Enabling Plug-and-Play Cameras:

Generalisable Methods for Self-Calibration and Multi-Modal Vision Systems

Ryan Ben Griffiths

A thesis submitted in fulfillment
of the requirements of the degree of
Doctor of Philosophy

THE UNIVERSITY OF
SYDNEY

Australian Centre for Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

March 2026

Statement of Originality

This is to certify that the content of this thesis is my own work. This thesis has not been submitted for any other degree or purpose.

I certify that the intellectual content of this thesis is the product of my own work, and that all assistance received in preparing this thesis and all sources have been acknowledged.

Ryan Ben Griffiths

March 2026

Abstract

Vision systems are foundational to a wide range of real-world applications, including autonomous vehicles navigating complex urban environments, drones performing infrastructure inspection, and robots operating in hazardous or remote settings. These applications increasingly depend on diverse and specialised camera hardware, such as fisheye, thermal, and multimodal systems, which challenge the assumptions of conventional computer vision pipelines. Existing approaches typically require labour-intensive calibration, handcrafted adaptation for each camera type, and large labelled datasets. These limitations hinder the deployment of custom vision systems that provide critical robustness and performance. The need for increasing diversity of camera hardware is clear. This thesis addresses the central question: how can we build plug-and-play vision systems that generalise across camera types and sensing modalities with minimal human intervention? To this end, we present three key contributions in the areas of self-calibration, network adaptation, and multimodal fusion.

First, we introduce NOCaL, a semi-supervised framework that jointly estimates camera intrinsics, distortion, and odometry using a rendering-based self-supervision signal. By leveraging light field neural representations and a hypernetwork architecture, NOCaL achieves accurate camera parameter estimation and egomotion using unlabelled data. Second, we propose RectConv, a deformable convolutional layer that enables pretrained convolutional neural networks to operate on previously unseen camera geometries such as fisheye lenses. This is achieved without retraining or explicit image rectification. Third, we develop a transformer-based architecture for multi-camera, multi-modal integration. The model introduces a ray-based rotary positional embedding that encodes both viewpoint and modality information. This embedding enables effective integration of RGB and thermal imagery into a shared, geometrically consistent scene representation.

Extensive experiments on real and synthetic datasets show that these methods reduce the need for labelled data, manual calibration, and camera-specific engineering. Together, they demonstrate that ray-based, self-supervised representations can support flexible and generalisable vision systems that adapt to new hardware and sensing configurations. The contributions of this thesis have potential impact in domains where robust perception is critical, such as autonomous navigation, environmental monitoring, planetary exploration, and field robotics. By reducing the cost and complexity of deploying custom vision systems, this work helps pave the way towards more adaptable, accessible, and intelligent machine perception.

Acknowledgements

This thesis would not have been possible without the help of a great number of people. First and foremost, my supervisor Dr. Donald Dansereau, who has been extraordinarily generous with his time, knowledge and support; also for his invaluable feedback over the past years. Additionally, I would like to thank Don for instilling in me a genuine intrigue for computer vision and research, without whom I might never have discovered.

Thanks goes to the Robotic Imaging Group with all of its members past and present. They have provided a fruitful and engaging environment to discuss and get feedback on ideas and challenges. It has been a pleasure to work with every single one of them.

I would like to express my gratitude to students and researchers at the Australian Centre for Robotics. It has been a supportive and creative environment while I completed my PhD. I would also like to thank the administrative staff for their support and care. I never would have been able to navigate the university administrative system without them.

Lastly, I would like to thank my friends and family for their support throughout the years; to Paul for generously proofreading this thesis; and a special thanks to my partner, Lily, for her unconditional belief in me.

This research was supported in part through the NVIDIA Academic Grant Program. This research was also undertaken with the assistance of resources from the National Computational Infrastructure (NCI Australia), an NCRIS enabled capability supported by the Australian Government. Providing crucial compute resources.

Thanks to EPIImaging LLC for their support in providing hardware.

Financial support was provided by an Australian Government Research Training Program (RTP) Scholarship, as well as being supported in part by funding from Ford Motor Company.

During the preparation of the thesis ChatGPT was used as a tool for text editing, uses included paraphrasing, sentence structuring and spelling. Where text was modified by generative AI, the content was reviewed for possible errors, inaccuracies, and bias. All ideas and content presented in this thesis were conceptualised without the aid of AI.

Author Attribution Statement

Parts of this thesis are based on the following:

- Chapter 3 of this thesis is published as [40], R. Griffiths, J. Naylor, and D. G. Dansereau, "NOCaL: Calibration-free semi-supervised learning of odometry and camera intrinsics" In IEEE International Conference on Robotics and Automation (ICRA), pages 4056–4062, 2023.
  I designed the study, analysed the data and wrote the drafts of the manuscript.
- Chapter 4 of this thesis is published as [38], R. Griffiths, and D. G. Dansereau, "Adapting CNNs for Fisheye Cameras without Retraining" In IEEE International Joint Conference on Neural Networks (IJCNN), 2025
  I designed the study, analysed the data and wrote the drafts of the manuscript.
- Chapter 5 of this thesis is published as [39], R. Griffiths, and D. G. Dansereau, "RoRE: Rotary Ray Embedding for Generalised Multi-Modal Scene Understanding" In The International Conference on Learning Representations (ICLR), 2026
  I designed the study, analysed the data and wrote the drafts of the manuscript.

In addition to the authorship attribution statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Ryan Ben Griffiths

March 2026

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Dr. Donald Dansereau

March 2026

This thesis is dedicated to the approximately 3200 *cups of tea it took to complete.*

Contents

List of Figures

List of Tables

Nomenclature

List of Acronyms

AbsRel	absolute relative error
CNN	convolutional neural network
DPT	dense prediction transformer
FFT	fast fourier transform
FOV	field-of-view
IMU	inertial measurement unit
IoU	intersection over union
LFN	light field network
LPIPS	learned perceptual image patch similarity
LWIR	long-wave infrared
MLP	multi-layer perceptron
MIoU	mean intersection over union
MSE	mean squared error
NeRF	neural radiance fields
NUC	non-uniformity correction
NVS	novel view synthesis
PSNR	peak signal-to-noise ratio
ReLU	rectified linear unit
RectConv	rectified convolution
RMSE	root mean squared error
RoPE	rotary positional encoding
SSIM	structural similarity index measure
STN	spatial transformer network
ViT	vision transformer

Chapter 1

Introduction

"The eye, the window of the soul, is the chief means whereby the understanding can most fully and abundantly appreciate the infinite works of nature."

— Leonardo da Vinci

Vision is critical in almost all tasks and is widely regarded as humans' most important sense. In a world increasingly moving toward automation in all domains, computer vision already plays and will continue to play a central role in advancing this transformation. The primary objective of this thesis is to make progress towards *plug-and-play* cameras. A philosophy that deploying cameras in diverse situations should be as easy as possible, without the need for redesigns or specialist knowledge.

1.1 Motivation

Cameras have become indispensable sensing modalities across a wide array of disciplines, from robotics and autonomous vehicles to medical imaging and environmental monitoring. In manufacturing, machine-vision cameras inspect parts at micron-level precision; in agriculture, drones equipped with multi-spectral cameras monitor crop health across vast fields; in healthcare, endoscopic and fluorescence cameras guide surgeons through minimally invasive procedures. As these applications grow in complexity and scale, the limitations of one-size-fits-all vision systems become more apparent for real-world applications: different tasks demand different optical properties, spectral sensitivities, and fields of view. To push the boundaries of automation, whether it be a self-driving car navigating crowded city streets or a service robot operating in unstructured human environments, we must carefully evaluate what kinds of vision systems are most appropriate, and how they should be configured for specific tasks and contexts.

This work is concerned with learning how to interpret and use the data collected from these different vision systems in efficient ways without having to redesign our calibration procedures, interpretation architectures, and models for each individual system. The key motivation for this work is to allow easier deployment of custom vision systems which enable more robust and reliable autonomous platforms and vision systems.

1.1.1 What is Vision Anyway?

When seeking inspiration for artificial vision, we can turn to the myriad solutions that evolution has crafted. In the animal kingdom, eye placement and structure often reflect an animal's ecological niche: predators such as eagles sport forward-facing, high-resolution eyes optimised for depth perception, while prey species like rabbits exhibit laterally placed eyes offering nearly 360° peripheral vision [34]. Compound eyes, found in insects, take a radically different approach: a fly's eye comprises some

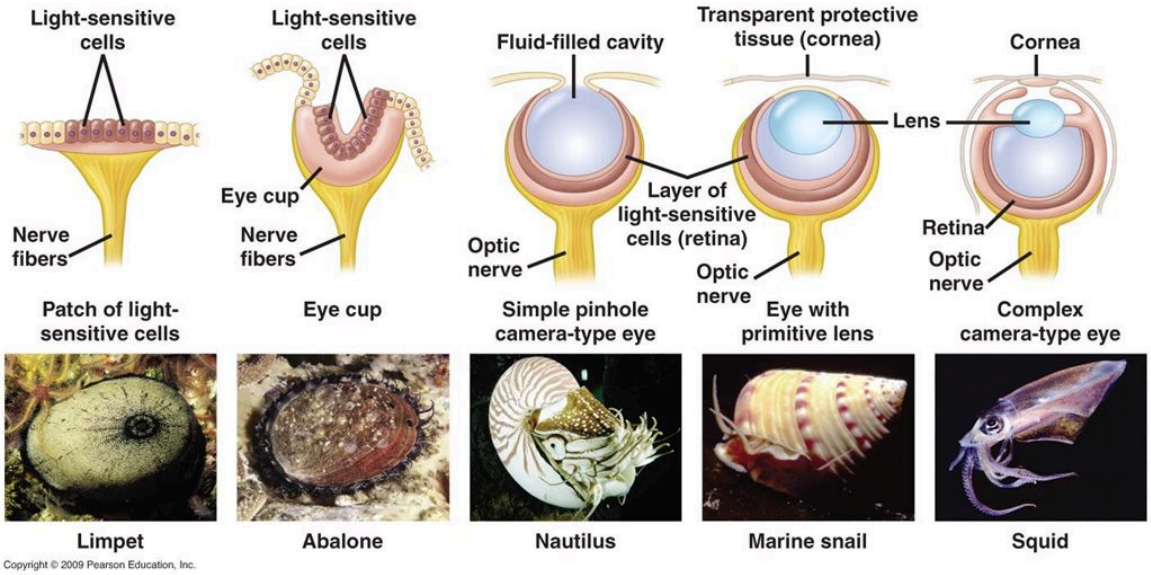

Figure 1.1 – Example in the animal kingdom with different vision. Here we show examples in the marine world of different biological optical systems. Looking at progressively more complex optics for different families of animals [9].

3,000 individual ommatidia, each sampling a tiny portion of the visual field [67]. This mosaic arrangement trades spatial resolution for exceptional motion sensitivity and a very wide field of view, enabling flies to detect and react to even the slightest changes in their surroundings in a fraction of a second. Figure 1.1 show an illustration of the myriad of vision solutions found by evolution [9].

This diversity in eye architecture illustrates a broader principle: visual systems evolve to meet the specific perceptual demands of their environment and behaviour. The high-resolution, front-facing eyes of predators support targeting and tracking, while the panoramic vision of prey species maximises early threat detection. Insects like flies prioritise rapid motion detection, at the expense of fine spatial detail . These examples demonstrate that there is no single, universally optimal way to 'see'; rather, the form and function of a visual system are tightly coupled to the organism's ecological tasks, sensory constraints, and available neural processing resources.

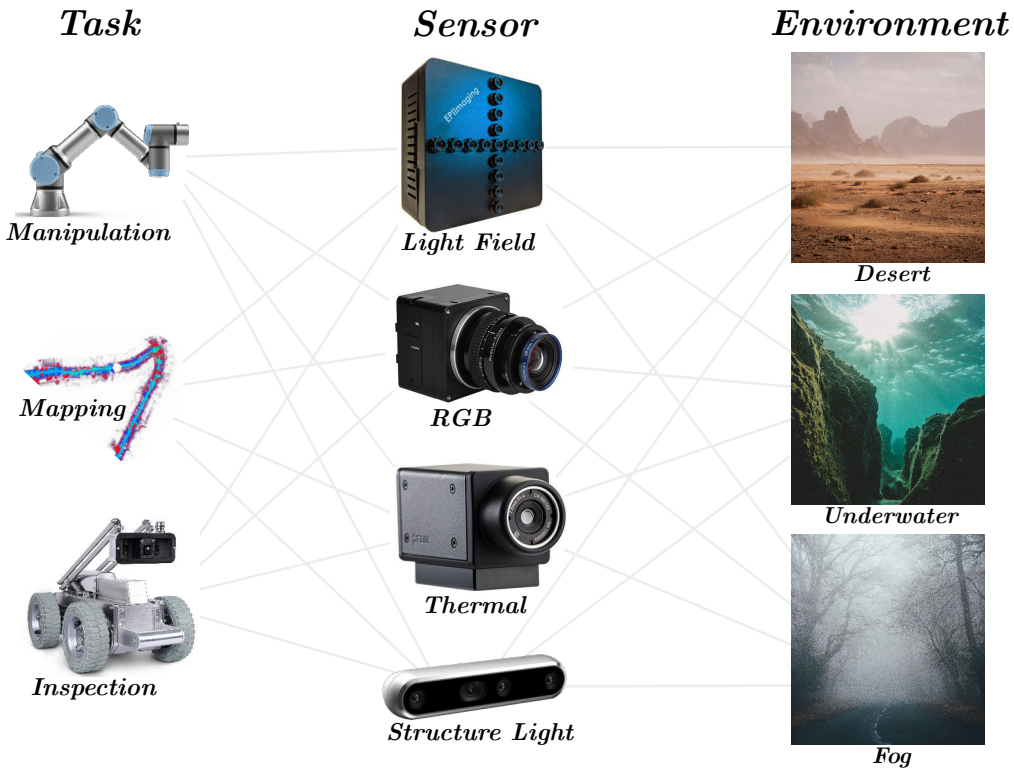

Figure 1.2 – Illustration of the number of combinations even for a limited number of tasks, sensor types and environments. There are many numbers of possible combinations of these, each with their own unique challenges from a vision perspective

1.1.2 Designing to See

As humans we often design things in our own image. This is true when it comes to cameras, our vision systems are designed to be comprehensible by humans. However, in parallel with biological insights, the way humans see is only one of many possible options. In this work we take the philosophy that to design robust artificial vision systems, they must adopt an end-to-end perspective: from the ultimate task requirements through to the sensor hardware, calibration models, and downstream algorithms [83]. This holistic approach is inherently challenging; engineers must reason about lens selection, sensor resolution, distortions, spectral bands, and mechanical mounting, all while ensuring that the resulting image data can be processed effectively.

When designing these vision systems we have the unique opportunity to join multiple heterogeneous cameras such as thermal and RGB. Pairing these cameras enables critical benefits and redundancy that would not be possible from a single homogeneous cameras [70]. This, however comes with another set of challenges that have been an active area of research for decades: how do you effectively fuse dissimilar sensor information into one coherent understanding of the environment? Depending on the environment (e.g. day vs. night), one modality might be more reliable than the other, meaning this fusion needs to understand when the different sensors are, each in their own way, trustworthy or not [6]. Suffice to say, with the countless combinations of different cameras and different environments manually designing for every configuration is infeasible. An illustrative example of different camera systems and environments is shown in Figure 1.2. Even in this simplified example there is a huge number of possible combinations.

Designing new cameras is a complex, multidisciplinary challenge and a rich area of ongoing research, encompassing innovations in optics, sensor materials, and spectral imaging. This thesis does not aim to contribute directly to that domain. Instead, it focuses on how to process and integrate the output from diverse camera systems, treating camera design as an upstream component.

Traditionally, computer vision processing pipelines are handcrafted for a specific task, sensor and environment. Any change in optics or sensor often necessitates laborious recalibration and new bespoke signal-processing pipelines. If a handcrafted processing pipeline had to be designed for each possible combination in order to see which is the best vision system to pick for a given application and environment, it would be practically impossible to test all options. Further as these systems move from research to products which are mass produced, reducing reliance on manual or time consuming operations such as calibration becomes of increasing importance. Additionally, it is important to note that custom imaging systems have no available labelled datasets, unlike conventional imaging systems which have many existing publicly available datasets.

We propose moving to a more flexible paradigm which designs algorithms to be agnostic to the precise camera characteristics: instead of hard-coding intrinsics or distortion models, vision methods could learn directly from sensor data. This not only provides the ability to adapt to new cameras and modalities without explicit calibration, it also provides the opportunity to discover new capabilities. By leveraging advancements in machine learning with its ability to ingest and reason in a data-driven way there is a path to move toward truly plug-and-play vision systems that simplify deployment and accelerate innovation.

1.2 Problem Statement

Despite the tremendous progress in camera hardware and computer vision algorithms, deploying new sensors in real-world systems remains a labour-intensive process. A key challenge in achieving plug-and-play vision is the scarcity of labelled data for many sensor configurations and environments. Unlike large-scale image benchmarks that provide dense ground-truth annotations, most real-world deployments yield only unlabelled imagery, often without reliable measurements of camera intrinsics, extrinsics, or scene geometry.

Our hypothesis is that new data-driven methods with reduced labelled data can automate critical deployment tasks: camera calibration, sensor integration, data interpretation, and robust multi-modal fusion. This will overcome existing limitations and address the goals outlined above. Our methods aim to recover intrinsic and extrinsic parameters, reconcile disparate sensor modalities, and feed normalised data into generic vision algorithms. Such a paradigm not only reduces the engineering overhead of deploying new cameras but also allows vision systems to adapt dynamically to changing sensor configurations and environmental conditions.

This is a broad problem with many avenues and nuances. This thesis tackles two specific areas associated with this broader goal:

1. Removing barriers to deploying new sensors, including reduction of bespoke algorithms and computationally expensive tasks; and

2. Reducing the reliance on densely labelled data, as it is a manual and time consuming task that is a barrier to deployment when not available.

The ultimate goal of this thesis is to realise truly plug-and-play vision systems: platforms that can accept arbitrary cameras and continue operating seamlessly, without bespoke engineering or human intervention.

1.3 Contributions

The primary focus of this thesis is to remove barriers to the deployment of cameras, addressing challenges with calibration, data constraints and multimodal fusion. We introduce a range of specific methods that tackle these challenges of deployment.

These methods include the following contributions:

Semi-supervision for Joint Calibration and Odometry using Light Field Networks - partially published as [40]

- A novel architecture that jointly learns camera parameters and odometry with minimal prior knowledge of the camera being used;

- A novel self-supervision signal which uses ray based neural rendering, in our case a light field network (LFN), that can take advantage of a large amount of unlabelled data;

- A semi-supervision training scheme that, while still benefiting from the large amount of unlabelled data, allows the training process to benefit from a small amount of labelled data, that constrain the odometry to metric space using semantic cues; and

- Improved performance in odometry compared to unsupervised methods while also performing calibration, demonstrating the efficacy of the semi-supervised approach.

Adapting Convolutional Neural Networks for New Camera Geometries Without Retraining - partially published as [38]

- A novel convolutional layer, RectConv, that enables networks to natively handle previously unseen camera geometries without requiring retraining or reprojection of input imagery;
- An approach for automatically adapting existing convolutional networks to RectConv networks, allowing pre-trained networks to be applied with new cameras; and
- Comparison with naive and rectification-based methods, showing improved performance for wide-FOV images on multiple network architectures, cameras, and tasks.

Ray Based Rotary Embeddings for Multi-Modal, Multi-Camera Integration

- A novel transformer architecture that uses a ray based rotary positional embedding that allows multi-camera systems to be jointly integrated into a single representation;
- A multi-modal training scheme that allows for a single model to accept different configurations of modalities across an arbitrary number of cameras;
- A unified network that enables a robust vision pipeline that is resilient to occlusions due to its masked patched structure;
- Comparison to state-of-the-art rgb-only alternatives, showing faster convergence, and comprehensive evaluation of multi-modal performance under different operating configurations.

Parts of this work also appear in [27]

These contributions each address a distinct barrier to the deployment of new camera systems, which can be broadly categorised into three areas: camera calibration; the adaptation of existing methods to new camera geometries; and the integration of multiple cameras. Despite targeting different challenges, the contributions share common themes. First, they all emphasise the reduction of reliance on labelled data, which has been identified as a critical obstacle to deployment. Second, each contribution demonstrates that understanding images in terms of light rays gives us the expressive capability required to achieve each goal.

Collectively, these contributions advance the thesis's central goal of making vision systems easier to deploy and more robust to variation in sensor configuration and environment. We believe this work helps lower the barrier for researchers and practitioners building autonomous systems for challenging, dynamic, and unstructured environments, such as underwater exploration, disaster response, and planetary robotics.

1.4 Outline

Chapter 2 lays the groundwork by surveying the key concepts and methodologies on which this thesis builds. We begin with camera fundamentals; models of projection, target-based and self-calibration techniques, ray representations, light-field parameterisation, and diverse sensor modalities such as RGB, thermal, and event cameras. We then introduce neural rendering approaches, contrasting implicit radiance-field methods with feed-forward light-field networks and positional encoding strategies that underpin our algorithms. Finally, we discuss core vision applications, odometry, depth estimation, novel-view synthesis and the evaluation metrics (trajectory error, depth error, photometric loss) used throughout this work.

The first technical Chapter 3 presents NOCaL, a semi-supervised framework that jointly learns camera intrinsics, lens distortion, and relative pose from unlabelled video, with only minimal motion cues (e.g., wheel odometry or inertial measurement

unit (IMU)) providing metric scale. A pretrained light-field hypernetwork renders novel views, enabling self-supervision for both calibration and odometry. We demonstrate that NOCaL achieves real-time inference with accuracy on par with classical target-based and fully supervised methods, while requiring no external calibration targets.

Chapter 4 is building on the limitations of applying standard convolutional neural networks (CNNs) to wide-field or non-perspective imagery, this chapter introduces RectConv: a deformable convolutional layer that reshapes its sampling grid according to a calibrated camera model. By adapting kernels rather than rectifying images, RectConv enables pre-trained models to operate directly on fisheye and other distorted imagery without fine-tuning. We validate on segmentation and detection benchmarks, showing substantial performance gains over naive rectification and patch-based methods, with only a modest computational overhead.

In Chapter 5 we investigate the fusion of heterogeneous sensors, specifically RGB and thermal cameras, through a transformer-based architecture augmented with ray based embeddings. The model encodes rays, and modality within rotary positional embeddings, then attends across all inputs to predict cross-modal novel views and depth maps. We detail a self-supervised training strategy where one modality's view is predicted from another, and present a comparison to state-of-the-art rgb-only alternatives and multi-modal simulation results demonstrating integration of heterogeneous sensors into a single geometrically consistent understanding.

The final Chapter 6 summarises the thesis's contributions, with how they relate to the broader goal of plug-and-play cameras, and the impacts of this work. We then identify promising directions for extending this research, including support for additional camera archetypes, online adaptation to changing environmental conditions and tighter integration of the proposed methods.

