conclude that by photometrically reconstructing the 4D light field, we teach our networks something far more useful about geometry and the light field that enables them to locate a smaller minimum in the topographic loss landscape.

6.1.2 Enforcing Scale Consistency in Translation and Rotation

Aside from applying a heavier supervision loss signal to the two networks, the light-field-reconstruction method also penalises scale inconsistencies. We discussed in Chapters 3 and 4 the weaknesses of

monocular approaches - because the information used to generate the photometric warp is sourced from a single pinhole view of the scene, we are restricted to 2D data. When adjusting the approach to reconstruct the entire 4D light field however, our networks are required to learn the relationship between the sampled image data and the geometry of the scene. In the monocular case, a depth network in the absence of better information will happily estimate a twice-as-large room, if of course the pose network estimates a twice-as-large translation. Notably however, this does not extend to rotations - the image formed by scaling the rotation of the camera is not supplemented by scaling the magnitude of the depth map. It is surprising then, that the light-field-reconstruction method produces the largest boost in performance in terms of rotation, not in translation. We observe that when using the light-field-reconstruction pipeline, the rotation error estimate is on average 39% better across all input methods. Translational errors on the other hand experienced only a 9.7% performance improvement.

We speculate that this performance boost arises from the combination of a stronger supervision signal and the enforcing of scale consistency through the 4D warp. The stronger supervision signal helps the two convolutional networks find a smaller local minimum in the loss topography, while the scale-consistency constraint provides improved certainty in the scale of the translation, the geometry of the scene, the relationship between sub-apertures, and the ability to resolve depth in the scene. Rotational errors are more likely to exacerbate the photometric loss compared to translational errors, and through this combination of factors, we believe that our pose network has learned a more robust function for estimating rotations.

6.1.3 Compromising Between Pipeline Flexibility and Performance

An important capability that arises from our pipeline is the ability to perform visual odometry and depth estimation using an *uncalibrated* camera array. The attractiveness in this approach is that the algorithm is suitable for use in mobile robotics which are frequently subjected to temperature change, vibration and shock. This ideally means that we have minimal prior knowledge about the camera being used - however we acknowledge that the light-field-reconstruction pipeline sacrifices some flexibility, as this variant requires prior knowledge of the *shape* of the camera being used. However, we also argue that even in the light-field-reconstruction case, the absolute magnitude of the relationship between sub-apertures are not required. While depth and pose estimates will indeed be adversely affected by these phenomena, we argue that even as the calibration error of the camera array accumulates, the 'best' solution to the photometric-warp problem (i.e. the global minimum

for the photometric warp loss) converges to the correct pose and depth estimate. This is an error that cannot be ignored using stereo or even trinocular imagery, but as the number of sub-apertures increase, the 'best' photometric warp solution converges to that generated using the correct pose and depth.

6.1.4 Computational Cost

Both of our suggested pipelines operate on arbitrarily large images, meaning images may be cropped and scaled, as long as the intrinsic matrix K is also scaled appropriately. The execution time for inference using both algorithms is predictable and repeatable as long as the dimensionality of the input remains constant, even as the model is trained and fine-tuned on new data. This is an attractive capability in robotics, where prior knowledge of an algorithm's bandwidth greatly simplifies system design.

Of our algorithms, the best performance for depth and pose estimation used the light field tiled as EPIs as the input. This performance bump is however accompanied by a tradeoff in computational runtime - processing around 5 frames per second. The poorest performing variant - monocular imagery - executes in the shortest amount of time at around 10 frames per second. This is illustrative of a classic tradeoff in system design - performance versus resource consumption. With performance optimisations and the right hardware however, we expect that it should be possible for our tiled EPI approach to achieve real-time performance.

6.2 Comparison between Input Methods

One of the more exciting investigations to be conducted in this work is on the utility of different representations of the light field for neural networks in geometry-based tasks. In this section we compare the performance of our three insertion strategies, and speculate on the success of the different methods. Previously, we described the challenge as finding the best way of organising 4D data, as 2D slices in an informative and meaningful way for learning depth and visual odometry. Experimentally, we found that the use of tiled epipolar plane images produced the best performance of our three tested methods.