Chapter 2

Background

> *"Seeing comes before words. The child looks and recognises before it can speak."*
>
> — John Berger, *Ways of Seeing*

This chapter provides foundational knowledge to support the technical chapters that follow. It introduces the key concepts, models, and tools used throughout the thesis, ensuring clarity and a shared understanding of the terminology and algorithms. The goal is to establish a common baseline from which the proposed methods can be better appreciated, both in terms of their design rationale and practical relevance.

2.1 Cameras

Here we provide some background information on the cameras and process that are relevant for the work conducted in this thesis. This includes looking at different camera technologies and modalities as well as how calibration is conventionally performed.

2.1.1 Camera Technologies

Three distinct camera modalities are employed in this thesis, and their characteristics are described in the following sections.

Conventional Cameras. Throughout this thesis, we make reference to "conventional" cameras and imagery. By this, we refer specifically to imaging systems that operate within the visible spectrum (i.e., RGB cameras), capture data using standard optics, and can be reasonably approximated by a pinhole projection model. These cameras typically do not exhibit extreme distortion or wide fields of view, and their behaviour can be effectively modelled using a small number (2–4) of distortion parameters such as radial and tangential coefficients. Such systems are representative of the most common sensors used in consumer electronics, robotics, and industrial vision applications, and serve as the baseline modality for much of the work presented in this thesis.

Fisheye Cameras. Fisheye cameras deviate significantly from the assumptions underlying the pinhole model. Unlike conventional cameras that rely on rectilinear projection, fisheye cameras use highly non-linear projection models that deliberately introduce significant radial distortion to capture extremely wide fields of view, often exceeding 180 degrees. Our philosophy is that this distortion is not an artefact to be corrected, but rather a fundamental feature of the optical design that enables the camera to map a hemispherical scene onto a finite image plane. As such, the image formation process in fisheye cameras does not conform to a single consistent focal length or linear ray model, which makes standard calibration and interpretation methods inadequate.

To model fisheye cameras, alternative projection functions are employed, such as the equidistant, equisolid-angle, stereographic, or orthographic mapping [51]. Each of these defines a unique relationship between the angle of incidence of a ray and its projected distance from the image centre. While the pinhole model assumes a linear relationship between angular displacement and pixel displacement, fisheye models generally require non-linear functions that can only be approximated with higher-order distortion terms or entirely different formulations. As a result, the camera intrinsics for fisheye systems cannot be represented accurately using just a focal length and principal point with a few radial distortion coefficients. This poses challenges not only for geometric calibration but also for applying existing vision models that assume linear projection, necessitating dedicated architectures or adaptation layers to account for the complex imperfect mapping of a sphere to a plane.

Thermal Cameras. RGB cameras, based on silicon image sensors and standard optical lenses, are the most common type of vision sensor. They capture visible light across three colour channels and are used in the majority of consumer, industrial, and robotic imaging systems. However, many real-world environments and tasks demand information beyond the visible spectrum.

Thermal or long-wave infrared (LWIR) cameras operate in a different region of the electromagnetic spectrum and are sensitive to emitted thermal radiation rather than reflected light. These cameras offer the ability to perceive heat patterns, which is particularly useful in low-light conditions or for detecting living beings.

Despite serving similar roles imaging a scene, RGB and thermal cameras exhibit fundamentally different behaviours. The images they produce vary drastically in texture, contrast, and signal-to-noise characteristics. These differences make cross-modality tasks such as point correspondence or sensor fusion particularly challenging. Identifying matching features between an RGB image and its thermal counterpart generally requires sophisticated domain-specific approaches.

Unlike RGB cameras that rely on Bayer-pattern colour filters to produce colour images, thermal cameras detect absolute temperature differences in the LWIR range (typically 8–14 µm) [107]. Figure 2.1 shows where thermal cameras are sensitive to

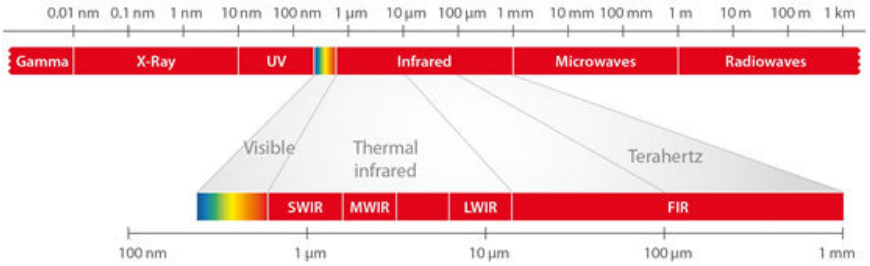

Figure 2.1 – Electromagnetic spectrum. RGB cameras are sensitive to the visible frequencies, while the thermal camera operate in the infrared frequencies. Typical thermal cameras are specifically LWIR cameras, which are sensitive to LWIR frequencies.[1]

the electromagnetic spectrum. These sensors do not require visible illumination and often function passively, but they also suffer from a different set of physical constraints. For instance, thermal optics are typically made from materials such as germanium or chalcogenide glass, which have very different refractive properties than visible-spectrum lenses. Furthermore, thermal imaging lacks sharp texture and edges, especially in scenes with low thermal gradients, resulting in low spatial frequency content. This makes conventional feature-based algorithms e.g., SIFT [78], ORB [92], or edge detection less effective when applied to thermal images directly.

A distinctive challenge in thermal imaging is the presence of spatially varying fixed-pattern noise, caused by the sensors being thermally sensitive. This manifests as a static horizontal and vertical lines superimposed on the thermal signal. To mitigate this, thermal cameras employ a process called non-uniformity correction (NUC), which calibrates out pixel-wise gain and offset differences by periodically capturing flat-field references often using an internal mechanical shutter that occludes the sensor. While effective, this correction is imperfect and drifts over time and with temperature, meaning residual artefacts often persist in thermal imagery. These artefacts are not only visually apparent but can confound downstream tasks such as feature matching, segmentation, or reconstruction, especially in learning-based pipelines that

[1] `https://www.viewsheen.com/blog/what-is-nir-swir-mwir-lwir-fir-spectral-range-/`

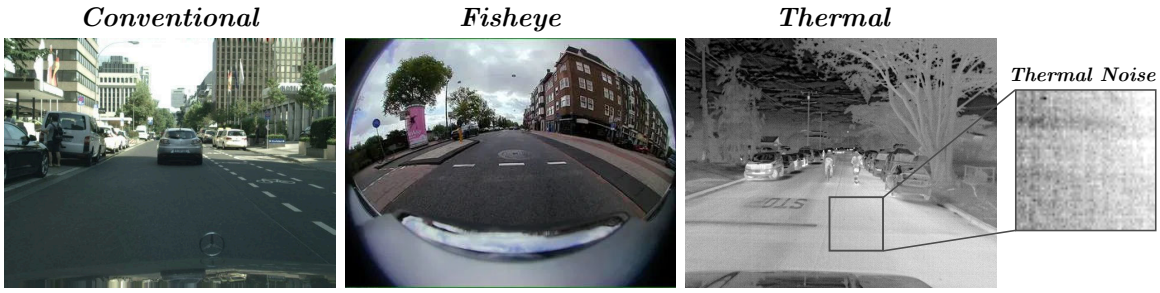

Figure 2.2 – Example images of conventional [20], fisheye [125] and thermal [33] camera types, each with their own characteristics. See thermal inset for example of the distinctive noise pattern that is common for thermal imagery.

assume clean and consistent input data. An example of a thermal image is shown in Figure 2.2

2.1.2 Calibration

Accurate calibration is critical for any vision system tasked with metric reasoning about the 3D world. Calibration refers to the process of determining the mapping from image pixels to rays in 3D space, and it underpins tasks such as reconstruction, localisation, and sensor fusion.

Camera Intrinsics. At the core of camera calibration lies the intrinsic model, which defines how 3D rays are projected onto a 2D image plane. This model is typically parameterised by a small set of variables, including focal length, principal point, and lens distortion coefficients. These parameters serve as simplified abstractions of the camera's physical properties, such as sensor dimensions, lens configuration, and optical alignment.

Various intrinsic models have been developed, ranging from the basic pinhole model to more advanced formulations that accommodate wide-angle lenses and significant non-linear distortions. The standard pinhole camera model assumes an ideal pinhole projection and is often augmented with radial and tangential distortion terms to handle mild lens imperfections. In contrast, the fisheye camera has alternative calibration methods, such as Kannala–Brandt [62], accounts for extreme wide-angle

distortion using higher-order polynomial functions of the incident ray angle. These fisheye models are particularly useful for omnidirectional systems and are commonly used in robotics and automotive applications.

The simplest and most widely used camera model is the pinhole camera model, which assumes an idealised projection through a single point onto an image plane. Under this model, a 3D point $X = (X, Y, Z)^\top$ in the camera coordinate frame projects to image coordinates (u, v) according to

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix}, \tag{2.1}$$

where $\mathbf{K}$ is the intrinsic calibration matrix

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.2}$$

Here (f_x, f_y) represent the focal length in pixel units and (c_x, c_y) denotes the principal point, corresponding to the projection of the optical centre onto the image plane.

Real lenses deviate from the ideal pinhole model due to optical distortion. The most common form is radial distortion, which causes image points to shift outward or inward relative to the image centre. Radial distortion is typically modelled as a polynomial function of the radial distance $r = \sqrt{x^2 + y^2}$ in the normalised image plane.

These distortion models form the basis of the widely used Brown–Conrady camera model [8], which is implemented in many calibration toolkits such as OpenCV. For cameras with relatively narrow fields of view, the combination of the pinhole projection and low-order distortion terms provides a sufficiently accurate approximation of the imaging process.

New sensor types with unconventional imaging geometries, such as omnidirectional and light-field cameras, often require the design of novel calibration models. For these systems, ongoing research aims to develop accurate and tractable parameterisation that can integrate effectively within contemporary computer vision and machine learning frameworks.

Wide-FOV Camera Models. Conventional perspective camera models assume that image formation follows the pinhole projection model. While this approximation is valid for cameras with moderate fields of view, it becomes inaccurate for wide-angle and fisheye lenses where strong radial distortion is present. In these systems the relationship between the angle of an incoming ray and its projected image location is no longer well approximated by the perspective projection equation.

Fisheye camera models therefore commonly represent projection as a function of the angle between the incoming ray and the optical axis. This representation is particularly important for methods that operate directly in distorted image space, where both the forward projection from 3D points to image coordinates and the inverse mapping from pixels to 3D rays must be available.

In this thesis these mappings are required to transform between image coordinates and rays in 3D space when adapting convolutional operations to non-perspective camera geometries. The explicit forward and inverse projection functions defined below are therefore presented in detail, as they are used directly in the formulation of the camera-aware convolution described in Section 4.3.1.

Let a 3D point in camera coordinates be denoted by $p = (X, Y, Z)$. The angle of incidence θ between the viewing ray and the optical axis can be written as

$$\theta = \arctan 2(\chi, Z), \tag{2.3}$$

where

$$\chi = \sqrt{X^2 + Y^2}. \tag{2.4}$$

Rather than projecting points according to the perspective relation $r = f\tan(\theta)$, fisheye models define a function that maps this angle to a radial image distance ρ from the optical axis. A commonly used formulation is the polynomial projection model employed by the WoodScape dataset [125]. In this model the radial distance is expressed as

$$\rho(\theta) = k_1\theta + k_2\theta^2 + k_3\theta^3 + k_4\theta^4, \tag{2.5}$$

where $k_1, \ldots, k_4$ are calibration coefficients.

Forward Projection. Given a 3D point $p = (X, Y, Z)$ in camera coordinates, the projection to distorted image coordinates is defined as

$$f_{2D}(p) = \begin{bmatrix} u' \\ v' \end{bmatrix} = \frac{\rho(\theta)}{\chi} \begin{bmatrix} X \\ Y \end{bmatrix}, \tag{2.6}$$

where $\rho(\theta)$ is computed using the polynomial model above. The coordinates (u', v') represent distorted lens-plane coordinates relative to the optical centre.

Back-Projection. The inverse mapping converts a distorted image coordinate into a 3D point or ray in camera space. Given a lens-plane coordinate (u', v'), the radial distance is

$$\rho = \sqrt{u'^2 + v'^2}. \tag{2.7}$$

The corresponding incidence angle θ is obtained by numerically inverting the polynomial relation $\rho(\theta)$. For a specified ray length n, the corresponding 3D point is then recovered as

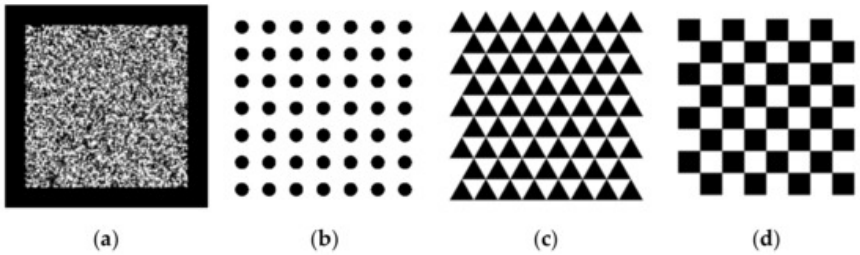

Figure 2.3 – Example of different calibration targets [61], using (a) speckle; (b) dot; (c) triangle; and (d) chessboard patterns.

$$\chi = n \sin \theta, \tag{2.8}$$

$$X = \frac{\chi}{\rho} u', \tag{2.9}$$

$$Y = \frac{\chi}{\rho} v', \tag{2.10}$$

$$Z = n \cos \theta. \tag{2.11}$$

This formulation defines the inverse mapping

$$p = f_{3D}(u', v', n), \tag{2.12}$$

which returns a 3D point along the viewing ray corresponding to the image coordinate.

Camera Extrinsics. Camera extrinsics describe the pose of the camera in a global coordinate system, capturing both its position and orientation. These parameters define the rigid transformation between the camera frame and the world frame and are essential for multi-view geometry, 3D reconstruction, and sensor fusion. Accurate extrinsic calibration ensures that observations from multiple viewpoints or sensors can be aligned within a consistent spatial context. In dynamic or modular systems, where camera positions may shift over time or vary between deployments, estimating or adapting extrinsics becomes a critical component of system performance.

Target Calibration. The most common calibration techniques involve imaging an object or target with known geometry or structure, typically a checkerboard or circle

grid (see Figure 2.3 for examples of different targets), then an optimisation is performed for the model parameters that best explain the observed projections. While effective, this process can be tedious and highly sensitive to setup conditions. The calibration process is also sensitive to target degradations; for instance, if the target were to warp, it would no longer be able to be used to perform accurate calibration. Calibration must also be repeated any time the sensor or optics are moved or modified, which adds challenges during deployment. Anyone with experience calibrating cameras will attest that it is often a time-consuming task. Achieving good results can require practice and careful attention to detail.

Self-Calibration. Self-calibration, sometimes called targetless calibration or auto-calibration, refers to methods that recover camera parameters without relying on a physical calibration target. Instead, these techniques infer intrinsics and/or extrinsics from structures in the captured data. Examples include structure-from-motion (e.g., COLMAP [93]) and recent learning-based approaches [77, 109] that estimate camera parameters from single images or videos. These methods are particularly appealing in scenarios where deploying calibration targets is impractical. Another school of thought is that we do not need to perform explicit calibration, instead utilising methods that implicitly estimate the camera parameters then directly output the desired downstream task [112, 69].

2.1.3 Ray Parameterisation

This thesis focuses on vision systems that capture images interpretable as sets of rays. A ray represents the path of light travelling through the scene, and defining how these rays are parameterised is essential for modelling, calibration, and rendering.

Two-Plane Parameterisation. One intuitive method is the two-plane parameterisation [43], where a ray is described by its intersections with two parallel planes, commonly referred to as the (uv, st) planes, see Figure 2.4 a). This formulation is well-suited for light-field cameras and is conceptually simple. However, it breaks down

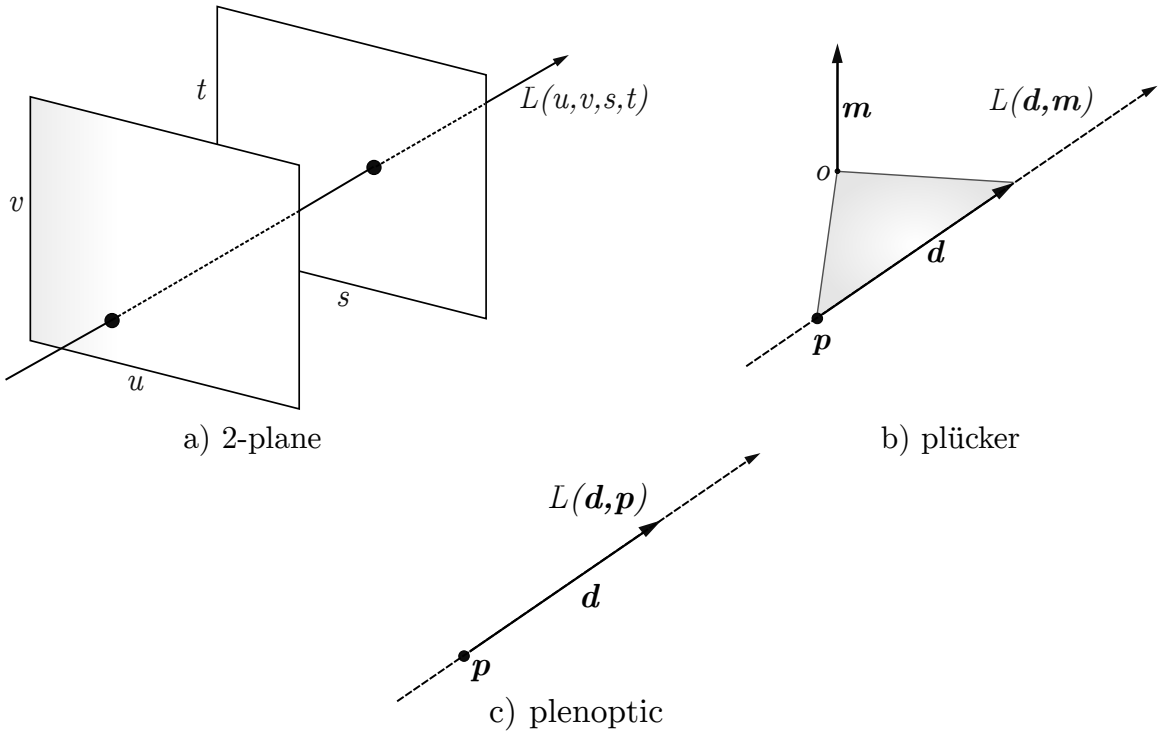

Figure 2.4 – Example of two possible ray representations. a) Shows the 2-plane representation where ray are parameterised as intersection points of the ray and two planes uv plane and st plane. b) Shows Plücker representation where the ray is parameterised as a ray direction ($\mathbf{b}$)and a moment ($\mathbf{m}$). Where $\mathbf{m}$ is the cross-product of the point and direction ($\mathbf{p} \times \mathbf{d}$). c) Shows the plenoptic representation which is parameterised as a direction d and a point alone the ray p.

for rays parallel to the planes, limiting its applicability in some domains. For that reason it is not used in this thesis.

Plücker Coordinates. Plücker coordinates [86] provide a more general and mathematically elegant representation. A ray in 3D space is defined by a direction vector $\mathbf{d}$ and a moment vector $\mathbf{m} = \mathbf{p} \times \mathbf{d}$, where $\mathbf{p}$ is any point along the ray. This formulation captures both the orientation and spatial embedding of a ray and is especially useful in geometric deep learning and neural rendering. A graphical representation is shown in Figure 2.4 b). Plücker coordinates require 6 numbers to express, they are an overparameterisation of the 4D light field. In this thesis, Plücker coordinates are the primary representation used for modelling and rendering rays. The main benefit is enabling representation of all angles without breaking down for certain rays, as with

the two-plane parameterisation. This parameterisation is used in Chapter 3 and in Chapter 5.

Plenoptic Coordinates. The plenoptic representation describes a ray using two components: a point along the ray $\mathbf{p}$ and its direction $\mathbf{d}$, see Figure 2.4 c). This is a conceptually straightforward and physically meaningful parameterisation. There are multiple ways to define plenoptic coordinates, but in the most basic form, they consist of three spatial coordinates and two angular coordinates, 5D. This representation is a subset of the more general plenoptic function, which can also include additional dimensions such as time and wavelength, offering a more comprehensive description of light. However, for most practical vision tasks, the 5D formulation is sufficient. In this thesis, a variation of the 5D plenoptic representation is used in Chapter 5: the point $\mathbf{p}$ encodes the camera's position in world coordinates (x, y, z) and a normalised direction vector encodes the direction of the ray.

2.1.4 Light Fields and the Plenoptic Function

The light field is a structured representation that captures the radiance along a collection of light rays in a scene. It enables us to encode how light travels through space, recording the colour or intensity associated with each ray. In practice, the 4D light field describes radiance as a function of two spatial and two angular dimensions, often based on a two-plane parameterisation.

$$L(u, v, s, t) \tag{2.13}$$

where (u, v) represent spatial coordinates on an image plane and (s, t) represent angular coordinates corresponding to the position of the ray on a second plane. Together these parameters uniquely define a ray in free space under the two-plane parameterisation. The function $L(u, v, s, t)$ therefore represents the radiance travelling along that ray.

This is an efficient way to represent the radiance of a scene. It is particularly well suited for applications such as view synthesis, depth estimation, and rendering, where the objective is to model how a scene appears from different viewpoints. The plenoptic function generalises this concept by describing radiance as a function of several physical variables.

$$P(x, y, z, \theta, \phi, \lambda, t) \tag{2.14}$$

where (x, y, z) denotes a point in 3D space, (θ, ϕ) specify the direction of the ray, λ represents wavelength, and t represents time. This formulation describes the radiance of every ray at every point in space, making it the most complete representation of the visual information in a scene.

While the light field provides a compact and practical representation, it has limitations. Since it stores a single radiance value per ray, it cannot represent multiple surfaces that intersected along a ray. One solution to this is the surface light field [118]. Another solution which we use in this thesis is the full plenoptic function, it allows for more comprehensive modelling of scenes where there are multiple surfaces that need to be represented by a single ray. It does this at the cost of higher dimensionality.

In this thesis, light fields are used as a supervisory signal in learning frameworks, predominantly in Chapter 3. However, we use the plenoptic function in Chapter 5 in order to explicitly predict depth. By representing scenes as continuous functions over ray space, we enable differentiable rendering and novel view synthesis. These capabilities allow self-supervised learning objectives, where a model is trained to predict unseen views or modalities from known inputs. This view-based supervision is employed to refine both radiometric and geometric consistency, and is a core component in the self-calibration and multi-modal fusion approaches presented in later chapters.

2.1.5 Heterogeneous Camera Systems

Many real-world perception systems rely on multiple sensors that observe the environment through different physical modalities. Such heterogeneous camera systems combine sensors with distinct imaging characteristics, such as RGB, depth, thermal infrared, or event cameras. Each modality captures different aspects of the scene, and by integrating these observations a system can often achieve greater robustness and reliability than would be possible using a single sensor alone.