6.2.1 Performance of Epipolar Plane Images

Epipolar plane images explicitly encode depth in the 'slope' of their characteristic sheared lines, which is known as the gradient-depth constraint. Not only does this provide an explicit feature-set for a neural network to learn geometry from, but the derivatives of this 4D space when extended into the time domain allows motion to be directly inferred, as we saw in the *plenoptic flow* algorithm.

We attribute the success of this input format to the clear expression of depth information, through simple 2D features which are easily decoded through the 2D convolution of signals. Furthermore, our method includes information from three different feature spaces - in U, V, as well as S, U and T, V. Now S, U and T, V are similar feature spaces in that the depth information contained in both represents the same scene geometry. U, V slices on the other hand depict a pinhole-projection of the scene, allowing for the rich textural details and semantic features of the scene to be learnt. By combining these feature spaces, we suggest that our method is capable of learning to approximate the 4D signal processing functions required for performing depth and pose estimation.

Recall that our EPI insertion strategy includes the use of a feature extractor which preprocesses the tiled EPI prior to feeding to our depth and pose networks. The purpose of this encoder module was to down-sample the resolution of our tiled images. We recognise however that this is an additional 'learnable' module that effectively increases the number of learnable parameters in both our depth and pose networks.

6.2.2 Performance of Volumetric Stacks

Stacking the U, V slices volumetrically is a simple way of formatting the light field data - its formation is easy to understand, and the computational cost is relatively cheap. This format contains raw, unprocessed data from the light field. In performance, we found that volumetrically stacking the images exhibited poorer performance than either tiled EPIs or focal stacks. We suggest that the performance did not match the other formats for two reasons. Firstly, the data is unprocessed and does not re-slice the light field to present any meaningful geometric features first - instead this approach is a simple extension of the monocular case, including several monocular views of the scene in the channels dimension. It is possible that a 2D convolutional filter may learn features relating to parallax and occlusion through the channel dimension, but the convolutions take place in the 'angular' dimension rather than the 'spatial' dimension. Secondly, our tiled EPI input format has a clear advantage over either of our two other formats in that it is able to take advantage of 3 feature

spaces. Volumetrically stacking the images on the other hand, only allows our networks to learn simple 2D convolutions in a single feature space.

6.2.3 Performance of Focal Stacks

Focal stacks are highly configurable - aside from choosing the number and the depth of the focal planes comprising the stack, the depth-of-field may also be configured by varying the number of sub-apertures used to construct it.

In our experiments we tested two variants - in both cases we employed all sub-apertures (this has the effect of making a very thin focal plane), and tested the results when focusing at 5 versus 9 planes. To our surprise, the 5-plane experiment exhibited the superior odometry results. Our expectations for this input format are that objects' depths are encoded in the focal-plane which they occupy - i.e. whether they are resolved clearly or blurrily at a specific focal depth.

This is a surprising result, as we expect the number of possible depth encodings to be greater with the 9-plane variant. We suggest three ideas that may have given rise to this result. The first is a matter of experimental setup - the specific depth planes that were chosen to construct the stack. It may be the case that our choice of 9 focal planes simply introduced redundant information that was already encapsulated in the 5-plane data, making it more difficult for the network to select useful features. Secondly, the construction of focal stacks assumes a texturally rich scene - it is difficult to tell whether regions of low texture are in or out of focus. Thirdly, as we have seen, using sparse camera arrays for synthetic aperture focusing is prone to aliasing, creating artifacts in the image data - which is clearly visible in Figure 4.6. With a larger number of focal planes being used, the volume of artifacts, and potentially detrimental non-existent features interpreted by our depth and pose networks increases.

While we have suggested these three ideas for how this result may have arisen, we believe it is most likely a combination of several factors. We remain unconvinced that the provision of additional data through focus stacking is a detrimental factor in learning geometry-related tasks, and recommend future work to investigate more precisely the impacts of varying the number of planes, the depth-of-field, and the types of scenes being imaged.

6.3 Comparison to Existing Monocular Approaches

Our experimental data indicates that, unsurprisingly, our approach using light field data outperforms the state-of-the-art monocular approach described in [62]. We expected this to be the case, as the data we have collected and provided to our networks is far more informative than simple monocular imagery when used for geometry based learning. Our models outperform the monocular approach not only in depth and odometry errors, but also in the average photometric warp error - each image that is synthesised is a better match for the target image using our pipeline than in the monocular pipeline.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Throughout this work, we have advocated for robust, data-driven, computer-vision algorithms which will enable the deployment of autonomous robotics in unconstrained environments.

In this work we identified an opportunity to address a knowledge gap - namely the ability to perform visual odometry using an uncalibrated camera array. Meanwhile, we have blended the successes of two promising technologies - light field imaging and deep learning - to address the visual odometry problem. Not only does our proposed algorithm build robustness through experience, but it has been designed to operate on minimal prior knowledge of its parent system, adapting to adverse environments and improving performance *in situ*.

In this work, we have provided evidence that computational imaging - the co-design of optics and algorithms - will continue to propel advancements in imaging technology for the foreseeable future. Most of us need not look any further than our own pockets to find a piece of technology that is emblematic of the progress computational imaging has made. Since we commenced this work, the typical number of cameras embedded in flagship mobile devices has evidently graduated from two to three (and four in some devices) - which is incidentally enough to acquire a sparse 4D light field.

7.1.1 Applications

We envisage a number of potential applications for our algorithm. Primarily, it has been designed as an odometry component for autonomous systems, simplifying the process of navigation, localisation

and mapping. Robotic systems rarely operate in the absence of feedback data, especially in intelligent, mobile systems. Where robotics is concerned, we frequently hear about sensor-fusion - a simple and practical way of combining multiple sources of truth to build the best probabilistic picture of the robot's operating environment. A simple application is hovering-in-place using an AUV in a GPS denied area - a problem that in any practical scenario cannot be solved with dynamic system models alone, requiring some form of feedback data. Feedback using visual odometry is good, but feedback using a fusion of visual odometry and inertial measurements for example would be better.

(a) iPhone 11 Pro **(b)** Samsung Galaxy S20 Ultra **(c)** Pelican Imaging Camera Array

Figure 7.1: Consumer electronics producers are embracing computational imaging, with the number of cameras on many mobile devices now exceeding 3, not only allowing users more control over their photography but enabling those in the computer vision community to continue innovating.

While we have primarily focused on robotics, we also recognise that as computational imaging becomes more pervasive in consumer electronics, numerous potential applications of our algorithm emerge. For example, the ability to track camera motion while simultaneously reconstructing a 3D model of the scene has applications in Augmented Reality, enabling 3D models to be rendered over the top of a real world scene. Similarly, the ability to predict camera motion from one frame to another may be employed to smooth motion in a shaky video. In a similar vein, video frames may even be interpolated using the photometric warp technique that we have used for training our models - effectively increasing the frame rate of the footage.

7.2 Future Work

There are numerous exciting avenues of research to branch off this work, which we describe in the following. In suggesting these research questions, we begin with more immediate extensions of the present work, and build up to longer-term research goals.

7.2.1 Real-time Inference and Online Learning

We have suggested throughout this work that our algorithm is capable of online learning. A simple yet exciting extension of this work is to implement a system that is capable of such online learning, perhaps on a mobile robotic platform or visually equipped quad-rotor. Depending on the implementation requirements, this may mean minifying the computational cost of the algorithm to enable training on the edge device, or alternatively modifying the training routine to operate in the cloud whilst collecting data simultaneously from several devices. Deployment options may also be explored - weighing up the comparable advantages of inference at the edge versus in the cloud. Devices such as the Coral [1] and Jetson Nano [2] have seen 'edge' computing rise in popularity, while the scalability of cloud platforms has similarly made their services very popular.