A common example is the combination of RGB and thermal cameras. RGB sensors measure reflected visible light and provide rich colour and texture information, which is useful for many vision tasks under well-lit conditions. Thermal cameras, in contrast, measure emitted infrared radiation and therefore capture temperature differences in a scene. As a result, thermal imagery is largely invariant to visible lighting conditions and can remain informative in environments where RGB cameras struggle, such as at night, in low illumination, or in the presence of smoke or fog. By combining information from both modalities, heterogeneous sensing systems can leverage the complementary strengths of each sensor, improving perception performance across a wider range of operating conditions.

From an algorithmic perspective, integrating heterogeneous sensor data requires some form of sensor fusion. Traditional approaches often rely on hand-crafted fusion strategies, where features extracted independently from each modality are combined using engineered rules or geometric alignment procedures. For example, feature-level fusion may concatenate descriptors from multiple sensors, while decision-level fusion may combine predictions from separate models. While these approaches can be effective, they often depend heavily on carefully designed feature representations and may struggle to fully exploit complex relationships between modalities.

More recently, data-driven methods have been developed to learn cross-modal representations directly from data. In these approaches, neural networks are used to jointly process multiple modalities and learn a shared representation that captures complementary information from each sensor. Such learned fusion methods can auto-

matically discover relationships between modalities that would be difficult to model manually [105]. This paradigm has become increasingly common in multimodal perception systems, including applications in autonomous driving, robotics, and remote sensing.

In this thesis, heterogeneous sensing is explored in the context of neural rendering and scene understanding. In particular, Chapter 5 investigates how information from multiple imaging modalities can be integrated within a learned framework to improve scene representation and rendering performance.

2.2 Neural Rendering

Rendering refers to the process of generating images from a scene representation. Traditional graphics pipelines rely on explicit geometric models and hand-crafted shading techniques. In contrast, neural rendering replaces parts of this pipeline with learned components implemented using neural networks. These models can represent complex scene geometry, appearance, and sensor effects directly from data, enabling flexible rendering pipelines that are difficult to construct using traditional analytic models.

This thesis incorporates several common neural network architectures to support the rendering process, including multi-layer perceptrons (MLPs) [91], CNNs [68], and vision transformers (ViTs) [28]. Each architecture offers different inductive biases and computational characteristics that make them suitable for different components of the rendering pipeline. The following sections briefly introduce these architectures and describe their role within the methods proposed in this thesis.

Multilayer Perceptrons. MLPs are one of the most fundamental neural network architectures. An MLP consists of a sequence of fully connected layers, where each layer applies a linear transformation followed by a non-linear activation function.

MLPs are widely used in neural rendering because they can represent continuous functions over spatial coordinates. In many neural scene representations, such as

neural radiance fields (NeRF) and implicit light field models, an MLP maps spatial coordinates or rays to quantities such as density, colour, or radiance. Within this thesis, MLPs are primarily used as implicit function approximators that map encoded spatial inputs to scene properties. They form a core component of the rendering models described in Chapter 3 and Chapter 5.

Convolutional Neural Networks. CNNs are neural networks specifically designed to process grid-structured data such as images. Instead of fully connected layers, CNNs use convolutional filters that operate locally across the input. This structure enables CNNs to efficiently capture spatial patterns while sharing parameters across the image domain.

Because convolutional filters operate locally and share parameters across spatial locations, CNNs are particularly effective at extracting hierarchical image features while remaining computationally efficient. In this thesis, CNNs are used to process image observations and extract spatial features prior to rendering or geometric inference. They are used extensively in the models presented in Chapter 3 and form the basis of the rectified convolution architecture introduced in Chapter 4.

Vision Transformers. Vision Transformers (ViTs) extend the transformer architecture, originally developed for natural language processing, to visual data. Rather than processing images with convolutional filters, ViTs divide an image into a sequence of patches which are embedded into a high-dimensional feature space. These embeddings are then processed using layers of self-attention.

Self-attention allows each element in the sequence to attend to every other element, enabling the model to capture long-range dependencies across the entire image.

Transformers provide a powerful mechanism for modelling global relationships between features, which can be beneficial in multi-view or multi-modal settings where information must be aggregated across different sensors or viewpoints. Internally transformers utilise MLPs as a fundamental building block. In this thesis, a ViT-based architecture is used in Chapter 5 to fuse information from multiple sensing

modalities, enabling the network to reason about spatial relationships across heterogeneous inputs.

2.2.1 Representations

Explicit. Explicit methods use structured representations such as voxel grids, triangle meshes, or point-based primitives like 3D Gaussians. These models offer direct control over scene geometry but often require substantial memory or pre-processing.

Implicit. Implicit approaches model scene properties, such as colour or density, as continuous functions of spatial coordinates. These properties can be represented in a number of ways. One popular method is by fitting neural networks.

This thesis focuses primarily on implicit representations encoded into some latent space via neural networks. Implicit models provide a flexible and memory-efficient way to encode complex scene content and are well suited to tasks involving novel view synthesis, depth prediction, and self-supervised learning from image data.

2.2.2 Approaches

We categorise neural rendering broadly into two groups: regression-based methods and feedforward methods.

Regression Methods. NeRF [80] introduced a major advancement in implicit scene representation by using an MLP to model volumetric radiance fields. The network learns to map 3D spatial locations and viewing directions to RGB colour and density values, enabling photorealistic novel-view synthesis. NeRF have since inspired a large body of research into differentiable rendering and neural scene representations. Although NeRFs are not used directly in this thesis, they are referenced and serve as comparative alternatives.

Recent developments in explicit representations, such as Gaussian Splatting [63], provide high-fidelity rendering with real-time performance. Although this method is not

based on neural networks, it offers a compelling alternative by using explicit, compact point-based structures for scene representation. These Gaussian based approaches are regressed in a similar fashion to NeRFs.

Feedforward Methods.

Neural rendering approaches can also be distinguished by how the scene representation is estimated. Many early methods formulate reconstruction as a regression problem that is solved independently for each scene. For example, models such as NeRF learn the parameters of a scene representation by directly optimising them against a set of input images. This process requires iterative gradient-based optimisation for every new scene and can therefore be computationally expensive.

Feedforward methods instead learn to predict a scene representation directly from observations using a neural network trained across many scenes. At inference time, the representation can be produced in a single forward pass, removing the need for per-scene optimisation. This substantially reduces reconstruction time and enables applications that require rapid inference, though it typically requires larger and more diverse training datasets to achieve strong generalisation.

This distinction concerns how the representation is estimated rather than how images are rendered. Both optimisation-based and feedforward approaches commonly rely on ray-based rendering once a scene representation has been obtained.

Feedforward prediction can be applied to both explicit and implicit scene representations. Recent work based on Gaussian splatting, such as NoPosPlat [121] and PixelSplat [11], learns to predict sets of Gaussian primitives directly from images, producing explicit scene representations in a feedforward manner.

Feedforward implicit approaches can also be applied to light field representations. LFNs [95] represent a scene by directly learning a function that maps an oriented light ray to the radiance observed along that ray. Rather than modelling volumetric density and performing numerical integration along the ray as in NeRF, an LFN predicts the radiance of a ray in a single network evaluation.

Formally, an LFN parameterises the 4D space of rays $\mathcal{L}$ using a neural network

$$\Phi_\phi : \mathcal{L} \to \mathbb{R}^3, \qquad c = \Phi_\phi(r), \tag{2.15}$$

where r denotes an oriented camera ray and c is the RGB radiance observed along that ray.

To represent rays in a continuous 360-degree domain, LFNs use the 6D Plücker coordinate parameterisation of a ray

$$r = (d, m), \qquad m = p \times d, \tag{2.16}$$

where $p \in \mathbb{R}^3$ is a point on the ray, $d \in S^2$ is the unit direction vector, and m is the moment vector defined by the cross product. Although Plücker coordinates lie in $\mathbb{R}^6$, valid rays occupy a 4D manifold within this space, allowing the representation of all oriented rays in an unbounded scene.

Given camera intrinsics K and extrinsics E, a ray $r_{u,v}$ corresponding to a pixel coordinate (u, v) can be constructed and evaluated by the network to obtain its colour

$$c_{u,v} = \Phi_\phi(r_{u,v}). \tag{2.17}$$

Because the colour of each ray is predicted directly, rendering requires only a single network evaluation per ray. This is in contrast to volumetric neural renderers such as NeRF, which typically require tens or hundreds of evaluations along each ray.

This approach was made to be feedforward by applying a hypernetwork which learns to encode an image and output the weights for a light field rendering network that is able to render the scene of the input image in a single forward pass.

LFNs are used in Chapter 3, and the conceptual foundations also inform the design of the rendering framework in Chapter 5. LVSM [60], which we compare against in Chapter 5, is another example of a feedforward implicit rendering model, this time based on vision transformers.

Overall, this thesis focuses on feedforward implicit neural rendering as a practical alternative to per-scene optimisation approaches, enabling scalable and generalisable vision systems.

2.2.3 Positional Encoding and Embedding

Injecting raw position information into neural networks is challenging due to positions close to each to other looking very similar to the network despite potentially having vastly different outputs. To allow neural networks to represent fine spatial details, positional encoding techniques inject high-frequency information into coordinate inputs. A common method is the sinusoidal encoding used in both NeRF and transformers.

The frequencies used in this encoding vary across dimensions. A common formulation, used in transformers [106], defines the encoding as:

$$\begin{aligned} \phi_{2i}(p) &= \sin\left(\frac{p}{b^{2i/d}}\right), \\ \phi_{2i+1}(p) &= \cos\left(\frac{p}{b^{2i/d}}\right). \end{aligned} \tag{2.18}$$

Here p denotes the position being encoded, d is the dimensionality of the embedding, and b controls the frequency scaling. The index i ranges from 0 to $\frac{d}{2} - 1$, producing paired sine and cosine components that together form a d-dimensional positional encoding. By mapping a scalar position into a higher-dimensional space of sinusoidal functions with increasing frequencies, the network is able to represent both low- and high-frequency spatial variations.

This sinusoidal encoding is used extensively in Chapter 3 and Chapter 5. Figure 2.5 shows an illustrative example of this encoding.

Positional embedding is related to but slightly different from positional encoding, and the terms are often conflated in the literature. Positional encoding typically refers to a deterministic mapping that transforms a position into a representation

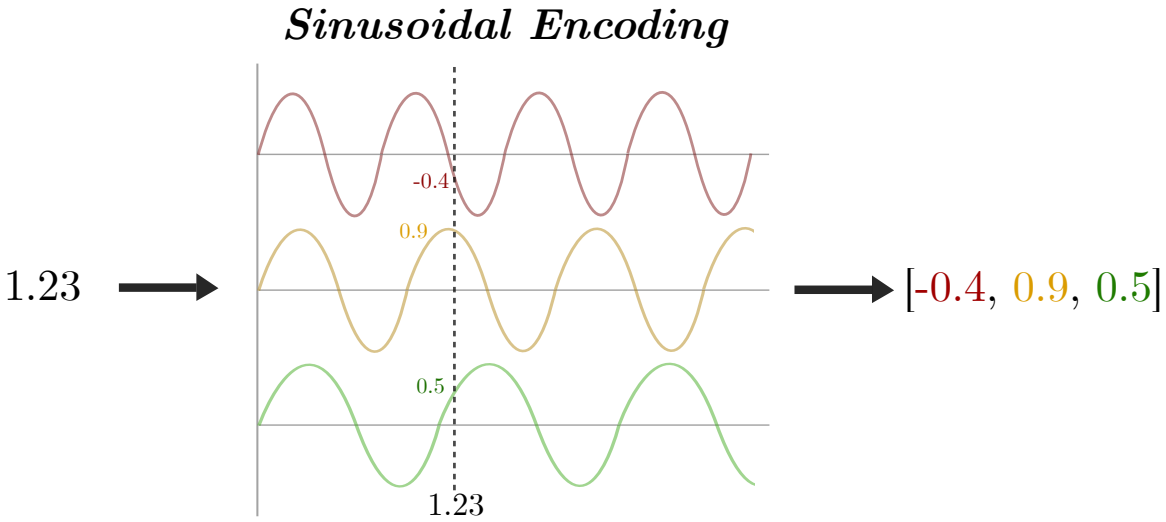

Figure 2.5 – Illustration of a sinusoidal encoding typically used of positional encoding in many neural rending networks.

that can be more easily utilised by a model. These encodings are usually handcrafted mathematical functions and may take several forms, including sinusoidal encodings or spherical harmonic representations [84]. Such encodings are widely used beyond neural networks whenever structured positional information must be represented in a higher-dimensional space.

In contrast, positional embeddings refer to a learned representation in which each position is assigned a vector within an embedding space of fixed dimensionality. These embeddings can either be generated from deterministic encodings, such as the sinusoidal representation described above, or learned directly from the training data such that every possible position has an associated embedding vector. In practice, the terminology varies across the literature. For example, the influential transformer architecture refers to sinusoidal mappings as positional encodings [106], while later works often use the term positional embedding to describe learned position representations [113].

A further distinction can be made between *absolute* and *relative* positional embeddings. Absolute positional embeddings encode the position of each element with respect to a fixed global coordinate system. While effective in many applications, this approach can limit generalisation because the model must learn relationships

between specific absolute positions. Relative positional embeddings instead encode the positional relationship between elements, such as the relative offset between two tokens or spatial locations. This formulation allows the model to focus on the geometric or sequential relationships between inputs rather than their absolute locations, which often improves generalisation and robustness to shifts or changes in scale.

One example of a relative positional embedding method is rotary positional encoding (RoPE) [97], which was developed specifically for transformer architectures. RoPE incorporates positional information by rotating pairs of embedding dimensions using a position-dependent rotation matrix. If $x \in \mathbb{R}^d$ denotes an embedding vector, the rotary positional embedding applies a rotation to each two-dimensional subspace of the vector according to the position index. This rotation implicitly encodes relative positional relationships, as the inner product between two rotated embeddings becomes dependent on the difference between their positions rather than their absolute values.

This embedding process is heavily used in Chapter 5. Further details on this method and the modifications made are discussed there.

2.3 Applications

In this section, we describe the downstream tasks used throughout this thesis to demonstrate the effectiveness of the proposed methods. These tasks: odometry, depth estimation, novel view synthesis, and image segmentation, are representative of fundamental challenges in scene understanding. Each task requires different combinations of geometric awareness, photometric consistency, and semantic interpretation, making them ideal for evaluating the generalisability and adaptability of vision systems across diverse camera models and modalities.

2.3.1 Odometry

Odometry is the task of estimating the motion of a camera (or robot) over time. It is fundamental to autonomous navigation and serves as a testbed for end-to-end systems that require calibration, tracking, and scene understanding. This thesis uses odometry to demonstrate the utility of self-calibrated vision systems. Odometry is used in Chapter 3.

2.3.2 Depth Estimation

Depth estimation involves predicting the distance to objects in the scene from single or multiple images. It can be achieved through classical stereo matching or learnt monocular depth prediction. The work in this thesis builds on learnt approaches, often using depth as a latent or auxiliary target in self-supervised tasks. Chapter 5 uses depth estimation.

2.3.3 Novel View Synthesis

Novel view synthesis (NVS) aims to generate unseen views of a scene from a limited set of observations. It requires an accurate understanding of scene geometry and texture, making it a rigorous test of the underlying scene representation. Many of the models presented in this thesis use NVS as both a learning signal and a benchmark for generalisation. This is used in both Chapter 3 and Chapter 5

2.3.4 Image Segmentation

Image segmentation refers to the task of assigning a semantic label to each pixel in an image, effectively partitioning the image into meaningful regions. It plays a central role in perception systems that require high-level understanding of scene content. Accurate segmentation depends not only on texture and appearance cues but also on consistent geometry and viewpoint. In this thesis, segmentation is used to evaluate

the ability of pretrained models, adapted via camera-aware layers, to generalise across different sensor geometries without requiring retraining or manual annotation. This is used in Chapter 4.

2.4 Evaluation Metrics

For completeness we provide brief explanation and formulas for the metrics used throughout this thesis.

Mean Error. Mean squared error (MSE) measures the average squared difference between two signals. Using pixel values as an example, when finding the error between a predicted image $\hat{I}$ and reference image I. It penalises larger errors more heavily and is sensitive to absolute pixel differences,

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{I}_i - I_i\right|^2, \tag{2.19}$$

here, N is the total number of pixels, $\hat{I}_i$ is the predicted colour value at pixel i, and I_i is the ground-truth colour value at the same pixel. Root mean squared error (RMSE) applies an additional square root providing a linear scaling to the error,

$$\text{RMSE} = \sqrt{\text{MSE}}. \tag{2.20}$$

Peak signal-to-noise ratio (PSNR) [52] is a commonly used metric for evaluating the quality of image reconstructions. It expresses the ratio between the maximum possible pixel intensity and the power of the distortion (i.e., the error). PSNR is particularly useful because it provides an interpretable scalar value that increases with better image quality, higher values indicate that the reconstructed image is closer to the ground truth. Because it is conventionally reported on a logarithmic scale, PSNR provides a more compressed and human-perceivable measure of reconstruction quality than raw pixel-wise errors,

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{L^2}{\text{MSE}}\right), \tag{2.21}$$

L is the maximum possible value (e.g., 255 for 8-bit images).

SSIM. Structural similarity index measure (SSIM) [114] is a perceptual metric that quantifies image similarity by comparing luminance, contrast, and structure between a predicted image x and a reference image y.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{2.22}$$

here, μ_x and μ_y are the local means, σ_x^2 and σ_y^2 are the local variances, and σ_{xy} is the covariance between x and y. Constants C_1 and C_2 are included to stabilise the division when the denominator is small. SSIM values range from -1 to 1, with 1 indicating perfect structural similarity.

LPIPS. Learned perceptual image patch similarity (LPIPS) [128] compares perceptual similarity between images using deep feature representations. Let $\phi_l(I)$ be the l-th layer features of image I from a pretrained network:

$$\text{LPIPS}(\hat{I}, I) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot \left(\phi_l(\hat{I})_{h,w} - \phi_l(I)_{h,w}\right)\right\|_2^2 \tag{2.23}$$

where w_l are learnt weights, and $\odot$ denotes element-wise multiplication.

AbsRel. Absolute relative error (AbsRel) [12] measures the error scaled by the true value. This is useful for depths metrics where larger depth values inherently have more error,

$$\text{AbsRel} = \frac{1}{N}\sum_{i=1}^{N} \frac{\left|d_i - \hat{d}_i\right|}{d_i}. \tag{2.24}$$

Threshold Accuracy. This measures the percentage of measurements which deviate less than 25% of the true value. This metric is used for depth,

$$\delta_1 = \frac{1}{N}\sum_{i=1}^{N} max\left(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}\right) < 1.25. \tag{2.25}$$

MIoU. Intersection over union (IoU) [55] is used as a metric for semantic segmentation for a given class. Mean intersection over union (MIoU) applies the mean over multiple classes. For a class c, let TP_c, FP_c, and FN_c denote true positives, false positives, and false negatives respectively,

$$\text{IoU}_c = \frac{TP_c}{TP_c + FP_c + FN_c}, \tag{2.26}$$

$$\text{MIoU} = \frac{1}{C}\sum_{c=1}^{C} \text{IoU}_c. \tag{2.27}$$

Precision, Recall, and F1 Score. Precision is a measure of the accuracy of the positives, while Recall is a measure of a model's ability to identify all positive instances. F1 Score provides a balanced measure of both metrics [17]. Given TP, FP, and FN,

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{2.28}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{2.29}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.30}$$

Chapter 3

Semi-Supervised Learning for Self-Calibration and Odometry

"Chaos is merely order waiting to be deciphered."

— José Saramago

Now with a common understanding and language of the concepts used throughout this thesis we move on from the background chapter to present our first of three technical chapters. The goal of the research discussed in this chapter is to alleviate the barriers to deployment associated specifically with calibration. We introduce NOCaL, a framework designed to automate integration of a camera into a robotic system by leveraging prediction as a feedback signal. It achieves this by leveraging an understanding of ray geometry.

Parts of this work are published as [40] and the code and additional visualisation are available at: `https://roboticimaging.org/Projects/NOCaL/`.

3.1 Overview

As discussed in Chapter 1, vision is a critical sense in many application ranging from robotics to autonomous driving to medical screening. With the increasing reliance on computer vision and its broadening applications we can no longer rely on a one size fits all approach to camera system design. Whilst novel cameras are being developed to address shortcomings in existing modalities, this raises a key problem in deploying these sensors quickly and on new platforms: calibration and low-level interpretation.

Calibrating and interpreting new imaging devices is skilled and time-consuming work. Emerging devices like event cameras and light field cameras have taken years and even decades to adapt in robotics. Solutions generally involve the use of bespoke models, calibration procedures, and low-level interpretation and sensor fusion algorithms. Additionally, static camera characteristics are generally assumed, with device changes due to vibration, temperature, replacement or upgrading requiring re-calibration as these changes can affect performance [96, 32]. This makes both integrating new cameras and managing fleets of robots onerous and complex.

In this chapter, we propose a framework to automatically interpret previously unseen cameras by jointly learning to estimate novel views, odometry, and camera parameters – see Figure 3.1. Our framework utilises advancements in neural rendering to provide self-supervision, leveraging the large amount of imagery available from a newly introduced uncalibrated camera. To benefit from the availability of unlabelled training data from existing cameras, we employ a hypernetwork that learns to construct light field renderers, so that the hypernetwork can be trained on multiple scenes. This would not be possible with a fixed rendering network as this could only be trained on a single scene. Finally, to ground our odometry estimates in metric space, we employ a small labelled training set that is easy to collect where a complementary source of odometry is available.

Through unsupervised learning of camera parameters and odometry, our approach benefits from the large amount of unlabelled data available from existing cameras as well as a newly introduced camera. By introducing a small labelled training set we

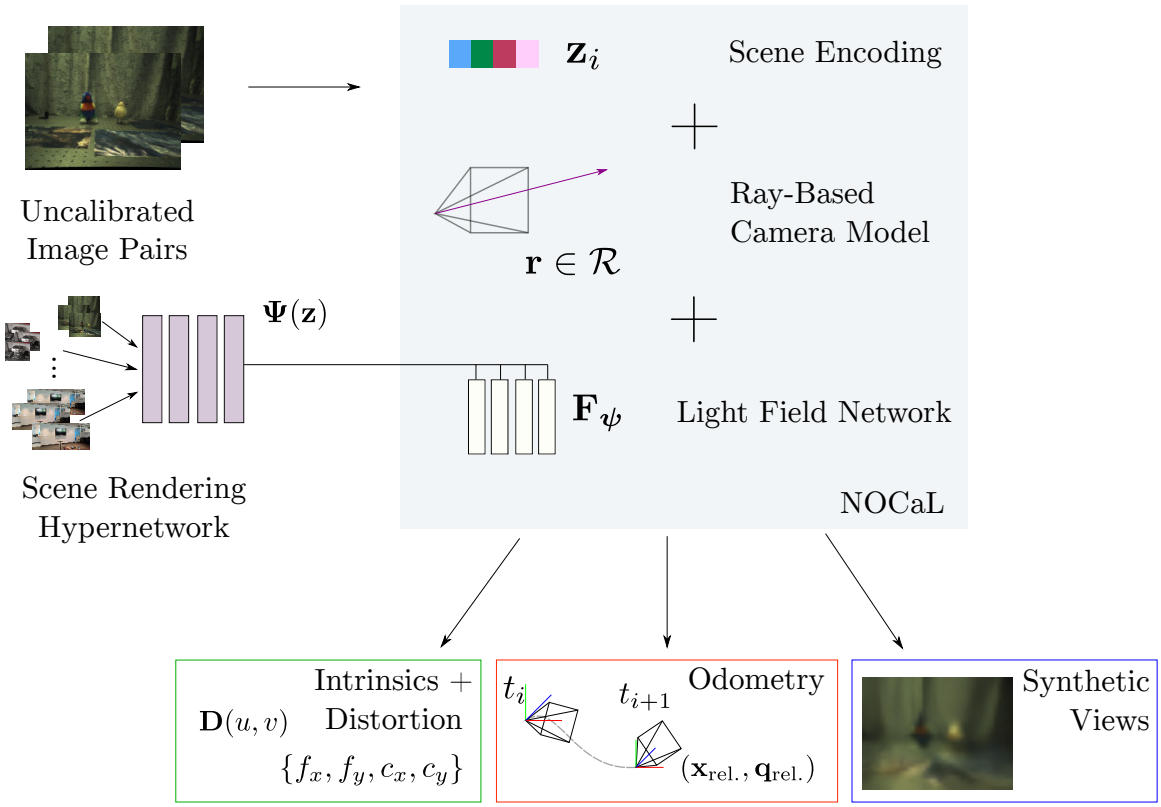

Figure 3.1 – NOCaL learns odometry and novel view synthesis from previously unseen cameras by leveraging a rendering hypernetwork pre-trained on existing cameras. Uncalibrated input image pairs are encoded in a latent vector $\mathbf{z}_i$ that drives the hypernetwork $\mathbf{\Psi}(\mathbf{z})$ to generate a light field rendering network $\mathbf{F}_\psi$. A differentiable ray-based camera model drives the renderer, enabling estimation of camera parameters, relative pose, and novel views. This work is a key step towards automatically interpreting more general camera geometries and emerging camera technologies.

constrain the solution to a metric space with known scale, while there is always some ambiguity with monocular scale estimation, using this approach allows the network to use semantic information to get reliable scale when working in a similar environment to that of the training data [59].

To demonstrate our approach, we employ cameras which are well described by a pinhole projection model with a freeform ray-based distortion model. This allows use of a broad range of monocular cameras without calibration. In future, we envision further relaxing this model to an entirely freeform ray-based model, leading to a

broader range of cameras including stereo, multi-aperture and light field, and fisheye, that are all well described by ray-based geometry.

We validate our approach on both captured and rendered images of indoor scenes, using cameras with different focal lengths and distortion parameters. We demonstrate our system accurately estimating camera intrinsics, distortion models, and relative pose, i.e. odometry.

To position our work, we compare against a fully supervised approach and an unsupervised approach that requires calibration. Perhaps surprisingly, our semi-supervised but uncalibrated approach outperforms both fully supervised and unsupervised approaches in accuracy of odometry, demonstrating the strength of combining small labelled datasets with readily available unlabelled data. We also include an ablation study that establishes the importance of the distortion model when using cameras that deviate substantially from an ideal pinhole model, comparing against variants of our method that lack a distortion model and that estimate no camera parameters at all.

We anticipate this work to find broad applicability where recalibration is difficult and camera parameters can change, either due to optical shifts or replacement of hardware. Deployed systems on planetary missions, in harsh environments, and in domestic applications like robotic vacuums for example are typically difficult to recalibrate. Changes to on-board calibration can occur due to vibration and thermal effects, and replacing or upgrading cameras can be an expensive proposition especially where new camera models and/or recalibration are required. Our framework requires no prior knowledge of hardware or camera parameters, allowing such robots to perform accurate camera pose estimation without need for recalibration or manual intervention.

Limitations: Whilst our network is designed to work with a family of cameras, cameras not well-described by a pinhole projection and freeform distortion profile are unlikely to be well supported by our camera model. Another limitation is that our method requires a rough initial guess of the camera intrinsics, which potentially requires some prior knowledge of the camera. In Chapter 4 we address challenges associated with

non pinhole cameras, specifically fisheye cameras. Pose estimation requires substantial overlap between the input images, and so the approach also breaks down for fast motion and low-overlap input pairs.

Since the publication of this work substantial progress has been made in the use of large transformer networks to solve the goals outlined in this work [112]. These large transformers show very promising resutls in the joint optimisation of the 3D scene, camera poses and intrinsics. These large networks are able to benefit from the large pool of training data to understand geometric priors. This is also a key benefit of the hypernetwork structure from our work. These transformer architectures have demonstrated an impressive ability to scale in size compared to the network architecture used in our work. We adopt these transformer architecture in a later contribution.

3.2 Literature Review

State-of-the-art approaches to monocular visual odometry jointly learn scene depth and odometry of images using unsupervised learning [129, 46, 124, 126, 36]. This generally uses a warp function to predict images in a sequence based on depth and estimated pose. This warping usually requires accurately calibrated images which are often not available on robotic platforms that are deployed in harsh environments. More recent work by Fang et al. [32] and Gordon et al. [37] have also been able to jointly learn the camera model, which alleviates one of the main difficulties with this approach. Ultimately these approaches still use a warp function which will limit the types of camera geometries that can be learnt using this method. Warp based approaches also have no ability to deal with view dependent objects, such as non-Lambertian surfaces. This is something that ray based methods, such as the one presented in this work, can accommodate.

Digumarti et al. [27] demonstrated the performance of warping using a novel 4D warp function by extending it to a new family of cameras: sparse light-field cameras. In using a warp function, these studies are limited to scenes with simple well-explained

phenomena, for example these methods cannot handle view dependent phenomena (reflection, refraction).

Recent studies on neural novel view synthesis [80] have demonstrated state of the art results, with applications to many computer vision applications. While this has been adapted to robotics in [1, 100, 54, 123], the fundamental limitation of such approaches exists in being only able to represent a specific scene, and within the spatial region captured by the input data. The computation required to ray-march is expensive, but provides a dense and continuous scene representation. Instant-NGP [85] addresses this limitation to an extent by employing a multi-resolution hash encoding, which drastically speeds up training and inference, but leads to more sparse geometry.

Light field network (LFN) [95] performs a single query of the network unlike prior works, enabling a reduction in training and inference time by several orders of magnitude compared to NeRF [80]. Recent studies [3, 72, 101, 110], leverage LFNs to produce comparable results to that of NeRFs, with tradeoffs between visual fidelity and speed.

Published after this work, an alternative representation to the LFN is the 3D Gaussian Splat. This representation has recently been successfully used to perform the same joint optimisation goal performed in this work allowing for self calibration of wide field of view cameras [25]. This Gaussian Splating representation differs from the LFN in that it is an explicit representation compared to the implicit representation of the LFN, which has some advantages in the interpretability of the model.

Back propagation of gradients through a neural field MLP provides networks an ability to refine parameters such as pose. While bundle-adjusting radiance fields [73] refines poses during the formation of a radiance field, using a NeRF as a supervisory signal for absolute pose regression [16] shows advantages in accuracy around convex and extended scenes. NeRFs do not perform well on few-shot datasets and require dense image coverage of a scene to create high fidelity results, however once generated it is possible to perform accurate pose regression on a minimal dataset [82]. Of key note, the refinement or determination of pose can be performed in addition to estimation

of other parameters within the neural field, such as shape, reflectance functions or illumination [7].

Joint learning of camera intrinsics and neural fields show improved extrinsics estimation. Wang et al. [115] demonstrate an ability to jointly learn focal length and extrinsics, achieving similar results to traditional methods like COLMAP [93].

Jeong et al. [58] jointly learn a complex non-linear distortion camera model with the neural field in several stages, allowing the framework to learn a simple pinhole model followed by complex components including non-linear distortion parameters. In essence, this curriculum learning approach enables the network to obtain correct scene geometry without warping the scene to agree with camera distortion.

The *Raxel* model [41] represents each photosensitive element as a virtual *ray pixel* that samples a bundle of rays along a principal direction, enabling cameras to be described entirely in terms of ray geometry without assuming perspective projection. Discrete raxels specify positions and directions for each pixel, while continuous models define a ray surface over the image plane, with caustics used to represent these surfaces for complex optics. Non-geometric properties such as point spread, radiometric response, and lens fall-off can also be incorporated, resulting in a complete imaging model that generalizes conventional cameras. While not directly used in this work, an extension of this formulation could be integrated into future ray-based neural representations.

Hypernetworks [45] allow one network to produce weights for other networks that can perform additional tasks. The framework presented by von Oswald et al. [108] has the ability to retain a vast amount of memory for multitask learning using a hypernetwork. This benefits the individual networks and allows for reuse, owing to the commonalities between the learning tasks.

Sitzmann et al. [95] also employs hypernetworks to leverage the ability to learn latent-based embeddings to produce a rendering network that can render the specific scene represented by latent embedding. Unlike prior works which focus on single scene, this network has the ability to render multiple scenes based on a single image.

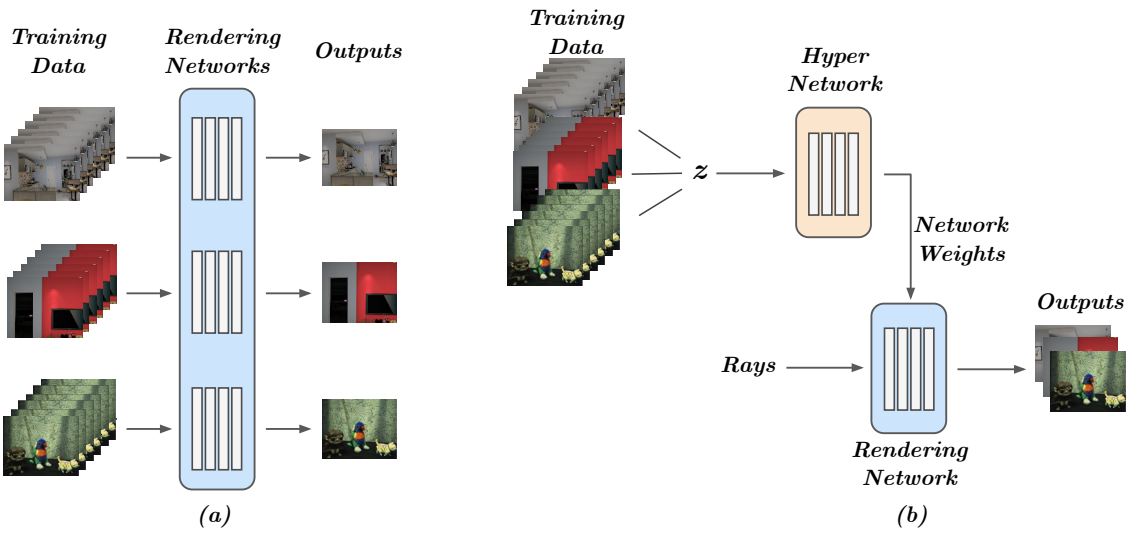

Figure 3.2 – Illustration of different neural rendering networks a) A standard non hyper network approach for neural rendering, each scene and network are separate. b) A hyper network approach means the network can learn from multiple datasets. A latent vector z is derived from input data.

Figure 3.2 illustrates the difference with the hyper network approach; it allows for one network to learn to produce a rendering network from a range of scenes, while the non hypernetwork approach has to learn from scratch for each new scene.

3.3 Method

3.3.1 Network Architecture

The two main learnable parts of the proposed network are the encoding network and the hypernetwork. These two parts work together to be able to jointly learn pose and scene geometry from input images. Details for implementation are discussed in Section 3.4.2.

Encoding Network

The camera interpreter portion of our pipeline serves as an encoding network, which converts a pair of input frames from a specific camera to a pose and a latent rep-

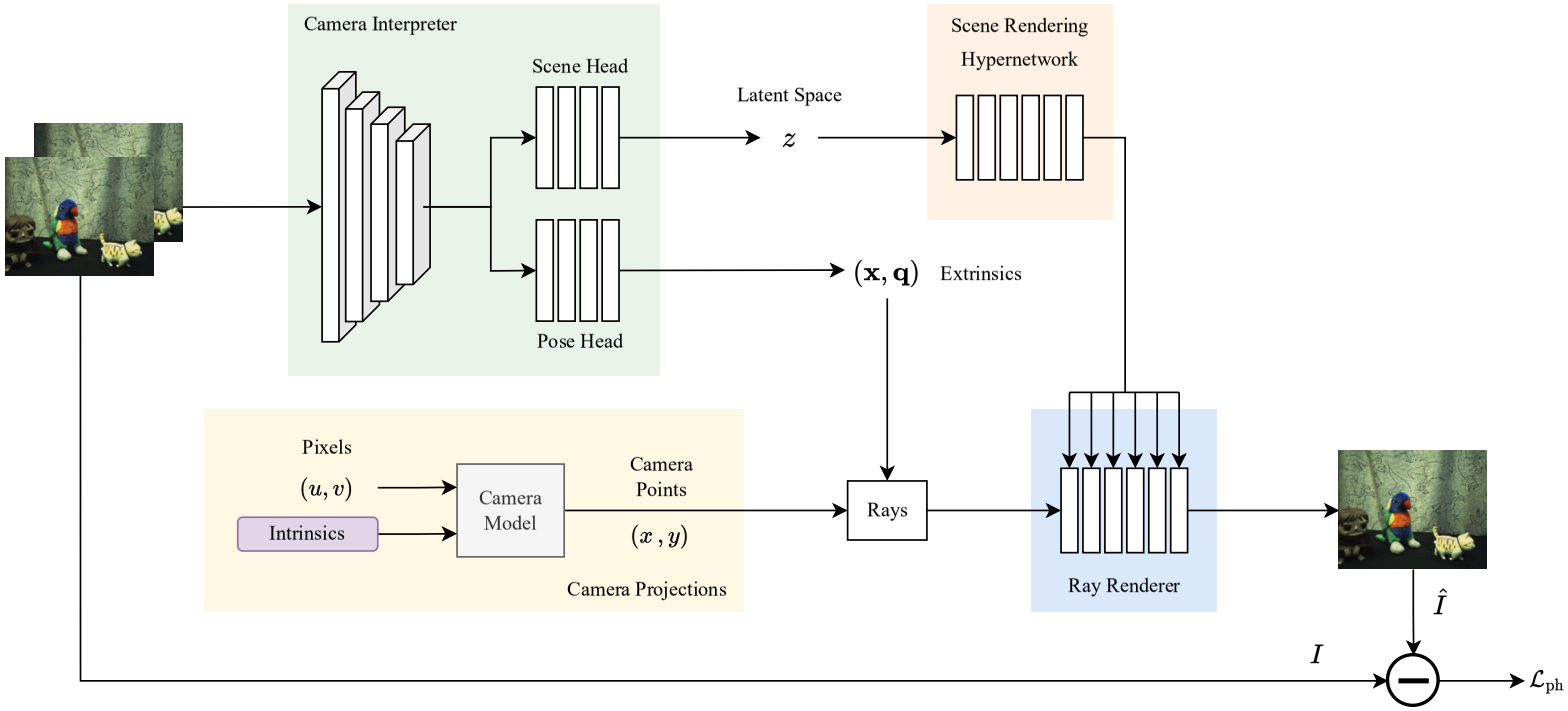

Figure 3.3 – The proposed NOCaL network architecture is made up of 4 separate sub-modules. The camera interpreter (green) encodes input frames into a scene-description latent space $\mathbf{z}$. The scene rendering hypernetwork (orange) uses the latent space to produce weights for the light field network (blue) which renders the scene captured by the input images. The camera model (yellow) has learnable parameters to be able to estimate the camera intrinsics used in capturing the input frames, and generates rays used by the light field network.

resentation of the scene. The encoder is implemented as a CNN with two separate heads.

The pose head predicts a translation vector $\mathbf{t} \in \mathbb{R}^3$ and a 6D rotation representation $\mathbf{r} \in \mathbb{R}^6$, following the continuous parameterisation of [131]. The 6D rotation vector is interpreted as two unconstrained 3D vectors $\mathbf{a}_1$ and $\mathbf{a}_2$:

$$\mathbf{r} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix}, \qquad \mathbf{a}_1, \mathbf{a}_2 \in \mathbb{R}^3. \tag{3.1}$$

A valid rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ is recovered via a Gram–Schmidt orthogonalisation:

$$\mathbf{b}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|}, \tag{3.2}$$

$$\mathbf{b}_2 = \frac{\mathbf{a}_2 - (\mathbf{b}_1^\top \mathbf{a}_2)\mathbf{b}_1}{\left\|\mathbf{a}_2 - (\mathbf{b}_1^\top \mathbf{a}_2)\mathbf{b}_1\right\|}, \tag{3.3}$$

$$\mathbf{b}_3 = \mathbf{b}_1 \times \mathbf{b}_2, \tag{3.4}$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix}. \tag{3.5}$$

This construction guarantees $\mathbf{R}^\top \mathbf{R} = \mathbf{I}$ and $\det(\mathbf{R}) = 1$, ensuring $\mathbf{R} \in \mathrm{SO}(3)$ while maintaining continuity of the representation.

The full predicted pose is therefore

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{3.6}$$

Additionally the scene head outputs a 256-dimensional latent vector $\mathbf{z}_i$ which contains the information required to build a rendering of the scene that the two input frames come from.

Hypernetwork

In this architecture the hypernetwork $\boldsymbol{\Psi}$ uses the latent vector $\mathbf{z}_i$ to output the weights for the rendering network $\boldsymbol{\psi}_i$,

$$\boldsymbol{\Psi}(\mathbf{z}_i) = \boldsymbol{\psi}_i. \tag{3.7}$$

Here the encoder has already distilled scene-specific information from the images, which ideally is independent of the camera geometry. The operation of the rendering network is on the level of light rays, requiring no camera model to generate new images. In this way, the hypernetwork is able to be used as a tool for multiple cameras, giving the rendering network an initialisation which may be used to train the extrinsics and intrinsics of unknown vision sensors.

Neural Fields for Supervision

We utilise the rendering from the light field network to train pose and camera intrinsics. There are a few key benefits from using view synthesis from a neural light field. Firstly, it is ray-based, providing a general model that extends to a large family of cameras, and resulting in novel views of sufficient visual fidelity ideal for use as supervision of odometry. Secondly, the implementation is fully differentiable which allows for an end to end system to be developed in which the input image into the neural field can be learnt. We utilise this second notion to learn camera parameters through a differentiable camera model.

The chosen light field network approximates a continuous scene in the form of an MLP $\mathbf{F}_{\boldsymbol{\psi}} : (\mathbf{o}, \mathbf{d}) \mapsto \mathbf{c}$ with weights $\boldsymbol{\psi}$. This formulation uniquely maps the ray direction, $\mathbf{d}$, through some origin, $\mathbf{o}$, using a Plücker coordinate encoding to a colour, $\mathbf{c}$. As noted by Sitzmann et al. [95], whilst providing a compact and unique encoding, complex phenomena such as occlusion are not readily dealt with. The utilisation of a preceding frame to inform the hypernetwork and sequential camera motion enables the rendering network to avoid this shortcoming by evolving the scene representation over a trajectory. For this work we propose using a LFN in preference to alternatives because of its speed advantages.

3.3.2 Camera Modelling

Camera parameters are estimated in two parts: intrinsics, consisting of focal lengths (f_x, f_y) and principal points (c_x, c_y), and a distortion model. Together these describe a large family of cameras, except those not well described by a pinhole projection.

Focal Length Estimation

The estimation of the focal length is performed through back-propagation of the rendering MLP. Setting the focal length as a tuneable parameter that can be optimised allows the network to change the physical model of the camera as it generates scene geometry. We found through experimentation that the camera model was fairly robust to intrinsics initialisation. However, if initialised outside a sensible range, the focal length would get stuck in a local minima. To ensure convergence of the focal length it is initialised as the width of the image in pixels following [115], this will typically place the focal length within a suitable error margin.

Given focal length is directly correlated to the scale of the geometry seen on the sensor, and we seek to jointly learn a representation of the scene and the camera values, small changes to focal length compared to geometry can trap the network in local minima. To this end, we use a higher learning rate to converge camera parameters prior to the network learning substantial scene geometry.

Implicit Non-Linear Distortion

To deal with the generality of cameras, and to avoid the limitations of any single camera model, we model the non-linear distortion using an MLP, $\mathbf{D}(u, v) = (\Delta x, \Delta y)$. The MLP determines a *correction* to coordinates on the camera plane using pixel coordinates (u, v). This pixel coordinate-based MLP effectively is modelling a differentiable, smooth and continuous distortion function of sufficient complexity to encompass most cameras covered by a pinhole model. Figure 3.4 shows a diagrammatic representation of the MLP based distortion function in conjunction with the camera

model. Formally the ray formulation process is as follows, for given pixel coordinate (u, v). The points on the image plane x_d and y_d are calculated,

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \tag{3.8}$$

Using the output $(\Delta x, \Delta y)$ from the distortion MLP $\mathbf{D}$, the points in camera can be found as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \mathbf{D}(u, v). \tag{3.9}$$

Given the undistorted normalized image-plane coordinates (x, y), a point on the image plane in camera coordinates is

$$\mathbf{p}_C = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{3.10}$$

Let the camera-to-world transformation be

$$\mathbf{T}_{C \to W} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{3.11}$$

which has been estimated by the encoder network. The ray direction in world coordinates is

$$\mathbf{d}_W = \mathbf{R}\mathbf{p}_C, \tag{3.12}$$

and the ray origin is

$$\mathbf{o}_W = \mathbf{t}. \tag{3.13}$$

The final Plücker ray for a given pixel becomes

$$\mathbf{r}_W = (\mathbf{d}_W,\ \mathbf{o}_W \times \mathbf{d}_W). \tag{3.14}$$

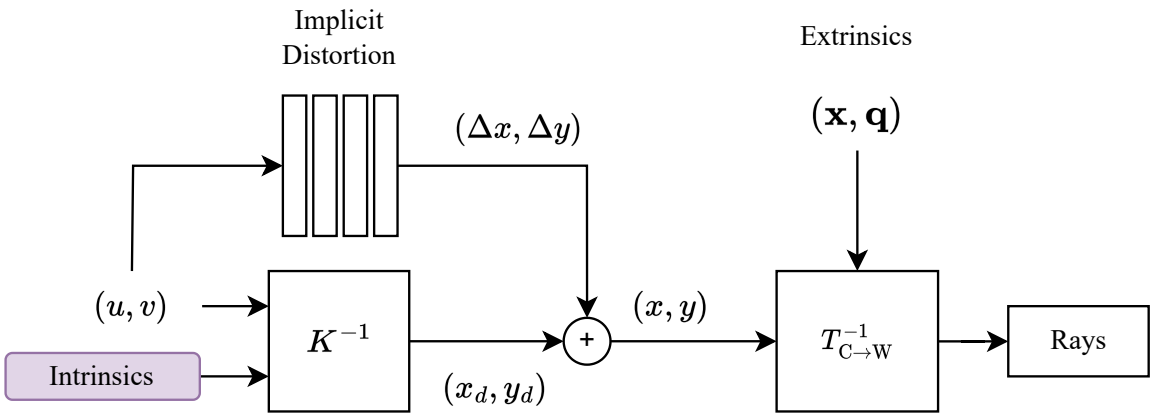

Figure 3.4 – Model of camera parameter estimation. The pixel coordinates (u, v) and the intrinsics K are first used to calculate the distorted points on the image plane (x_d, y_d). A distortion network takes as input (u, v) and determines the distortion values on a per pixel basis and produces a Δx and Δy. These corrections are applied to the points on the image plane producing undistorted points (x, y), which are converted to rays in world space by the homogenous transformation $T^{-1}_{C\to W}$.