7.2.2 Fusion and Filtering

Light field imaging is just one expression of computational imaging - meanwhile, RGB-D sensors, lidar and time-of-flight cameras can, and should be used to expand the versatility of our system. Current limitations of our algorithm - such as the challenges presented by low-texture materials - may be partially addressed by the fusion of multiple visual sensors. This makes sense especially in larger robotic platforms such as autonomous vehicles and industrial robotics - where safety and robustness are key. In a similar vein of research, we suggest that integration with existing SLAM algorithms is a useful research direction that will more rapidly enable the adoption of such algorithms in mobile robotics.

7.2.3 Deep Learning and Light Field Imaging

Deep learning is no longer an obscure area of research - software libraries and hardware-accelerated devices are making it more accessible than ever, inviting even the most idly curious to participate. Similarly we have witnessed a maturing of light field technology, with applications in robotics, consumer electronics and industry. Applications which have previously found success applying deep learning to conventional imagery now stand to also benefit from the rich 3D data captured by light field cameras. In combination, these technologies present exciting possibilities, and the potential applications are limited only by their broader adoption, and the ingenuity of the people who use it.

[1] https://coral.ai/products/dev-board/

[2] https://developer.nvidia.com/embedded/jetson-nano-developer-kit

Bibliography

[1] Edward H. Adelson and James R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20. MIT Press, 1991.

[2] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):698–700, May 1987.

[3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[4] Stefan Behrendt. *Isometric projection.* Aug 2006.

[5] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992.

[6] Jia-Wang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. 2019.

[7] Robert C. Bolles and H. Harlyn Baker. Readings in computer vision: Issues, problems, principles, and paradigms. chapter Epipolar-plane Image Analysis: A Technique for Analyzing Motion Sequences, pages 26–36. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[8] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian D. Reid, and John J. Leonard. Simultaneous localization and mapping: Present, future, and the robust-perception age. *The Computing Research Repository*, 2016.

[9] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145 – 155, 1992. Range Image Understanding.

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] Erwin Coumans and Yunfei Bai. *Pybullet, a Python module for physics simulation in robotics, games and machine learning.* 2017.

[12] Donald Dansereau. *Plenoptic Signal Processing for Robust Vision in Field Robotics.* PhD thesis, Australian Centre for Field Robotics, 2014.

[13] Donald Dansereau, Ian Mahon, Oscar Pizarro, and Stephan B. Williams. Plenoptic flow: Closed-form visual odometry for light field cameras. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4455–4462, 2011.

[14] Donald Dansereau, Oscar Pizarro, and Stefan Williams. Linear volumetric focus for light field cameras. *ACM Transactions on Graphics*, 34:1–20, 03 2015.

[15] Fengchun Dong, Sio-Hoi Ieng, Xavier Savatier, Ralph Etienne-Cummings, and Ryad Benosman. Plenoptic cameras in real-time robotics. 32(2):206–217, 2013.

[16] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *The Computing Research Repository*, abs/1406.2283, 2014.

[17] Á. Faluvégi, Q. Bolseé, S. Nedevschi, V. Dădârlat, and A. Munteanu. A 3D convolutional neural network for light field depth estimation. In *2019 International Conference on 3D Immersion (IC3D)*, pages 1–5, 2019.

[18] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[19] Paul Verlaine Gakne and Kyle O'Keefe. Tackling the scale factor issue in a monocular visual odometry using a 3D city model. 2018.

[20] Ravi Garg, Vijay Kumar B. G, and Ian D. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. *The Computing Research Repository*, 2016.

[21] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 2013.

[22] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *The Computing Research Repository*, 2016.

[23] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. *The Computing Research Repository*, 2018.

[24] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 43–54, New York, NY, USA, 1996. ACM.

[25] Wolfram Gothe. *Three-point perspective projection.* Jul 2009.

[26] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[27] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[28] S. Heber and T. Pock. Convolutional networks for shape from light field. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3746–3754, 2016.

[29] Berthold Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society A*, 4:629–642, 04 1987.

[30] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2017–2025. Curran Associates, Inc., 2015.

[31] Jian Jiao, Jichao Jiao, Yaokai Mo, Weilun Liu, and Zhongliang Deng. Magicvo: End-to-end monocular visual odometry through deep bi-directional recurrent convolutional neural network. *CoRR*, abs/1811.10964, 2018.

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger,

editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[33] Andreas Kuehefuss, Niclas Zeller, Franz Quint, and Uwe Stilla. Feature based rgb-d slam for a plenoptic camera. 2016.

[34] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[35] Titus Leistner, Hendrik Schilling, Radek Mackowiak, Stefan Gumhold, and Carsten Rother. Learning to think outside the box: Wide-baseline light field depth estimation with epi-shift. *2019 International Conference on 3D Vision (3DV)*, Sep 2019.

[36] Mark Levoy and Pat Hanrahan. Light field rendering. In *Proc. ACM SIGGRAPH*, 1995.

[37] Orly Liba, Kiran Murthy, Yun-Ta Tsai, Tim Brooks, Tianfan Xue, Nikhil Karnad, Qiurui He, Jonathan T. Barron, Dillon Sharlet, Ryan Geiss, and et al. Handheld mobile photography in very low light. *ACM Transactions on Graphics*, 38(6):1–16, Nov 2019.

[38] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian D. Reid. Learning depth from single monocular images using deep convolutional neural fields. *The Computing Research Repository*, 2015.

[39] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *The Computing Research Repository*, 2014.

[40] H. C. Longuet-Higgins. *A Computer Algorithm for Reconstructing a Scene from Two Projections*, pages 61–62. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[41] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[42] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *The Computing Research Repository*, abs/1512.02134, 2015.

[43] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, USA, 1969.

[44] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[45] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–652–I–659 Vol.1, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[46] Seymour Papert. The summer vision project. 1966.

[47] Pedro F. Proença. *Robust RGB-D Odometry under Depth Uncertainty for Structured Environments*. PhD thesis, University of Surrey, 2018.

[48] Frank Rosenblatt. *Principles of Neurodynamics*. Spartan Books, 1959.

[49] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[50] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[51] Changha Shin, Hae-Gon Jeon, Youngjin Yoon, Inso Kweon, and Seon Kim. Epinet: A fully-convolutional neural network using epipolar geometry for depth from light field images. 04 2018.

[52] K. A. Skinner and M. Johnson-Roberson. Towards real-time underwater 3D reconstruction with plenoptic cameras. In *2016 International Conference on Intelligent Robots and Systems*, pages 2014–2021, Oct 2016.

[53] Neal Wadhwa, Rahul Garg, David E. Jacobs, Bryan E. Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T. Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *CoRR*, abs/1806.04171, 2018.

[54] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. *CoRR*, abs/1709.08429, 2017.

[55] Ting-Chun Wang, Jun-Yan Zhu, Hiroaki Ebi, Manmohan Chandraker, Alexei A. Efros, and Ravi Ramamoorthi. A 4D light-field dataset and CNN architectures for material recognition. *The Computing Research Repository*, 2016.

[56] Xing Sun, Z. Xu, Nan Meng, E. Y. Lam, and H. K. . So. Data-driven light field depth estimation using deep convolutional neural networks. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 367–374, 2016.

[57] Li Yao, Yunjian Liu, and Weixin Xu. Real-time virtual view synthesis using light field. *EURASIP Journal on Image and Video Processing*, 2016(1):25, 2016.

[58] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian D. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. *The Computing Research Repository*, 2018.

[59] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces, 1994.

[60] Dingfu Zhou, Y. Dai, and Hongdong Li. Reliable scale estimation and correction for monocular visual odometry. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 490–495, 2016.

[61] Dingfu Zhou, Yuchao Dai, and Hongdong Li. Ground plane based absolute scale estimation for monocular visual odometry. *The Computing Research Repository*, 2019.

[62] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017.