The use of an MLP to model distortion provides a flexible and data-driven alternative to traditional parametric camera models. Rather than prescribing a specific functional form (e.g., polynomial radial or tangential distortion), the network learns a smooth mapping $\mathbf{D} : \mathbb{R}^2 \to \mathbb{R}^2$ directly from pixel coordinates to image-plane corrections.

While the cameras used in this work can be well approximated by a more conventional pinhole-based distortion model, the broader objective is to avoid mandating a specific camera parameterisation. This formulation enables extension to optical systems exhibiting complex, spatially varying, or asymmetric distortions, for which conventional low-order parametric models may be insufficient. The proposed implicit model therefore prioritises representational generality, allowing the calibration process to adapt to the data rather than constraining it to a handcrafted alternative.

3.3.3 Semi-Supervised Learning

In previous unsupervised learning of odometry work the camera parameters were required to be known and images undistorted prior to entering the pipeline [129]. In this work we proposed learning the camera parameters in addition to relative pose

within the pipeline. This adds a substantial degree of complexity to the network, which is less constrained.

To reintroduce some constraints, a small amount of labelled data is used to semi-supervise the network. This allows us to directly impose a loss on the encoder, instead of having to backpropagate through the hypernetwork and allows us to confine the pose to metric terms. Previous unsupervised works had to scale the results after training to recover a metric scale [129]. By imposing the learning of the camera parameters, we enable direct recovery of relative pose by avoiding scale ambiguity.

The trade-off between needing to know the camera parameters and needing a small amount of labelled data is often a preferable one. Using a platform such as a robotic arm enables a set of ground truth poses to be acquired, irrespective of the camera installed on-board, along a pre-defined trajectory. Novel cameras may require extensive processes to acquire accurate data for direct calibration, which may be costly to obtain in large volumes. Using a small amount of this data, it enables the network to extend automatically and generalise to new calibrations.

We demonstrate that this semi-supervised approach outperforms both a fully supervised and unsupervised approach. See Table 3.2 for results.

3.3.4 Training Losses

Similar to other works in neural rendering, we employ a photometric loss term $\mathcal{L}_{\text{ph}}$ as the primary loss function of our network between ground truth $\mathbf{c}$ and predicted $\hat{\mathbf{c}}$ pixel colours. This is calculated for all rays $r \in \mathcal{R}$, where $\mathcal{R}$ is the set of rays captured by an image,

$$\mathcal{L}_{\text{ph}} = \sum_{r \in \mathcal{R}} |\mathbf{c} - \hat{\mathbf{c}}|_2^2. \tag{3.15}$$

Where images have labelled poses during training, denoted as $\mathcal{I}'$, we enforce simple L_2-norms between the translations $\mathbf{x} \in \mathbb{R}^3$ and rotation matrices $\mathbf{R} \in \text{SO}(3)$,

$$\mathcal{L}_{\text{trans}} = \sum_{\mathcal{I}'} |\mathbf{x} - \hat{\mathbf{x}}|_2^2, \tag{3.16}$$

$$\mathcal{L}_{\text{rot}} = \sum_{\mathcal{I}'} |\mathbf{R} - \hat{\mathbf{R}}|_2^2. \tag{3.17}$$

Finally, we encourage the latent space to have a mean of zero by assuming a Gaussian prior [95],

$$\mathcal{L}_{\text{enc}} = \sum_{\mathcal{I}} \text{mean}(\mathbf{z}). \tag{3.18}$$

The overall loss function hence encompasses a loss from rendering, any available pose supervision and an imposed constraint to the latent space

$$\mathcal{L} = \lambda_{\text{ph}}\mathcal{L}_{\text{ph}} + \lambda_{\text{trans}}\mathcal{L}_{\text{trans}} + \lambda_{\text{rot}}\mathcal{L}_{\text{rot}} + \lambda_{\text{enc}}\mathcal{L}_{\text{enc}}. \tag{3.19}$$

3.3.5 Curriculum Learning

Given the challenges of jointly estimating scene and camera parameters, we employ a curriculum learning approach to sequentially recover camera parameters within an evolving neural scene. Initially, the encoding and hypernetwork are trained with fixed initial camera intrinsics, providing a rough low-frequency representation of the scene. This rough representation provides sufficient geometry to begin supervising the camera model. Prior to the geometry being fully converged, we enable the tuning of focal length in a simple pinhole model, allowing for adjustment of scene scale. Finally, the full implicit distortion model is added, giving a metric and geometrically representative scene representation, and providing a camera calibration. Learning in this way lets the network avoid local minima as higher frequency scene information is learnt.

3.4 Results

3.4.1 Datasets

We demonstrate the results for the proposed system on both real and synthetic data, showing the system working for multiple cameras and scenes.

The real-world dataset used was part of the LearnLFOdo Dataset [27]. While this dataset was originally collected using a light-field camera, the method proposed in this work operates on monocular imagery. Therefore, only the central sub-aperture image is used, which corresponds to a standard monocular camera view. This allows the dataset to be used directly while maintaining the same camera trajectories and scene diversity.

The LearnLFOdo dataset was captured using an EPIModule from EPIImaging, mounted on a UR5e robotic arm[1] executing multiple camera trajectories. The images are rectified using the calibration provided.

In total the dataset contains 46 camera trajectories captured across a range of indoor tabletop scenes. Following the protocol established in [27], the dataset is split into 37 trajectories for training, 6 trajectories for validation, and 3 trajectories for testing. The test split also contains objects and scene configurations not present during training or validation, enabling evaluation of the system's ability to generalise to unseen environments.

Ground-truth camera poses are provided for all trajectories via the robotic arm, allowing quantitative evaluation of the calibration and pose estimation accuracy.

The rendered dataset was trained separately to provide comparison between multiple cameras with defined intrinsics and distortion. This allowed for repeatable scene configurations and trajectories with multiple cameras of different distortion values. This was rendered using Blender. A single indoor shop scene was rendered with random motions ranging from 0.02-0.07m per frame. This scene was rendered under a range of radial distortion parameters and focal lengths, the principal points were not modified. This provided ground truth camera intrinsics for accuracy benchmarking.

3.4.2 Implementation Details

[1] https://www.universal-robots.com/products/ur5e/

Both the hypernetwork and the LFN are implemented as 6-layer MLPs with rectified linear unit (ReLU) activations [2]. The hypernetwork contains 256 hidden units per layer, while the LFN contains 128 hidden units per layer.

The encoder network is a 7-layer CNN with kernel size 3×3 to preserve spatial resolution. The pose head consists of a 3-layer CNN using 1×1 convolutions, while the scene head is a 3-layer fully connected network with 256 hidden units per layer. The implicit distortion model is implemented as a 4-layer MLP with 8 hidden units per layer.

All MLPs use Xavier uniform initialisation [35]. The latent scene representation has dimensionality 128. Positional encoding with 8 frequency bands is applied to ray inputs when using Plücker parameterisation.

Training is performed with a batch size of 4 for 1000 epochs. Relative frame pairs are sampled with a maximum temporal gap of 8 frames.

Loss Weights. The loss weights defined in Equation 3.19 are set to

$$\lambda_{\text{ph}} = 100, \quad \lambda_{\text{trans}} = 30, \quad \lambda_{\text{rot}} = 20, \quad \lambda_{\text{enc}} = 1 \times 10^{-6}.$$

A label ratio of 20% is used for semi-supervised training.

Optimisation. Separate Adam optimisers are used for each learnable sub-module. Independent learning rates are employed to stabilise joint optimisation:

$$\text{Encoder: } 5 \times 10^{-5}, \quad \text{Hypernetwork: } 7 \times 10^{-5},$$

$$\text{Distortion model: } 1 \times 10^{-3}, \quad \text{Intrinsics: } 5 \times 10^{-1}.$$

Using separate learning rates is critical for convergence, as the camera parameters and scene representation evolve at different scales. In particular, the intrinsics require a significantly larger learning rate to escape shallow local minima during early training.

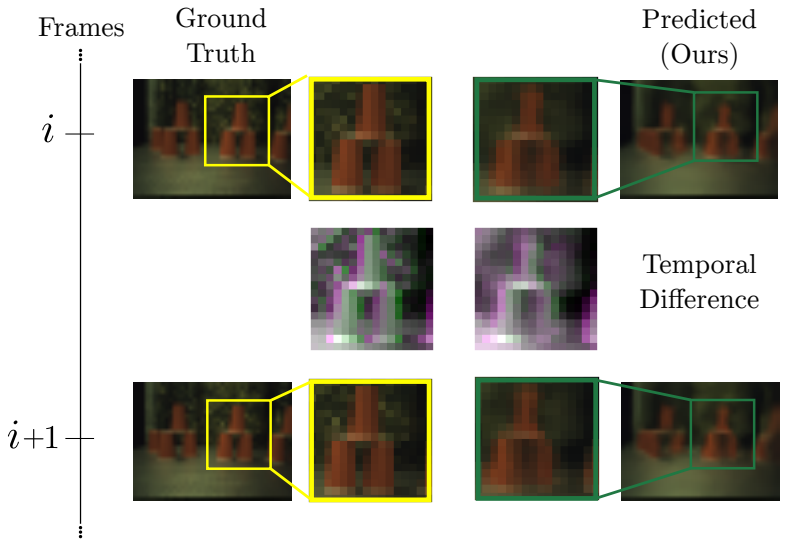

Figure 3.5 – View synthesis from the learnt LFN on a test scene. A pair of input images at times i, $i+1$ are shown on the left, with the corresponding predicted views on the right closely matching in visual appearance. The temporal difference frames shows the motion through time, it can be seen that the predicted and measured motion are very similar. Animations of this motion will be made available on the project website. The green and pink colours shown in the temporal figures represents positive and negative differences respectively.

Curriculum Learning Schedule. To improve stability when jointly estimating scene geometry and camera parameters, a staged curriculum is employed.

During the first 25 epochs, camera intrinsics are fixed and only the encoder and hypernetwork are trained. This allows the network to form a coarse, low-frequency scene representation.

From epoch 25 onward, the focal length parameters (f_x, f_y) are released and optimised jointly with the scene representation. This stage enables recovery of the correct metric scale.

From epoch 50 onward, the implicit distortion model is activated and optimised. Delaying distortion learning prevents the network from prematurely compensating for poor geometry estimates via high-frequency pixel warping.

Table 3.1 – Evaluating camera parameter estimation. Our method is able to refine and improve on initial estimates.

Camera	Initial		NOCaL (ours)		COLMAP [93]	
	f [px]	Error $\overline{\Delta r}$	f [px]	Error $\overline{\Delta r}$	f [px]	Error $\overline{\Delta r}$
Focal: 600px, no distortion	640.0	0.0225	601.1	0.0008	599.4	0.0003
Focal: 600px, large distortion	640.0	0.0440	595.3	0.0186	602.1	0.0010

Table 3.2 – Evaluating odometry performance on captured and rendered imagery. Unlabelled calibrated refers to [129]

Method	Labelled Images	Unlabelled Images	Translation Error [m]			Rotation Error [degrees]		
			Mean	STD	RMSE	Mean	STD	RMSE
Odometry accuracy on captured indoor imagery								
Fully supervised	800	0	0.025	0.009	0.027	1.553	1.847	2.414
Unlabelled calibrated	0	8000	0.029	0.016	0.033	1.522	0.969	1.808
NOCaL (ours)	800	7200	**0.020**	**0.008**	**0.022**	**0.412**	**0.295**	**0.505**
Ablation study using rendered indoor imagery with camera distortion								
Ours no intrinsics/distortion	100	900	0.157	0.060	0.168	8.026	9.180	12.194
Ours no distortion	100	900	0.147	**0.053**	0.156	4.790	2.209	5.275
Ours full	100	900	**0.145**	0.054	**0.154**	**4.024**	**1.971**	**4.481**

3.4.3 Scene Reconstruction

As shown in Figure 3.5, the framework is able to produce novel views of scenes it has not been trained on. Given a pair of inputs with some motion between frames, a pair of predicted frames can be retrieved from the network. As these views are used as the supervisory signal for the rest of the network, the temporal difference or motion between the predicted frames should reflect the same motion between the input frames. This is the case in Figure 3.5, which indicates that the network can be supervised with this signal. The reconstructions have lost some of the high frequency scene content, however the reconstruction quality is not critical, but rather how well the renderings can supervise the motion between the frames.

3.4.4 Camera Modelling

NOCaL is able to recover accurate camera intrinsics, close to ground truth and those attained by traditional methods. Table 3.1 demonstrates an ability to recover comparable results to COLMAP [93] in the absence of distortion, validating the case of an ideal pinhole camera by a significant reduction in error. We compare results based on the mean radial shift per pixel $\Delta r = \sqrt{\Delta x^2 + \Delta y^2}$. We sample the radial distortion function used by COLMAP on a grid to obtain a comparable result owing to how the distortion network is formulated.

We note that while our method is able to significantly reduce the error in the presence of large distortion, the fixed parametric camera model used by COLMAP provides a stronger structural prior and therefore offers a closer approximation to the true continuous radial distortion profile. In contrast, our distortion model is implicit and jointly optimised with the camera intrinsics, including focal length.

This joint optimisation introduces a degree of redundancy in the parameterisation. In particular, changes in focal length can be partially compensated by the learned distortion field, resulting in multiple parameter configurations that could produce nearly identical reprojection errors. The problem is therefore underconstrained, as

the photometric objective alone does not uniquely identify a single decomposition between focal scaling and radial distortion.

In practice, this redundancy does not significantly degrade performance. With sufficient data and optimisation time, the model converges to a stable solution that yields accurate reprojection and odometry estimates. We further mitigate instability through the staged curriculum learning strategy described in Section 3.4.2. By delaying optimisation of focal length and subsequently the distortion model, the network first establishes a coherent geometric structure before allocating capacity to higher-frequency distortion effects. This reduces the likelihood of solutions in which distortion prematurely compensates for incorrect focal estimates.

Nevertheless, additional structural constraints on the implicit distortion model could further reduce this redundancy. While such constraints may improve parameter interpretability, we observe that the current formulation does not negatively impact the achieved odometry accuracy.

3.4.5 Odometry Results

Odometry results for NOCaL are shown in Table 3.2. We compared NOCaL to two other odometry methods: a fully supervised approach with labelled imagery, and an unsupervised approach based on [129] that requires the camera to be calibrated and imagery rectified. The unsupervised method was provided with similar numbers of unlabelled images, around 8000, representative of the availability of unlabelled imagery in practical scenarios.

While the unsupervised approach did not require any labelled data, it is scale ambiguous, with the results needing to be correctly scaled before an error can be calculated. NOCaL does not require such scaling as the labelled data during training establish scale based on semantic information in the image. Furthermore our framework does not require camera calibration or rectification.

Perhaps surprisingly, our zero-calibration approach outperformed both fully supervised and calibrated unsupervised methods. This is partially explained by the amount

of training data available to each method: NOCaL and the fully supervised approach were provided with 800 labelled images, but NOCaL also has the benefit of additional unlabelled imagery. In the case of the unsupervised calibrated method, we hypothesise that our freeform distortion model did a better job of describing the camera distortion compared to the parametric model employed in typical camera calibration.

We performed an ablation study to measure the effectiveness of our camera model – the results are shown at the bottom of Table 3.2. We tested the full NOCaL, NOCaL without a distortion model, and a version that does not adjust the camera intrinsics or distortion model. This study employed rendered imagery simulating a camera with realistic and known radial distortion. Intrinsics were initialised close to but not exactly matching correct values, in line with a typical imaging scenario. The study shows both the distortion model and intrinsic refinement play an important role in NOCaL's strong odometry performance. The presented error metrics were calculated using [42].

3.4.6 Training and Inference Time

Typical training time for NOCaL on the LFodo dataset [27] was approximately 1.5 hours. Training time was tempered by use of the LFN for rendering and use of downsampled training images. As the rendering and the odometry can be uncoupled, inference is performed on the odometry network alone, yielding an inference time of 16.9 ms, compatible with real-time applications. Performing inference of the full framework takes 33.9 ms. The networks were trained and timed on an NVIDIA RTX 3060 12GB GPU.

3.5 Discussion and Future Work

In this chapter, we presented NOCaL, a semi-supervised learning framework for joint camera self-calibration and visual odometry estimation. By integrating differentiable

ray-based camera modelling with a light field rendering network generated via a hypernetwork, we believe NOCaL represents a small step toward general-purpose, plug-and-play vision systems capable of adapting to previously unseen and uncalibrated cameras. This has the potential to benefit practitioners deploying autonomous systems in environments where labelled data is scarce and recalibration is impractical or infeasible. Examples include scientific exploration missions in marine environments, planetary surfaces, or other remote and unstructured settings, where robust visual perception must be achieved with minimal manual intervention.

The results demonstrate that the proposed framework can accurately infer both camera intrinsics, including freeform distortion parameters, and camera egomotion from unlabelled imagery, even in the absence of prior calibration data. Crucially, this was achieved using only a small set of labelled trajectories to provide metric scale, while the vast majority of training data remained unlabelled. This highlights one of the core strengths of NOCaL: its ability to leverage large volumes of existing or passively collected data without requiring dense supervision, a key requirement for scalable deployment in real-world settings.

An ablation study further underscored the value of explicitly learning camera parameters, especially the distortion model. We observed that omitting this component degraded performance. This reinforces the necessity of modelling the complex, often non-linear, behaviour of real-world optics when deploying learning-based systems in unconstrained environments.

Despite these strengths, the current implementation of NOCaL is not without limitations. First, while the framework does not require known calibration parameters, it does rely on a reasonable initialisation of the intrinsics for successful convergence. In our experiments, standard pinhole-based initialisations sufficed for conventional cameras, but more exotic geometries, such as fisheye, may require additional strategies, such as meta-initialisation or coarse-to-fine estimation techniques. Secondly, training the hypernetwork is required either jointly with calibration or before calibration can occur, this is a computationally expensive task. While it only needs to be performed

once, it does exclude some potential applications, particularly those without access to a GPU.

Looking ahead, there are several avenues for extending this work in the future:

- Generalised Camera Models: Extend the current pinhole-plus-distortion model to support more complex and unconstrained camera geometries, such as fish-eye, omnidirectional, non-central, and light field cameras. This would further broaden the applicability of NOCaL to a wider spectrum of emerging sensors.
- Online Calibration and Adaptation: Incorporate mechanisms for real-time or continuous updating of camera parameters during deployment, enabling the system to adapt to changes in intrinsic or extrinsic parameters due to thermal drift, mechanical perturbations, or sensor replacement.
- Field Deployment and Long-Term Trials: Validate the framework in long-term deployments on real robotic platforms under diverse operating conditions (e.g., dynamic lighting, motion blur, varying environmental structure) to assess its robustness and reliability in real-world use.
- Improved Initialisation Techniques: Develop robust strategies for initialising camera parameters to enhance convergence in cases where the starting estimates are far from the true values. This is particularly important for cameras with highly non-linear distortion characteristics.

Ultimately, the goal of NOCaL is to automate the integration of cameras into a system by removing calibration as a deployment bottleneck, which is achieved by the with the ray-base self-supervised prediction signal. We believe the ability to jointly estimate scene geometry, camera motion, and optical parameters in a data-driven, label-efficient manner is a critical capability for future autonomous systems: systems that must operate in the wild, interpret diverse sensor inputs, and do so without expert human oversight.

This chapter is specifically interested in removing the manual calibration process which was demonstrated on conventional images. In the next chapter we move away from these conventional images instead focusing on wide-field-of-view (FOV).

Chapter 4

Adapting CNNs for Fisheye Images without Retraining

"The eye sees only what the mind is prepared to comprehend."

— Robertson Davies

In the previous chapter we looked at addressing the challenges of calibration in deploying cameras. In this chapter we look into the separate challenge of taking a system designed for one camera and adapting it to work with another. For this chapter, we assume we have previously calibrated our cameras, either by hand or a self-calibration method. In our second technical chapter, we ask the question can we take a model designed and trained for a specific camera and adapt it to new cameras, without having to go through the time, energy, and data expressive process of retraining? Specifically we adapt CNNs, trained on conventional images, to work with fisheye and omnidirectional inputs.

Parts of this work are published as [38] and the code and project page are available at: `https://github.com/RoboticImaging/RectConv`.

4.1 Overview

Neural networks have found widespread adoption across a breadth of imaging tasks. However, adapting these networks to emerging imaging technologies generally requires gathering extensive new datasets reflective of new camera properties, even when the operating environment remains unchanged. In this chapter we propose a training-free method that modifies pre-trained neural networks to operate with previously unseen camera geometries.

We believe our approach could have applicability across a broad range of neural architectures and camera technologies. In this work, we focus on CNNs trained on conventional monocular imagery, and demonstrate adaptation to wide-FOV fisheye-lens imagery. We show why adaptation is required and why our approach performs well where previous approaches like image rectification fail. We validate our method across multiple tasks, neural networks architectures and camera types.

CNN-based architectures generally assume translational invariance, where features appear consistent across the camera's FOV. However, many camera geometries including fisheye-lens cameras do not exhibit this invariance. CNNs trained on conventional imagery therefore perform poorly with fisheye-lens imagery and, in general, applying CNNs across camera geometries yields degraded performance.

Prior work has calibrated the camera and rectified its imagery such that translational invariance holds [125]. Alternatively, they rectify to a common equirectangular projection and adapt input convolutions to that geometry [98].

However, no general mapping to a rectified image is possible without cropping and losing parts of the original image [21] (see Figure 4.1). Such methods therefore do not address all imaging geometries including fisheye.

Patch-wise methods overcome some of these limitations but incur additional computational cost as multiple projections and inferences are required. Through tuning of patch parameters such methods trade computation against accuracy, incurring extensive computational cost for high-fidelity results [98]. They also only address local deformation.

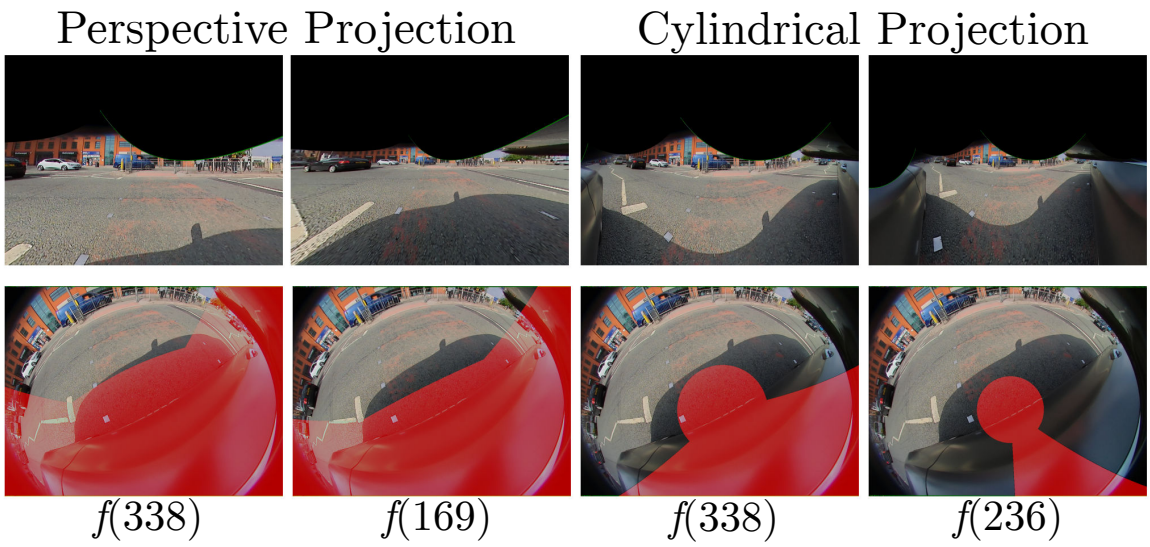

Figure 4.1 – Example of perspective and cylindrical camera projections applied to a wide field of view fisheye image. Regions in red show areas that are excluded from the rectified projection. Decreasing the focal length can reduce cropping but increases distortion.

Our approach introduces a modified convolutional layer called RectConv based on deformable convolutions [23] and spherical convolutions [98]. As depicted in Figure 4.2, rather than adapting the image to the network, we adapt the kernel shape to the image geometry.

Replacing normal convolution with RectConv layers allows pre-trained networks to operate on new imaging geometries with improved performance. To address both local and global deformation, we show RectConvs can be applied throughout the network, not only in the input layer. The only additional information required is a calibrated model of the camera, from which the RectConv deformations are computed.

The main contributions for this work are:

- The introduction of RectConv layers that enable networks to natively handle previously unseen camera geometries without requiring retraining or reprojection of input imagery;
- We develop an approach for automatically adapting existing networks to RectConv networks, allowing pre-trained networks to be applied with new cameras; and

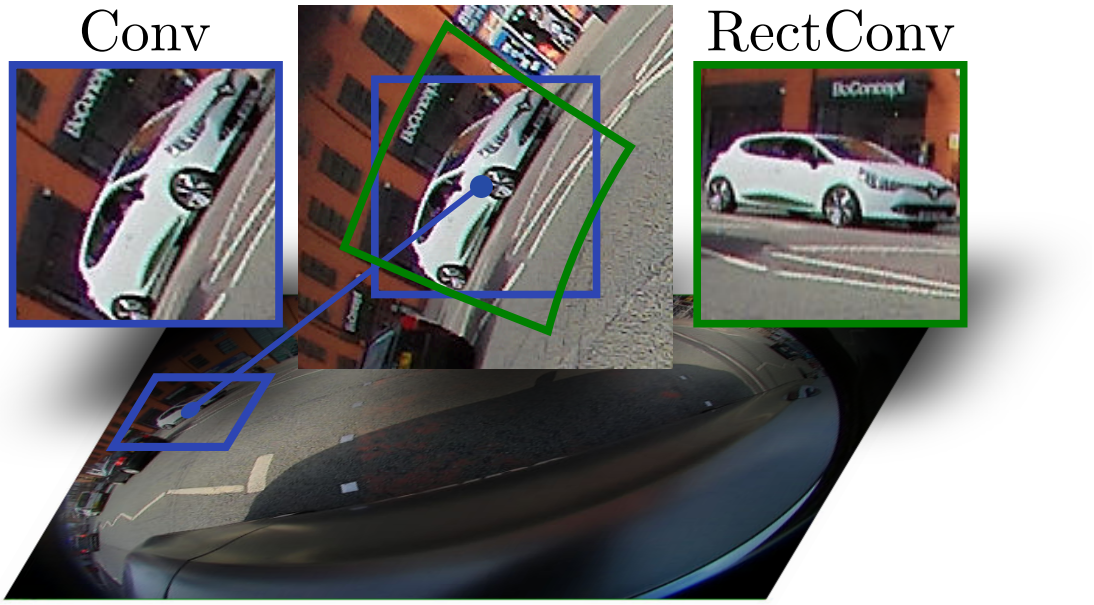

Figure 4.2 – An illustration of what regular convolution and RectConv see for a fisheye image at a given position in the image. Blue and green boxes indicate the kernel shapes for regular convolution and RectConv, respectively.

- We compare with naive and rectification-based methods, showing improved performance for wide-FOV images on multiple networks architectures, cameras, and tasks.

We believe this work will allow efficient deployment of existing solutions with a breadth of existing and emerging camera technologies. It will also reduce the energy usage of adapting to new camera geometries. While we focus on large-FOV cameras and fully convolutional neural networks, we anticipate extension to other network architectures and camera geometries is feasible.

4.2 Literature Review

Large-FOV cameras. Substantial research has gone into using large-FOV images, including fisheye [90] and panoramic [120] formats. These wider fields of view can be particularly beneficial for specific applications like autonomous driving [122]. One

common technique is to transform an existing perspective dataset to look like a large-FOV image e.g. fisheye [64, 14], to aid in the training process. This allows existing datasets of conventional perspective images to be used, but necessitates retraining per camera geometry and fails to fully capture the target domain behaviour.

While there is a trend toward transformer-based architectures, CNN-based methods remain state of the art for many fisheye applications [122]. These approaches require extensive datasets and training for the specific type of camera being used. This can be prohibitive, fails to leverage the extensive resources of existing pre-trained networks and datasets, and limits generalisation across different cameras.

Adapting convolutions. There are multiple works that aim to adapt convolutional layers to better suit a specific camera. [56] were among the first to adapt convolutions to learn spatial transformations, with their spatial transformer networks (STNs). Follow-on work proposed Active Convolutions [57] and applied a learnt offset, however this work only applied a single offset across the whole image, failing to address local deformation.

Deformable Convolutions [23, 132] are a more general version of active convolutions which learn an offset field mapping for each position in the image. This makes it much more general at the cost of an increased number of learnt parameters. Our approach builds directly upon deformable convolutions by employing camera calibration to derive a closed-form offset field to match the geometry of the input imagery, allowing us to operate directly on distorted images.

[31] introduce camera-aware convolutions which embed the camera parameters into the feature maps of the CNN. This approach addresses conventional pinhole camera geometry and it is unclear whether it would generalise to other camera geometries. In contrast, the proposed approach explicitly and efficiently addresses non-perspective camera geometries including fisheye.

Spherical convolutions. The line of work which is most closely related to ours is spherical convolution [98] and the many follow-on works [19, 29]. This work applies CNNs trained on perspective images to 360° images. [127] extend this to use trans-

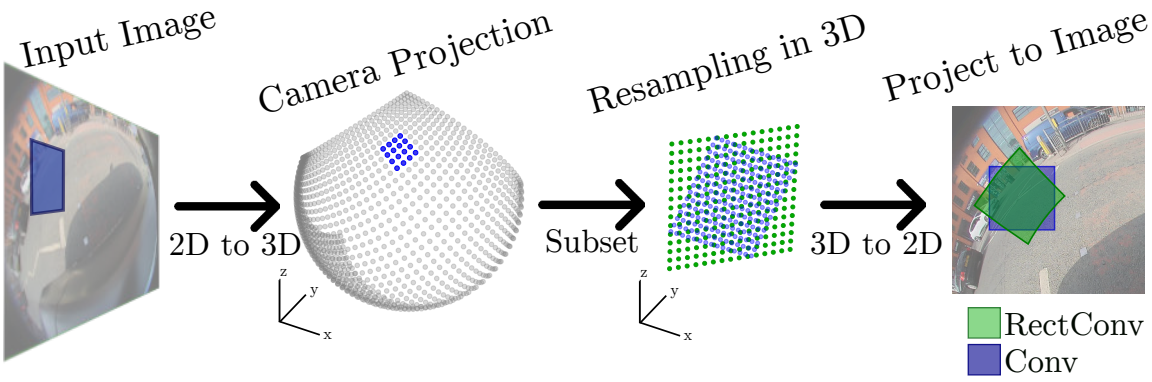

Figure 4.3 – For a given patch each pixel is converted to 3D space which is then sampled on a regular planar grid. This grid in 3D space is converted back to image locations that represent the kernel locations for that position.

formers instead of CNNs with a focus on panoramic images, and [18] employ the fast fourier transform (FFT) with increased speed and rotational invariance. However, in our case we do not want rotational invariance as rotation can be informative [99]. These approaches require additional training or fine tuning, whereas our objective is to enable adaptation without any retraining.

[103] who introduced Distortion-Aware Convolutions and [99] who introduced Kernel Transformer Networks, both have similar goals to ours in that they seek efficient adaptation of existing models from perspective imagery without retraining. They, however, focus on images given in an equirectangular projection specific to 360° imagery that does not generalise well to other camera geometries. As discussed by [125] "spherical models do not provide an accurate fit for fisheye lenses and it is an open problem". Our approach doesn't require an equirectangular projection and can handle many imaging geometries including fisheye images.

4.3 Method

4.3.1 RectConv Layers

We propose an adaptation of the convolutional layer which we call RectConv. Unlike standard convolutional layers, a RectConv layer adapts its kernel shape to match the local deformation at the point in the image that the kernel is being applied to. This local deformation results in kernel "offsets" which are calculated based on how the patch would be rectified. Figure 4.2 shows an example of the RectConv kernel shape and the corresponding view observed from that kernel. This adaptation of the convolutional layer is based on deformable convolutions, which provide a general framework for warping kernel shapes for each pixel location in an image. To achieve this, we require a way to calculate the local kernel offsets required for each pixel based on calibrated camera parameters. The offsets also need to be adjusted for each different network layer, especially for layers that modify size, such as pooling.

Camera Model. Our method requires an invertible camera model that maps between image coordinates and 3D rays in camera space. In general form this can be written as

$$p = f_{3D}(u, v, n), \tag{4.1}$$

where (u, v) are image coordinates, n is a ray length parameter, and $p = (x, y, z)$ is a point in 3D camera coordinates. The inverse mapping is

$$(u, v) = f_{2D}(p), \tag{4.2}$$

which projects a 3D point onto the image plane.

The camera model used in this work follows the polynomial fisheye projection model used in the WoodScape dataset [125]. This model represents rays by their angle of incidence with respect to the optical axis and maps this angle to a radial distance in the image through a calibrated polynomial distortion function. This formulation

enables both forward projection from 3D points to image coordinates and the inverse mapping required to generate viewing rays.

In the context of this work, f_{2D} maps a 3D point expressed in camera coordinates to distorted image coordinates, while f_{3D} converts an image coordinate into the corresponding 3D ray direction (or a point along that ray for a given depth parameter n). The explicit forms of f_{2D} and f_{3D} used in this work are provided in Section 2.1.2.

For our method to work it requires intrinsic calibration to be performed. The datasets used in this work have supplied the corresponding camera parameters from calibration.

Calculating Kernel Offsets. Here we derive the kernel offsets required at each image location. The process is depicted graphically in Figure 4.3. The first step is to convert kernel pixels to points of intersection with a reference surface in 3D space,

$$p_i = f_{3D}(u_i, v_i), \tag{4.3}$$

where i denotes the different positions in the kernel. The scale of the patch in 3D space is computed as

$$s = \frac{w_{grid} + h_{grid}}{2}, \tag{4.4}$$

where w_{grid}, h_{grid} are the horizontal and vertical size of the original grid and are calculated as $p_{max} - p_{min}$ in their respective dimensions. The rationale for this is to match the rectified kernel size to be the average of the original kernel size in the rectified space. A linear planar sampling k of scale s is calculated to be tangential to the point p_c at the centre of the original grid, from which new sample points can be calculated as

$$\hat{p}_i = p_c + k_i. \tag{4.5}$$

Here $\hat{p}_i$ is the new point in space at position i in the kernel. With the new list of points in 3D space that can be converted back to the image plane,

$$\hat{u}_i, \hat{v}_i = f_{2D}(\hat{p}_i), \tag{4.6}$$

where $\hat{u}_i, \hat{v}_i$ are the rectified pixel location on the image that the convolution should sample. For use within our framework these points need to be converted to the form

$$d_i^{offset} = (\Delta u_i, \Delta v_i), \tag{4.7}$$

where $\Delta u = \hat{u}_i - u_i$ and $\Delta v = \hat{v}_i - v_i$. This d_i^{offset} value needs to be calculated for every position in the kernel and every position in the image. To reduce computational overhead, offsets are precomputed for a subset of image locations and interpolated. As the cameras used had a continuous smooth projection, this effectively reduces computation without affecting performance.

Modifying Offset Fields.

The pre-calculated offset field must be adapted to match the spatial configuration of each convolutional layer. Let the input feature map have spatial dimensions $(H_{\text{in}}, W_{\text{in}})$, kernel size (k_h, k_w), stride (s_h, s_w), padding (p_h, p_w), and dilation (d_h, d_w). The effective kernel size under dilation is

$$k_h^{\text{eff}} = d_h(k_h - 1) + 1, \qquad k_w^{\text{eff}} = d_w(k_w - 1) + 1. \tag{4.8}$$

The output spatial resolution is therefore

$$H_{\text{out}} = \left\lfloor \frac{H_{\text{in}} + 2p_h - k_h^{\text{eff}}}{s_h} \right\rfloor + 1, \qquad W_{\text{out}} = \left\lfloor \frac{W_{\text{in}} + 2p_w - k_w^{\text{eff}}}{s_w} \right\rfloor + 1. \tag{4.9}$$

The offset field $\mathbf{O} \in \mathbb{R}^{2k_h k_w \times H \times W}$ is interpolated to match the convolutional output resolution:

$$\mathbf{O}' = \mathcal{I}\Big(\mathbf{O},\, H_{\text{out}}, W_{\text{out}}\Big), \tag{4.10}$$

where $\mathcal{I}$ denotes bilinear interpolation. Stride is implicitly handled by this resampling operation, as the output resolution reflects the stride-induced subsampling.

When dilation is used, the sampling locations are spaced by d_h and d_w pixels. To preserve geometric consistency, the magnitude of the offset vectors must be scaled accordingly:

$$\mathbf{O}' \leftarrow \begin{bmatrix} d_h \\ d_w \end{bmatrix} \odot \mathbf{O}', \tag{4.11}$$

where $\odot$ denotes element-wise scaling of the horizontal and vertical offset components.

Finally, when the feature map resolution is reduced by a scaling factor α (e.g., due to pooling or strided convolution), the distortion field must be scaled both spatially and in magnitude for a given pixel location (u, v):

$$\mathbf{O}_\alpha(u, v) = \alpha\, \mathbf{O}(\alpha u, \alpha v). \tag{4.12}$$

This ensures that offsets remain expressed in the correct coordinate system of the current feature map resolution. Together, these transformations guarantee that the deformable sampling grid remains geometrically consistent across layers with varying stride, dilation, padding, and spatial resolution.

Conversion from Conv to RectConv Layers. Conversion from a conventional CNN network to a RectConv version can be carried out efficiently and elegantly. Given a pre-trained model and camera parameters, a recursive search through the network modules identifies all the convolutional layers and replaces them with a RectConv layer. Offsets for the RectConv are computed as in the previous section, and weights and bias terms from the pretrained network are left unmodified. All convolution layers with a kernel size greater than one are converted to a RectConv layer in this manner. Layers with a kernel size of one require no modification and are left unchanged. The conversion is applied to all layers to address both local and global distortions.

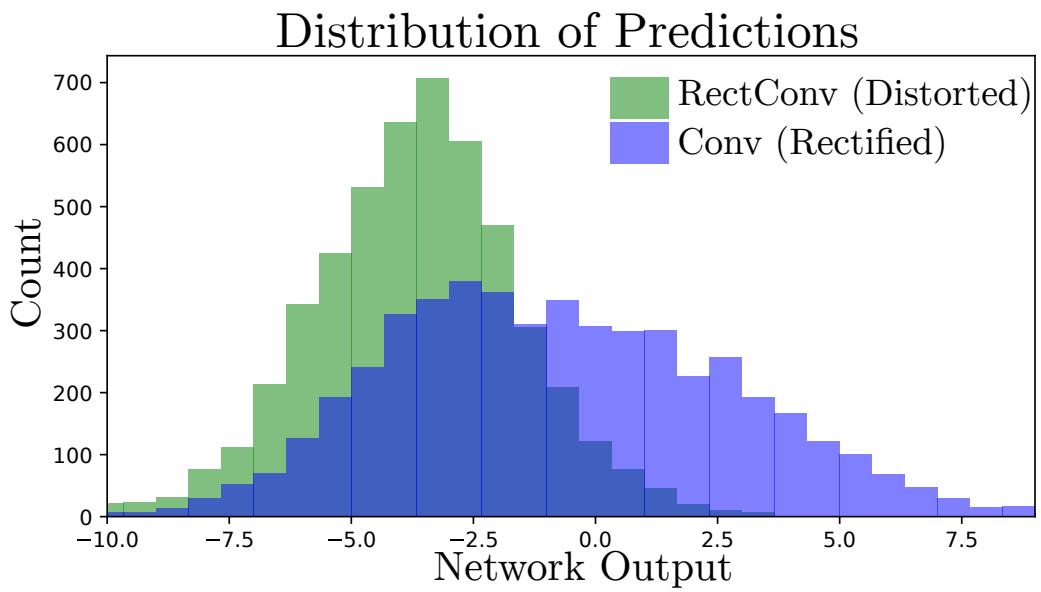

Figure 4.4 – A histogram of the outputs from a binary classification task showing how a RectConv layers result in a bias shift in the outputs.

4.3.2 Effects of Interpolation

A consequence of employing non-integer offset fields in the deformable convolutions is that samples must be interpolated from the input imagery. Our implementation employs bilinear interpolation [23]. This process is information destroying and is present in every RectConv layer. The slight error at each layer accumulates as it propagates through the network.

Figure 4.4 illustrates the effect of interpolation on the output of a network. This experiment shows a histogram of a binary classification network's output before a final classification layer is applied. The network used for this demonstration was a simple CNN, which has 4 convolutional layers with kernel sizes of 7, 5, 5 and 3, with no padding or stride. Then 4 additional 1×1 convolutional layers. The network was trained on a binary classification task involving cats and dogs. The figure compares the convolutional form of the network applied to rectified perspective imagery, and the RectConv version applied to a distorted version of the same images. For an ideal conversion between convolutional and RectConv networks the outputs would

be identical. However a shift in the distribution is evident, and we hypothesise this arises due to the compounded impact of interpolation in the RectConv approach.

It is important to note that this behaviour may also depend on the specific interpolation strategy used. In this work bilinear interpolation was selected due to its computational efficiency and its common use in deformable convolution implementations. However, alternative interpolation methods (e.g. bicubic or higher-order schemes) may reduce interpolation artefacts and potentially mitigate the accumulation of error across layers. A systematic investigation into the effect of different interpolation methods on RectConv performance is therefore a promising direction for future work.

While these results show RectConv conversion is imperfect, it nevertheless demonstrates competitive performance in adapting to new camera geometries without a need for retraining. We leave further exploration and mitigation of the impact of interpolation as future work.

4.3.3 Supported Model Architectures

RectConv layers can be applied to any convolutional layer. However, there are some criteria needed for the model architecture to be a strong candidate for RectConv conversion. The model should not have a fixed input image size, instead it should be able to accept an image of arbitrary size. This is required as the perspective images used to train the model will not be the same size as the target (e.g. fisheye) images that will be used for inference. One architecture family with this behaviour is fully convolutional networks.

Given these considerations we chose to demonstrate our approach using fully convolutional networks [76]. These architectures have seen success on a wide range of computer vision tasks. They do not have any fully connected layers and natively accept images of different sizes. We have also chosen networks that have no deconvolutional operations. While we do not demonstrate adaptation of deconvolutional

layers to a rectified alternative, we believe generalisation is feasible and leave this as future work.

4.3.4 Fine-Tuning

For this work we are explicitly interested in how networks can be adapted to new cameras without any additional training. We acknowledge that performing fine-tuning may enhance performance in many applications and mitigate the bias shift due to interpolation discussed in Section 4.3.2.

4.4 Results

Tasks. We believe the proposed approach is general and applicable across many vision tasks. There are however certain tasks which are better suited to RectConv conversion than others. Size-conserving and pixel-wise labelling tasks such as segmentation and depth estimation are a strong fit, and for this reason we chose segmentation to demonstrate the effectiveness of the approach.

More challenging tasks have outputs with different dimensions to the input. A key example is object detection for which the outputs are a list of bounding boxes with pixel locations. We chose this task as a more challenging example for RectConv networks. An interesting side effect of conversion to RectConv layers is that because the kernels only see rectified patches the bounding box extents have been rectified locally around the object being bounded. This necessitates an additional step in which the rectified box extents are projected back to the original input image.

Datasets. We demonstrated our approach on imagery from three different cameras drawn from two separate datasets. Firstly, Woodscape [125] is a multi-task, multi-camera fisheye dataset. Woodscape has four fisheye cameras deployed on a vehicle, with data collected throughout a city environment. We demonstrated results using two cameras which capture the diversity of imaging geometries present in the dataset.

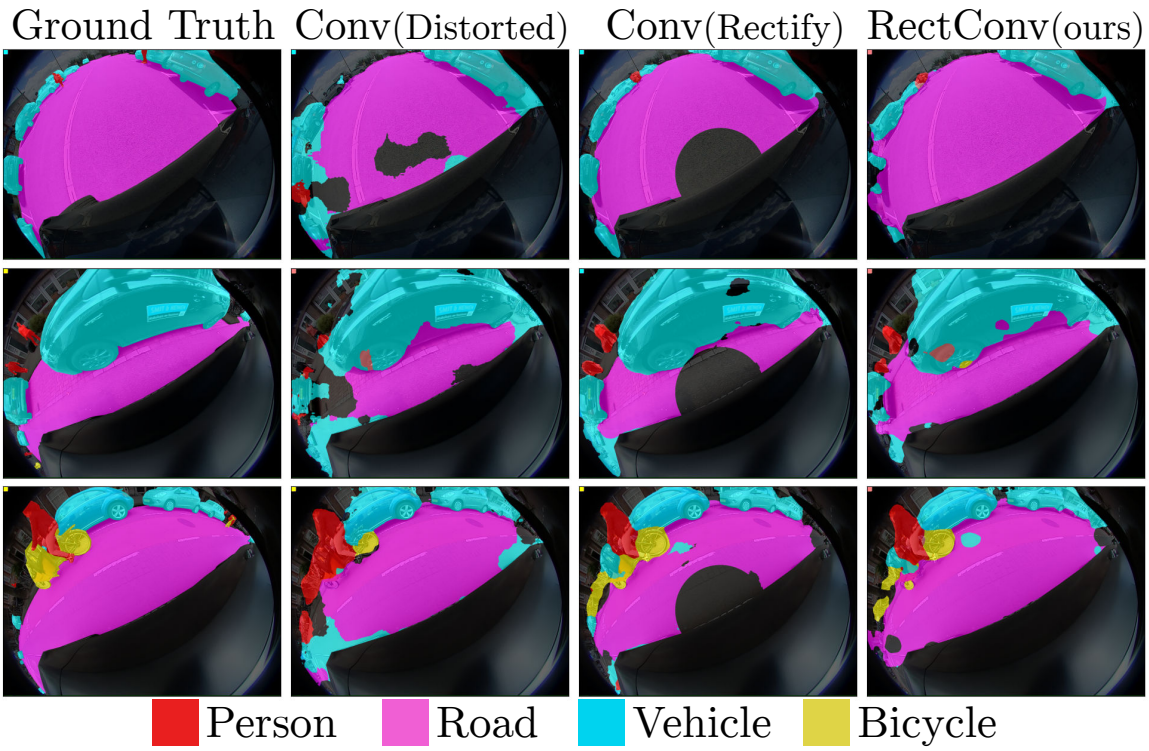

Figure 4.5 – Comparison of segmentation using a FCN-Resnet101 pre-trained on Cityscape. The unmodified pre-trained network shows poor performance; pre-rectification shows poor performance and suffers from dead zones that could not be included in the rectification; and the proposed RectConv shows the strongest performance while covering the entire image.

The second dataset we use is PIROPO [24]. This dataset tracks people moving around a room from both a perspective and omnidirectional camera across multiple sequences. We used only the omnidirectional camera and demonstrated both segmentation and detection. The ground truth data provided includes a single labelled point for each person.

These datasets provide calibration parameters and are described with a radial distortion modelled using a 4th order polynomial.

Models. We demonstrated our approach adapting four different pre-trained segmentation models constructed from two different backbones, ResNet50 and ResNet101 [48], and three separate architectures, FCN [76], DeepLabV3 and DeepLabV3+ [13]. We used an FCOS ResNet50 [104] for object detection. These are representative of stan-

Table 4.1 – Comparison of segmentation MIoU and pixel accuracy for pre-trained models applied to fisheye imagery from the Woodscape dataset.

Camera	Method	FCN(Resnet50)		FCN(Resnet101)		DeepLabV3(Resnet50)		DeepLabV3+(Resnet101)	
		Pixel Acc(↑)	MIoU(↑)	Pixel Acc(↑)	MIoU(↑)	Pixel Acc(↑)	MIoU(↑)	Pixel Acc(↑)	MIoU(↑)
Camera 1	Conv(Distorted)	83.16	24.90	82.60	24.90	73.64	22.68	81.03	22.76
	Conv(Rectify)	82.05	27.01	85.27	28.13	72.64	19.38	86.45	29.02
	Conv(Patches)	87.55	29.98	88.57	29.8	84.31	28.04	**91.54**	**32.47**
	RectConv(Ours)	**87.68**	**31.05**	**89.12**	**31.68**	**84.61**	**29.28**	89.56	30.84
Camera 2	Conv(Distorted)	84.64	24.09	84.97	24.53	73.12	20.74	79.77	21.31
	Conv(Rectify)	83.56	24.88	85.14	25.62	77.81	21.28	86.21	26.37
	Conv(Patches)	89.09	27.84	89.44	27.66	**85.72**	25.83	**91.48**	**28.54**
	RectConv(ours)	**89.59**	**28.75**	**89.68**	**28.83**	84.78	**26.23**	89.20	27.17

Table 4.2 – Comparison of segmentation and detection using pre-trained models on fisheye imagery from the PIROPO dataset.

Method	Segmentation Network						Object Detection Network		
	FCN(Resnet101)			DeepLabV3(Resnet101)			FCOS(ResNet50)		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Conv(Distorted)	69.09	32.90	44.57	64.10	28.86	39.80	84.64	63.64	72.65
RectConv(ours)	**76.91**	**48.05**	**59.15**	**81.00**	**44.30**	**57.28**	**86.67**	**65.66**	**74.71**

dard models for segmentation and detection, while also having readily available pre-trained weights.

Pre-trained Networks. Each test required a pre-trained convolutional network to be converted to the RectConv version.

For the Woodscape dataset all pre-trained models were trained on the Cityscape dataset [20]. Segmentation was evaluated using only the classes present in both Cityscape and Woodscape. In the case of DeepLabV3+ we use a publicly available network pre-trained on Cityscape. For the PIROPO dataset [30] all pre-trained models were trained on Pascal VOC. We used the pre-trained models supplied by pytorch which are readily available. For this test, only the person class was used and the other available classes were ignored.

4.4.1 Results: Woodscape

Table 4.1 and Figure 4.5 show quantitative and qualitative results for the Woodscape dataset. We compared our method to three baseline approaches: the naive method of applying the pre-trained network directly to the distorted fisheye image; pre-rectifying the input images before inference using a cylindrical projection, as this projection maintains a larger field of view compared to other projection; and a patch-based approach which splits the image into multiple patches, rectifies them individually, and then runs inference on each patch. The last of these required manual adjustment to our specific application, such as patch size, location, overlap.

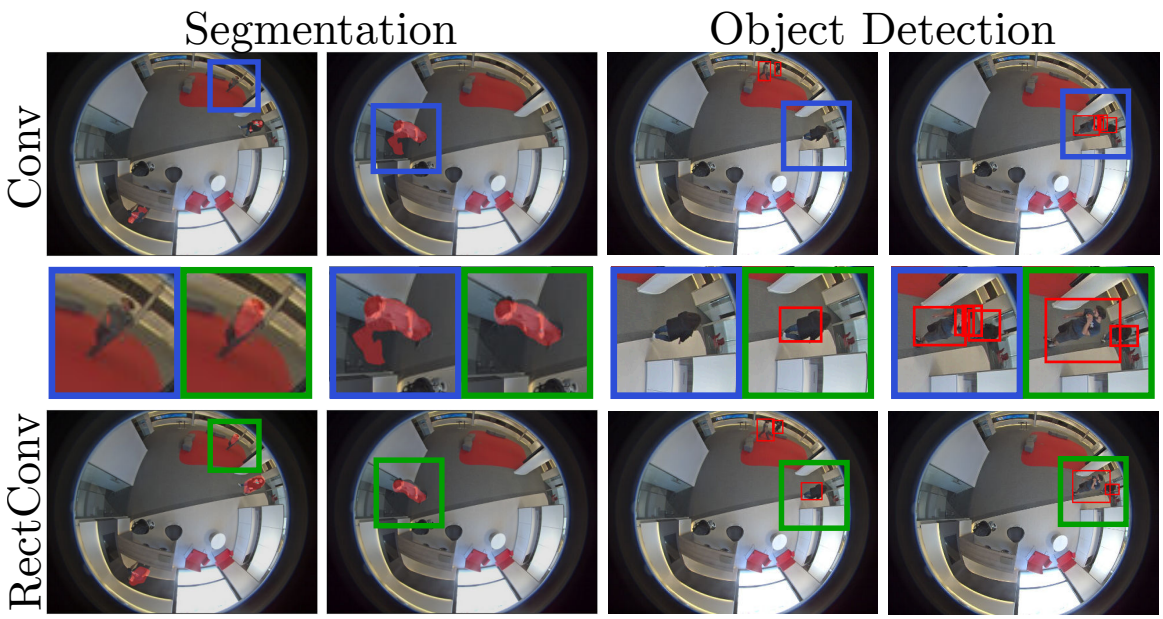

Figure 4.6 – Detection and segmentation results for people on the PIROPO dataset [24] using pre-trained segmentation (FCN-Resnet101) and object detection (FCOS-Resnet50) networks. RectConv has significantly improved the segmentation results and has a significant improvement in bounding box detections.

Our method outperforms the alternatives with stronger quantitative results, as well as stronger qualitative results that cover the entire FOV when compared to the Distorted and Rectify methods. Importantly, our approach requires no additional training, only a one-time closed-form conversion of the convolutional layers to their RectConv alternatives. The performance of the patch-based method is only slightly lower, as expected, since rectifying input patches solves many of the issues of the other approaches. We postulate the smaller performance gain seen for the DeepLabV3+ is due to head performing less standard convolutions operations. The main drawback of the patch-based method is the significant increase in inference time as discussed in Section 4.2 and demonstrated in the evaluation of inference time below.

Figure 4.5 clearly illustrates the differences between approaches and how they compare to the ground truth. For the pre-rectification method the results are distorted back to the original image geometry, making the cropped dead zone clearly visible in the centre of each segmentation mask. The additional errors from using a distorted input

Table 4.3 – Inference times using different methods. The % increase compared to the standard convolutions is shown.

Model	Inference Time Seconds - (% Increase)		
	Conv(Distorted)	Conv(Patches)	RectConv(Ours)
FCN Resnet50	0.30	0.83 (177%)	0.46 (53%)
FCN Resnet101	0.41	1.15 (180%)	0.66 (60%)
DeepLabV3 Resnet50	0.32	0.91 (182%)	0.55 (71%)
DeepLabV3+ Resnet101	0.25	0.69 (179%)	0.38 (52%)

are also clearly visible for the convolution method, especially looking at the road prediction.

4.4.2 Results: PIROPO

Results for PIROPO are shown in Table 4.2 and Figure 4.6. As there is no ground truth segmentation available for this dataset the quantitative results are shown as an accuracy of correct detection of people within the image. These results show that, not only does converting layers to RectConv increase true detection, it also reduces spurious detection. From the qualitative results we can see that the segmentation masks are cleaner compared to the naive approach.

In object detection, RectConv outperforms the conventional approach in all measures. The extent of the quantitative benefit is not as strong in detection as it is in segmentation. The conversion of bounding boxes in the RectConv network into image space is imperfect and is a topic for future research. Typical errors from the bounding box conversion are seen in Figure 4.6.

Inference Time. The conversion of a network to use RectConv layers incurs additional inference-time computational cost due to the additional step of deforming the kernels. Table 4.3 shows the average inference time for the four networks we evaluated operating on a single image, running on an NVIDIA RTX3060, and the percentage increase in time compared to standard convolutions. Across the four different models there was an average of 180% increase time for using the patch-based method. This was the best case scenario for the patch method, where only the relevant patches in the image were selected, significantly reducing the number of patches used. There was

Table 4.4 – Effect of different RectConv layers.

RectConv Backbone	RectConv Classification Head	Pixel Acc(↑)	MIoU(↑)
✗	✗	82.60	24.90
✓	✗	88.14	30.01
✗	✓	82.63	24.92
✓	✓	**89.12**	**31.68**

a 60% increase in time for using RectConv method, while slightly outperforming the patch-based method. We also expect that given optimisation the overhead required to run our method could be reduced.

4.4.3 Ablation Study

Table 4.4 shows the results of an ablation study on how converting different parts of the network to RectConv layers affects overall performance. This study was performed on the Woodscape dataset, using the FCN ResNet101 segmentation network. It can be seen that the vast majority of the performance gain comes from converting the backbone. As the backbone is the part that is extracting geometric features it is understandable why it benefits most from RectConv. This supports the potential of the proposed approach to generalise well to other applications as these backbones are used for a range of tasks.

4.5 Discussion and Future Work

In this chapter, we introduced RectConv, a training-free method for adapting pre-trained models to new cameras geometries without any additional training or data required. We adapt a CNNs to non-standard camera geometries, such as fisheye lenses, without the need for image rectification, network retraining, or domain-specific fine-tuning. RectConv modifies the sampling behaviour of standard convolutional layers to account for camera distortion at inference time by transforming the receptive field based on a calibrated ray model. This allows networks trained exclusively on

conventional rectilinear imagery to operate directly on wide-FOV images, preserving information typically lost or degraded through naïve rectification procedures.

Empirical results across multiple tasks including semantic segmentation and object detection demonstrate the robustness and generalisability of the RectConv approach. When applied to networks trained on the Cityscapes dataset and evaluated on fisheye datasets such as Woodscape and PIROPO, RectConv consistently outperformed both baseline methods: (i) direct application of pre-trained models to distorted input, and (ii) standard pre-rectification followed by inference. These findings underscore a key contribution of this work: bridging the geometric domain gap between training and deployment without requiring data augmentation or model retraining.

This work is particularly impactful in autonomous robotics, drone-based inspection, mobile mapping, and surveillance systems, which often necessitate the use of wide-FOV or custom optics, and where retraining models for every configuration is infeasible.

In an era where the energy demands of artificial intelligence are rapidly increasing, the ability to reduce energy consumption during model deployment is becoming increasingly important, particularly as training deep networks continues to grow more computationally intensive. A key advantage of the proposed approach is that it eliminates the need for retraining when adapting to new camera geometries, thereby significantly lowering the energy cost associated with system deployment. While this work contributes a step toward more energy-efficient multimodal vision systems, there remains a broader need within the field to develop methods that explicitly prioritise sustainability and energy-aware design in both training and inference.

Despite its effectiveness, RectConv has practical limitations that must be considered in deployment scenarios. For instance, the computational overhead introduced by the distortion-aware sampling procedure. As detailed in Table 4.3, RectConv incurs a modest increase in inference time, especially on deeper CNN backbones. While acceptable for many robotics and embedded vision applications, this overhead may be prohibitive in real-time systems with strict latency constraints or limited hardware acceleration. Furthermore, as discussed in Section 4.3.2, interpolation artifacts arising

from non-uniform sampling can slightly degrade output fidelity. This degradation is likely to be more pronounced in deeper architectures, where accumulated interpolation error may attenuate feature responses or introduce bias in downstream tasks.

Another area of ongoing investigation is the scope of the network conversion process. While the current implementation handles a wide range of convolutional layers, like dilated convolution, extending support to additional non-convolutional encoder heads remains an open challenge.

In terms of future directions, we identify several avenues for extending this work:

- New tasks: While this chapter focused on dense prediction (segmentation) and detection, the same geometric adaptation strategy is directly applicable to tasks such as monocular depth estimation, visual odometry, and view synthesis, where geometric accuracy is paramount.
- Generalised camera models: RectConv requires a calibrated model of the underlying imaging geometry. While this thesis focused on fisheye imagery, the same framework can accommodate many camera models for which rays can be parameterised, including catadioptric, and even multi-aperture systems.
- Self-calibrating integration: RectConv can serve as a downstream module within a broader self-calibration pipeline. A natural extension of this work is to integrate RectConv with the NOCaL framework introduced in Chapter 3. Such an integration would yield an end-to-end system capable of learning both the camera parameters and the downstream perception model adaptation jointly, enabling true plug-and-play deployment across arbitrary imaging geometries.

By decoupling geometric adaptation from supervised learning, RectConv advances the thesis's central objective: reducing the engineering burden of deploying new cameras in vision systems, while not being task or domain specific. This approach lowers the barrier for utilising wide-FOV and non-conventional cameras, particularly in domains where labelled data is scarce or retraining is costly. More broadly, this approach offers

a promising pathway towards modular, camera-agnostic vision pipelines capable of dynamically adapting to the constraints and opportunities of novel sensor hardware.

Chapter 5

Positionally Embedded Rays for Multi-Camera, Multi-Modal Vision

---

"The eye sees only what the mind is prepared to comprehend. But sometimes, two eyes see more than one."

— Adapted from Robertson Davies

In the preceding chapters, we addressed the challenges of deploying single-camera systems, first by removing the need for manual calibration, and then by adapting existing models to new camera geometries. In this final technical chapter, we turn our attention to the complex problem of deploying multi-camera systems, particularly those composed of heterogeneous modalities. These systems are increasingly common in robotics and autonomous platforms, where combining visual cues from multiple sensors, such as RGB and thermal cameras, can offer improved robustness and performance in diverse environments.

Parts of this work are published as [39] and the code and additional visualisation are available at: `https://roboticimaging.github.io/RoRE/`.

5.1 Overview

Multi-modal and multi-camera vision systems offer significant advantages over single-sensor setups by providing richer and more resilient scene understanding. Using multiple cameras, potentially of different modalities such as RGB and thermal, enables broader spatial coverage, redundancy, and access to complementary information. This is especially beneficial in environments with dynamic lighting, occlusion, or degraded visibility, where reliance on a single modality may lead to system failure [6]. For instance, thermal cameras are effective in low-light or obscured conditions such as smoke, while RGB cameras provide high-resolution texture and colour cues essential for semantic interpretation.

However, integrating such heterogeneous inputs introduces substantial challenges. Differences in spatial resolution, photometric response, field of view, and spatial or temporal alignment make it difficult to reason jointly across modalities and viewpoints. Most existing systems rely on manually engineered fusion strategies or architectures tailored to a specific sensor configuration [79], reducing adaptability and increasing the engineering effort required for deployment across varied hardware setups.

Building on this motivation, this chapter introduces a modality-agnostic, transformer-based framework for multi-camera, multi-modal scene understanding. The proposed system accepts an arbitrary number of camera inputs, each potentially from different modalities, and constructs a unified geometrically consistent representation of the scene without relying on modality-specific processing or rigid sensor configurations. Specifically, we focus on RGB and thermal imagery as a representative case, demonstrating how this approach can fuse heterogeneous inputs to infer depth and synthesise cross-modal novel views. A high level overview is shown in Figure 5.1.

The framework operates in a fully feedforward manner, requiring no explicit regression or alignment stages. The proposed architecture is in Section 5.3, we'll describe its content over the next sections, for now know that a key source of novelty is in leveraging a novel extension of RoPE [97] that embeds both ray information and sensor modality directly into the tokens, which represent patches in images, as they propagate through the network. This design enables the transformer to reason over bundles of rays (patches) in a geometry-aware fashion, preserving spatial and modality context even across disparate sensor types. To accomplish this we require posed images, i.e. operating with pre-calibarted arrays.

Like the earlier chapters, this work continues to operate in ray space, but extends the paradigm to operate at the patch level, which aligns naturally with transformer architectures. This allows the model to learn complex spatial and semantic correspondences across modalities. Similar to Chapter 3 it uses novel view rendering as a self-supervisory signal without relying on explicit hand-engineered fusion strategies.

One limitation of the patch based approach is that it assumes neighbouring pixels correspond to neighbouring rays. This assumption does not extend to a range of camera archetypes, such as lenslet-base light fields or coded aperture where its not a natural mapping of a single ray to a single pixel. For this contribution we focus on cameras where this assumption holds.

We validate the efficiency of our embedding against the current state of the art for RGB-only inputs [60]. We then validate the multi-modal integration on simulated RGB and thermal scenes, showing the model's ability to produce coherent depth maps and accurate cross-modal image reconstructions (e.g., RGB-to-thermal synthesis). The visual results reveal that the model implicitly learns correspondence across modalities, highlighting its potential for robust real-world deployment.

The key contributions of this chapter are:

- A novel transformer architecture that uses a ray based rotary positional embedding that allows multi-camera systems to be jointly integrated into a single representation;

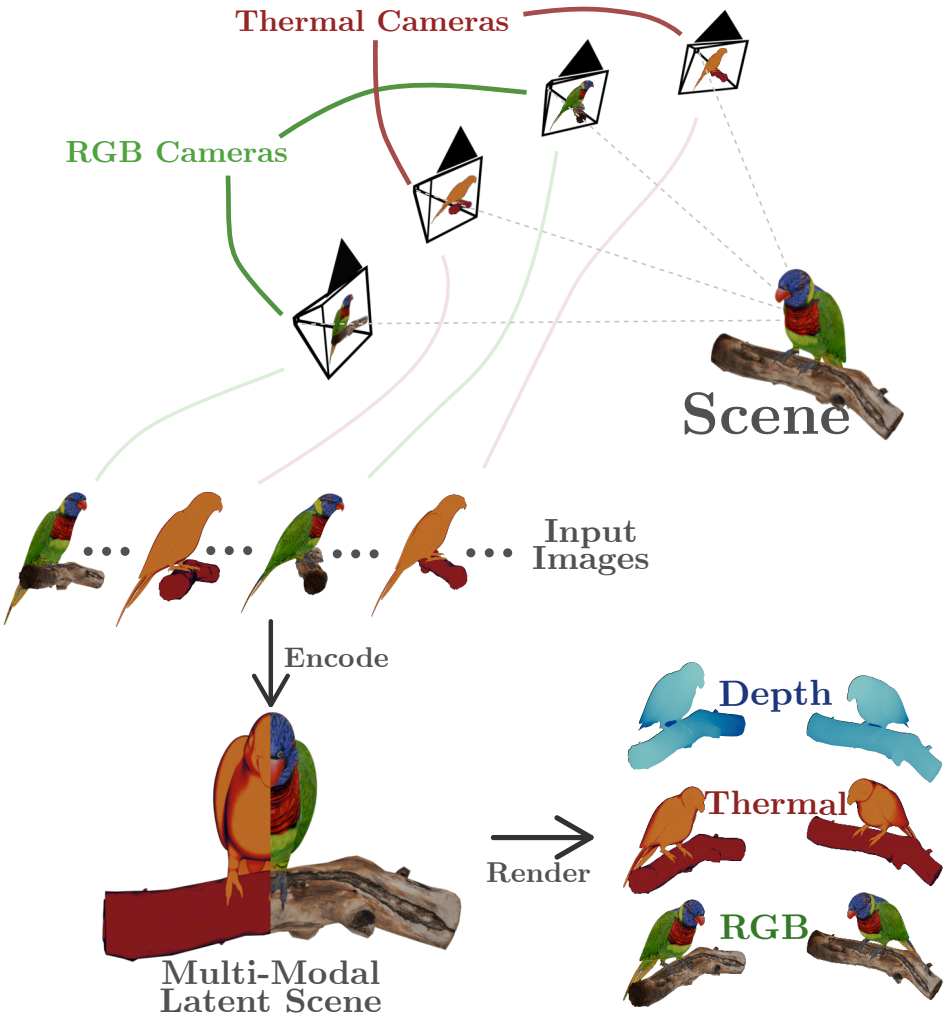

Figure 5.1 – Multiple images with different modalities are encoded using a ViT. This network can render novel views (photometrics) of the captured scene in any of the input modalities. The network also produces depth maps (geometry) of the scene.

- A multi-modal training scheme that allows for a single model to accept different configuration of modalities across an arbitrary number of cameras;
- A self-supervised training objective based on masked cross-modality view prediction that enables a robust vision pipeline that is resilient to occlusions;
- Comparison to state-of-the-art RGB-only alternative, showing faster convergence, and comprehensive evaluation of multi-modal performance under different operating configurations; and

- A simulated multi-modal dataset which consists of $\sim$ 4000 indoor rooms, with ground truth poses and depths, which will be publicly available upon publication of this work.

This work represents a step forward in building generalisable plug-and-play vision systems, capable of scaling across diverse sensing setups with minimal prior knowledge or engineering effort. We anticipate this framework to be particularly impactful in applications where multimodal perception is critical, such as in autonomous inspection, search-and-rescue robotics, and mobile systems operating across day–night cycles or visually degraded conditions. Furthermore, because the model is self-supervised, fully feedforward, and agnostic to the specific sensor arrangement, it can be be applied to countless downstream tasks in a flexible way. This lowers the barrier for integration into new platforms. In the broader context of autonomous vision, this contribution supports the shift away from rigid, task-specific vision systems toward adaptive, modality-aware architectures that generalise across hardware, environments, and missions.

5.2 Literature Review

Multimodal sensor fusion has been extensively studied across fields such as robotics, autonomous driving, and medical imaging. Traditional approaches rely on early or late fusion strategies [87], combining feature representations from each modality either at input level or decision level. However, these methods can struggle when modalities are misaligned or degraded. It is an active area of research to expand geometric reasoning to additional modalities [79, 66].

Recent advances in deep learning, particularly transformer architectures [106], have demonstrated superior capability in capturing complex relationships across inputs, due to its attention mechanism. Transformers have been extended to multimodal tasks such as vision-and-language integration [102], suggesting their potential for multimodal sensor fusion.

Bachmann et at. [4] demonstrated transformers' impressive ability to learn multi-modal image correspondences. However, this work focuses on images that are aligned so while it is able understand multiple modalities, it is unable to understand un-aligned pixels meaning it is not suitable for multimodal camera arrays because of the baseline between them.

Beyond transformer based archetectures Hassan et al. [47] adapt NeRFs [80] to work with RGB and thermal images while Chen et al. [15] adapts Gaussian splatting [63] techniques to work for thermal images. Lu et al. [79] extends it further to fuse thermal and RGB imagery into one scene understanding. These works demonstrate the benefits of complementary information across sensors and motivate extending view-synthesis transformers to multi-modal inputs, where alignment and fusion strategies remain open challenges. To our knowledge this is the first work that performs feed-forward multi-modal novel view synthesis, this is a key development of our work.

Shaw et al. [94] and following on work [53] propose schemes that embed position information relative to other positions, they show that this relative relationship has distinct benefits in multiple domains. This concept is extended further with RoPE [97] which embeds position information as rotations, that in a transformer architecture preserves this relative embedding property, we extend RoPE in this chapter.

Philippe et al. [117, 116] demonstrate a pre-training step for cross view infilling, i.e. images with a baseline, they do this with masked inputs similar to [4]. This work has impressive geometric understanding but is only designed for RGB. Extending geometric reasoning across multiple views, techniques like DUST3R [112] learn pose and scene geometry, tailored for RGB data. They demonstrate RoPE has improved the efficiency of positional encoding in vision transformers. Embedding spatial information relevant to multiple camera views remains a relatively under-explored challenge, especially for multimodal and cross-domain contexts.

Previous work that builds upon RoPE is either 2D [116] or 3D [111, 5, 75], including a temporal dimension. The proposed research builds upon these foundations by introducing a ray coordinate (6D ray space) embeddings into transformer models, enabling flexible and scalable multimodal integration across heterogeneous camera systems.

Jin et al. [60] proposed LVSM which embeds ray information into patches similar to what we propose in this chapter, they show impressive performance for NVS and are able to integrate multiple input images, however they perform this ray embedding using absolute embeddings, which we proposed could be improved upon by using a RoPE based embedding. Other work also propose using relative embedding spaces. CaPE [65] and GTA [81] investigate conditioning attention on relative camera transformations, based on rotational biases. Li et al. [71] propose PRoPE which encodes entire camera frustums as relative positional embeddings. These work use the fact that relative embedding should generalise better, however they move away from the ray-based representations, and this hinders generalisation. These approaches are also only designed for RGB images.

Self-supervised learning techniques for depth and pose estimation [129] have highlighted the power of cross-view consistency signals, while domain adaptation work [44] demonstrates the value of learning across modality gaps. He et al. [50] is recent work that learns a model which finds explicit correspondences between input images of differing types. However, to our knowledge no existing systems are able to generalise integration of information from arbitrary camera arrays and modality combinations without architecture re-design.

5.3 Method

We present the method used to achieve flexible, geometry-aware multi-camera, multimodal scene understanding. The proposed system operates under the assumption that all cameras are posed and calibrated, meaning their intrinsics and extrinsics are known in advance. This is a valid assumption for many practical setups where camera arrays are rigidly mounted, such as in autonomous vehicles, drones, or fixed surveillance platforms. Although recent approaches, such as VGGT [109], seek to learn these camera parameters, we argue that when camera poses are in a fixed relationship, this information offers valuable geometric structure that should be exploited.

Our work extends the capabilities of prior approaches like MultiMAE, which demonstrated the feasibility of cross-modal fusion within transformer-based architectures. However, MultiMAE was limited to confocal setups, where all modalities originate from the same spatial viewpoint. This assumption does not hold in most real-world camera arrays, where physical baselines between sensors complicates fusion but introduce geometrically informative perspective variation. Our method is designed to handle these non-confocal configurations, leveraging baseline-induced disparities to improve spatial understanding.

The core of our approach involves embedding both ray geometry and modality information into the transformer architecture in a way that enables consistent reasoning across views and sensor types. We build on the premise, echoed throughout this thesis, that operating in ray space, rather than image space, provides a unifying representation that facilitates integration across dissimilar imaging systems. The following sections detail our embedding strategies, architectural design, masking formulation, and training objectives that collectively enable high-fidelity reconstruction and depth estimation in multi-camera, multi-modal environments.

5.3.1 Embedding Ray and Modality Information

A central challenge in multi-camera, multi-modal vision is enabling the model to reason coherently across different viewpoints and sensor types. While transformer architectures are naturally well-suited for learning correspondences between disparate inputs, they rely heavily on positional encodings to inject spatial structure into the otherwise permutation-invariant attention mechanism. For multi-view, multi-modal perception, it is therefore critical to embed information about both camera pose and sensor modality in a form that enables the model to align and fuse observations into a unified, geometry-consistent representation.

To address this, we propose an embedding strategy that builds on RoPE due to its relative embedding space, we propose using a ray-based representation that encodes the spatial origin and direction of each patch, instead of the index of patches, see

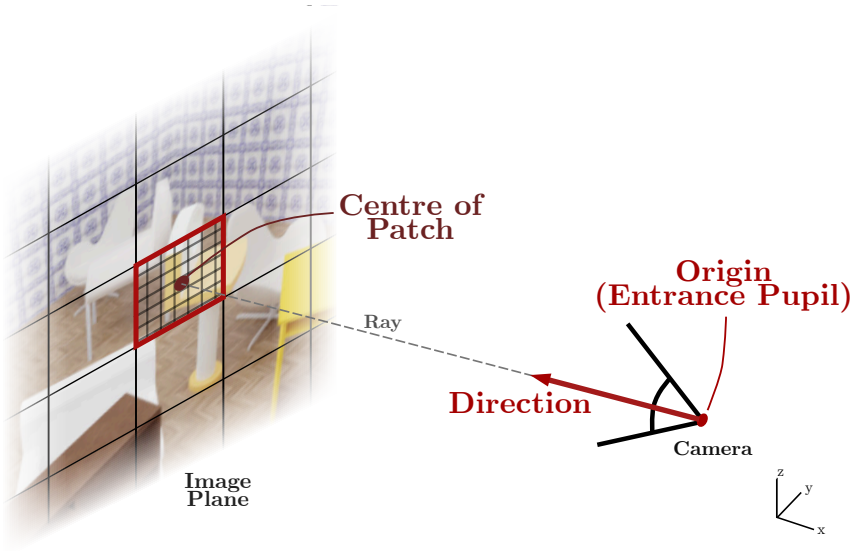

Figure 5.2 – Embedding ray-base information for patches in an image. This is the ray that is corresponds to the centre of a patch. It is parameterised as the using the camera origin and the direction of the ray.

Figure 5.2 for a graphical illustration. This perspective-centric formulation has several advantages, particularly when dealing with heterogeneous camera arrays where each view may observe the scene from a different pose or with a different field of view. This idea is consistent with prior work LVSM [60], though our approach differs in how the ray geometry is embedded within the network.

In addition to ray-based positional encoding, we explicitly embed the modality class associated with each input patch. This allows the transformer to distinguish between RGB and thermal inputs and to learn modality-specific patterns while still enabling cross-modal interactions. Together, the ray and modality embeddings form the foundation for consistent spatial reasoning across diverse inputs, facilitating accurate cross-view synthesis and depth estimation.

Rotary Embedding Rays

To embed ray-based positional information into the transformer architecture, we extend the concept of RoPE [97], originally designed for encoding 1D sequences in natural language processing. RoPE encodes positional information through rota-

tions, which preserve relative distances and directional relationships, making it an attractive foundation for injecting geometric information. Its original formulation was designed for one-dimensional sequences and must be extended to handle the higher-dimensional inputs required for ray-based vision.

We draw inspiration from several recent works that have extended RoPE beyond 1D. For instance, CroCoV2 [116] 2D variant to handle spatial information in image patches, while more recent efforts [111, 75] have proposed 3D extensions to incorporate temporal dimensions for video understanding. Building on this progression, we generalise to arbitrary n-dimensional ray data, which we refer to as RoRE.

RoRE allows us to embed high-dimensional inputs such as rays, which are parameterised by both position and direction in 3D space. A visual comparison of 1D, 2D, and our proposed RoRE embeddings is shown in Figure 5.3. Unlike traditional positional encodings that are added to the input, RoPE applies multiplicative transformations through rotations, maintaining spatial structure even after multiple layers of attention.

An interesting property of higher-dimensional RoPE is that it introduces structured constraints into the feature space. Increasing the dimensionality leads to fragmentation of the feature space, as more positional dimensions reduce the number of feature channels in which the positional information is embedded. While there is likely a practical upper bound on the dimensionality beyond which performance degrades, we did not observe such issues within our embedding configuration. Alternative approaches—such as learnt positional embeddings or coordinate MLPs—may offer complementary trade-offs without fragmenting the channel space, but a comprehensive comparison of these techniques is left for future work.

Ray Representation

A key design decision in our method is how to represent light rays in a form suitable for embedding within the transformer architecture. Two commonly used formulations in the literature are Plücker coordinates and the plenoptic function. While Plücker

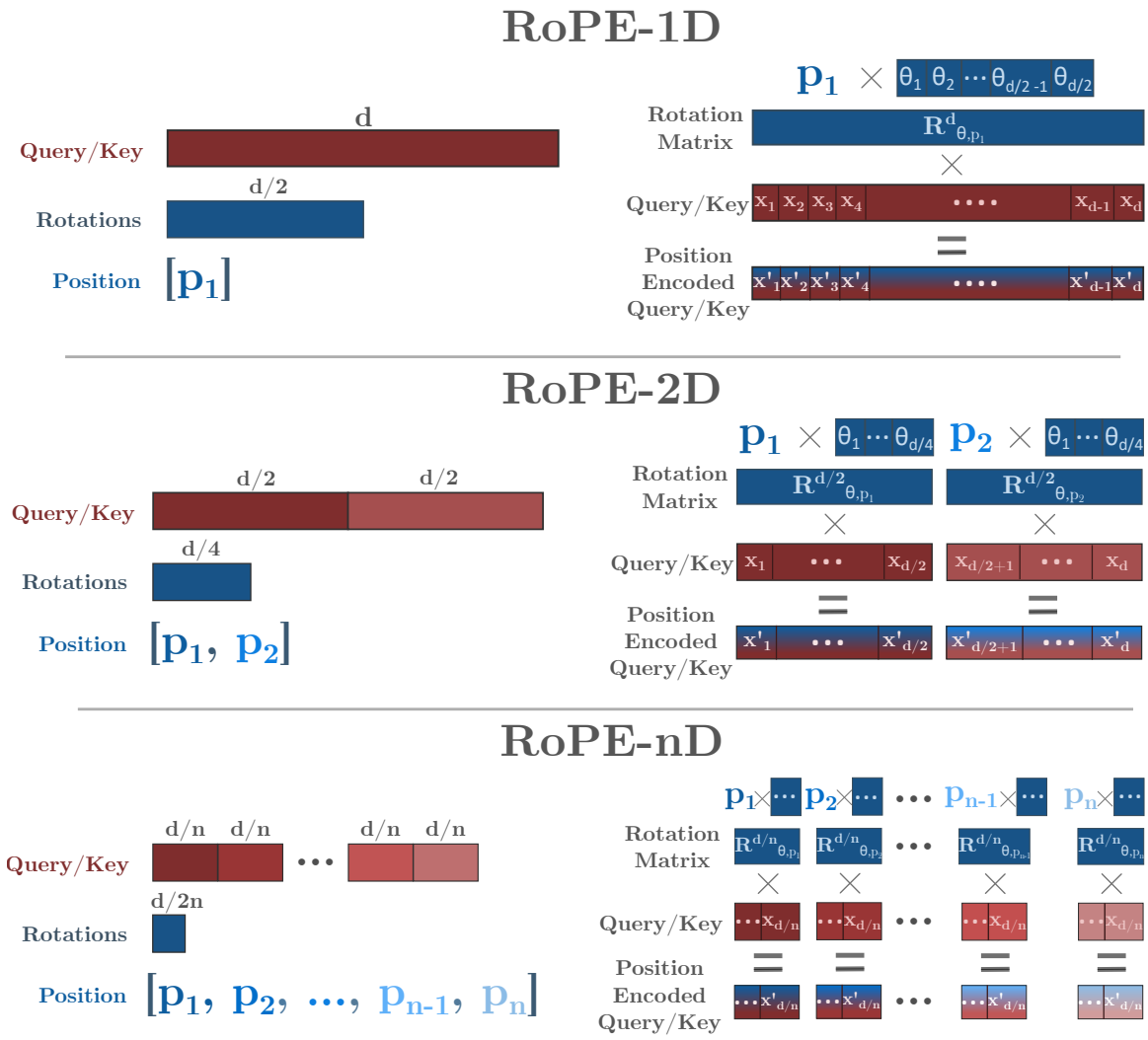

Figure 5.3 – Different dimensionality RoPE embedding. The RoPE-1D is the standard process [97] for 1D positional information used extensively in natural language processing. The process involves taking the query/key vector of size d and applying a rotation to this vector which are proportional to the value ofp_1. The RoPE-2D is an extension of the 1D case, for when 2D positional information is being embedded, by spiting the vector being encoded into two sub vectors. This is used in embedding patch location from images [112]. In this work to embed rays we require RoPE to embed a higher dimensionality, in our case 6. In this figure we show the n-dimensional case, which we refer to as RoPE-nD here. This allows us to embed additional positional dimensions of arbitrary size. However it should become clear that as we increase this dimensionality we fragment the latent space. This places additional constraints on the network, which we hypothesis is an undesirable side effect of increasing positional dimensionality.

coordinates provide a compact and algebraically elegant way to represent rays in projective space, they are less straightforward to work with when estimating depth, particularly in the context of feedforward learning systems.

We initially experimented with a local Plücker-based ray representation, similar to the approach used in LVSM [60], where scene geometry is inferred by computing ray intersections in a global coordinate system. However, this formulation introduces significant complexity when it comes to estimating per-pixel or per-patch depth, as it requires reasoning over geometric consistency in 3D space and computing local ray-frame intersections for every point. This added geometric burden proved difficult to integrate efficiently within the transformer framework.

Instead, we adopt a simpler and more flexible 6D ray formulation, which represents each ray as a combination of a point on the ray and its direction in 3D space. This representation is well-aligned with our RoRE embedding strategy and proved more effective in practice. In particular, we observed improved performance for depth estimation, with comparable results for photometric reconstruction.

The ray corresponding to each input patch is computed at the centre of the patch, and is represented as:

$$R^{patch} = [\mathbf{t}, \mathbf{d}] \tag{5.1}$$

where R^{patch} is the ray parameterisation for a patch, $\mathbf{t} = [t_x, t_y, t_z]$ is the 3D position of the camera entrance pupil in world coordinates, and $\mathbf{d} = [d_x, d_y, d_z]$ is the corresponding unit direction vector of the ray passing through the patch centre. These vectors are passed as input to the RoRE embedding described previously.

For the remainder of this chapter, when we refer to rays, we specifically mean this ray formulation for a specific path, unless otherwise stated.

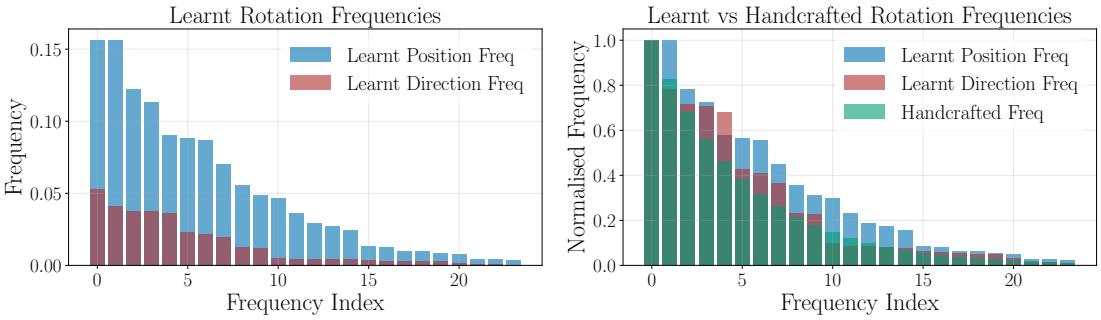

Figure 5.4 – Comparison of learned vs handcrafted frequency. Left compares the learned frequency for the position and direction dimension. It shows the magnitude of rotation that has been learned is larger for position than it is for direction. Right is comparing the normalised position and direction frequencies to the standard handcrafted frequency from (5.5). While similar the learned frequencies differ from the handcrafted ones.

Embedding Dissimilar Information

When embedding dissimilar types of information, such as ray origin and direction, there are no guarantees that the inputs will share the same spatial scale, nor that they should use similar frequencies in their sinusoidal encoding. In fact, in our setting, we know these elements operate at different scales: position values (ray origins) are typically much larger in magnitude than unit-normalised direction vectors. To account for this mismatch, we introduce two modifications to the standard RoPE formulation, both of which involve learning additional parameters that modulate the frequency and scale of the positional encoding.

Preliminaries

Su et al. [97] introduced RoPE which is designed to perform rotations of a d dimensional vector $\boldsymbol{x}$ as a way to embed relative positional information into a new vector $x_{rotated}$. Mathematically this is:

$$\boldsymbol{x}_{rotated} = f_{RoPE}(\boldsymbol{x}, m), \tag{5.2}$$

$$f_{RoPE}(\boldsymbol{x}, m) := \boldsymbol{R}^n_m x, \tag{5.3}$$

where $\boldsymbol{R}_m^n$ is an n dimensional rotation matrix constructed by multiple $2D$ rotation matrices:

$$R_m^n = \text{diag}\left[R^{2d}(m\theta_1), R^{2d}(m\theta_2),\ ...,\ R^{2d}(m\theta_{n/2})\right]_{d\times d}. \tag{5.4}$$

Here $R^{2d}(\theta)$ is the $SO(2)$ rotation matrix with angle θ. The θ is predefined based on the following:

$$\theta_i = 1000^{-2(i-1)/d}, \tag{5.5}$$

for $i \in [1, 2, ..., d/2]$. This provides decay of the rotation frequencies [106]. For further details and formal explanation see Su et al. [97].

While this was originally applied to one dimensional positional information for language models, it has also been applied to two dimensional positional embedding of pixel indices [116, 112, 69]. To do this the vector is split into two $\boldsymbol{x} = [\boldsymbol{x}_{/2}^{(1)}, \boldsymbol{x}_{/2}^{(2)}]$ where one half is rotated according to some pixel index u and the other half is rotated based on some pixel index v. Giving:

$$\boldsymbol{x}_{rotated} = [f(\boldsymbol{x}_{/2}^{(1)}, u), f(\boldsymbol{x}_{/2}^{(2)}, v)]. \tag{5.6}$$

In our case we do not want to use pixel indices instead we want to encode rays.

Embedding Rays

In this work, we embed a single ray for each image patch. Specifically, we use the ray corresponding to the patch centre, computed as the average of the rays of all pixels within that patch. This ray is then used to embed the position of a given patch. Building from base RoPE, we now have higher dimensional positions, in our case the 6 that are required to represent the Plücker ray. This has 3 position (or moment in the case of Plücker coordinates) dimensions t and 3 direction dimensions d. A new strategy is needed to developed to embed this information. A straightforward extension is to break up the embedding further into 6 parts $\boldsymbol{x} = [\boldsymbol{x}_{/6}^{(1)}, ..., \boldsymbol{x}_{/6}^{(6)}]$, leading

to:

$$x_{rotated} = \left[f\left(\boldsymbol{x}_{/6}^{(1)}, t_x\right), f\left(\boldsymbol{x}_{/6}^{(2)}, t_y\right), f\left(\boldsymbol{x}_{/6}^{(3)}, t_z\right), f\left(\boldsymbol{x}_{/6}^{(4)}, d_x\right), f\left(\boldsymbol{x}_{/6}^{(5)}, d_y\right), f\left(\boldsymbol{x}_{/6}^{(6)}, d_z\right)\right]. \tag{5.7}$$

We note, the more dimensions being embedded the more fragmented the vector becomes, essentially putting additional constraints on the latent, see Figure 5.3 for a graphical illustration of this effect. The position and direction components differ fundamentally in magnitude and semantic meaning. The magnitude of the frequencies required for translation values is likely to be different to that of the direction vectors.

We propose another approach by replacing the standard handcrafted base frequencies in (5.5) with frequencies for each dimension, superimposing their contributions without fragmenting the embedding space. This allows the network to learn how position dimensions interact with different parts of the latent space. Our frequencies $\boldsymbol{\theta}_{p\times d/2}$ have size $p \times \frac{d}{2}$, where $p = 6$ represents the ray dimensions and d is the query/key token dimension.

The final rotation around a given plane in the d dimensional space is the superposition of the of all the learned $\boldsymbol{\theta}$ values scaled by their respective ray-position values:

$$R_{RoRE}^{2d}(\mathbf{P}, \boldsymbol{\theta}_i) = R^{2d}\left(\sum_p \left(\mathbf{P}_p \cdot \boldsymbol{\theta}_{i,p}\right)\right), \tag{5.8}$$

where $\boldsymbol{P}_p$ is the position value vector $[t_x, t_y, t_z, d_x, d_y, d_z]$ containing the position values for a patch, $\boldsymbol{\theta}_i$ is the learned frequencies across all position dimensions for a given rotation plane $i \in [1, 2, ..., d/2]$. Our RoRE formulation then becomes:

$$\boldsymbol{x}_{rotated} = f_{RoRE}\left(\boldsymbol{x}, \boldsymbol{P}, \boldsymbol{\theta}\right), \tag{5.9}$$

$$f_{RoRE}\left(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{P}\right) := \boldsymbol{R}_p^n \boldsymbol{x}, \tag{5.10}$$

where

$$\boldsymbol{R}_p^n = \text{diag}[R_{RoRE}^{2d}(\mathbf{P}, \boldsymbol{\theta}_1), R_{RoRE}^{2d}(\mathbf{P}, \boldsymbol{\theta}_2), ...R_{RoRE}^{2d}(\mathbf{P}, \boldsymbol{\theta}_{d/2})]\boldsymbol{x}. \tag{5.11}$$

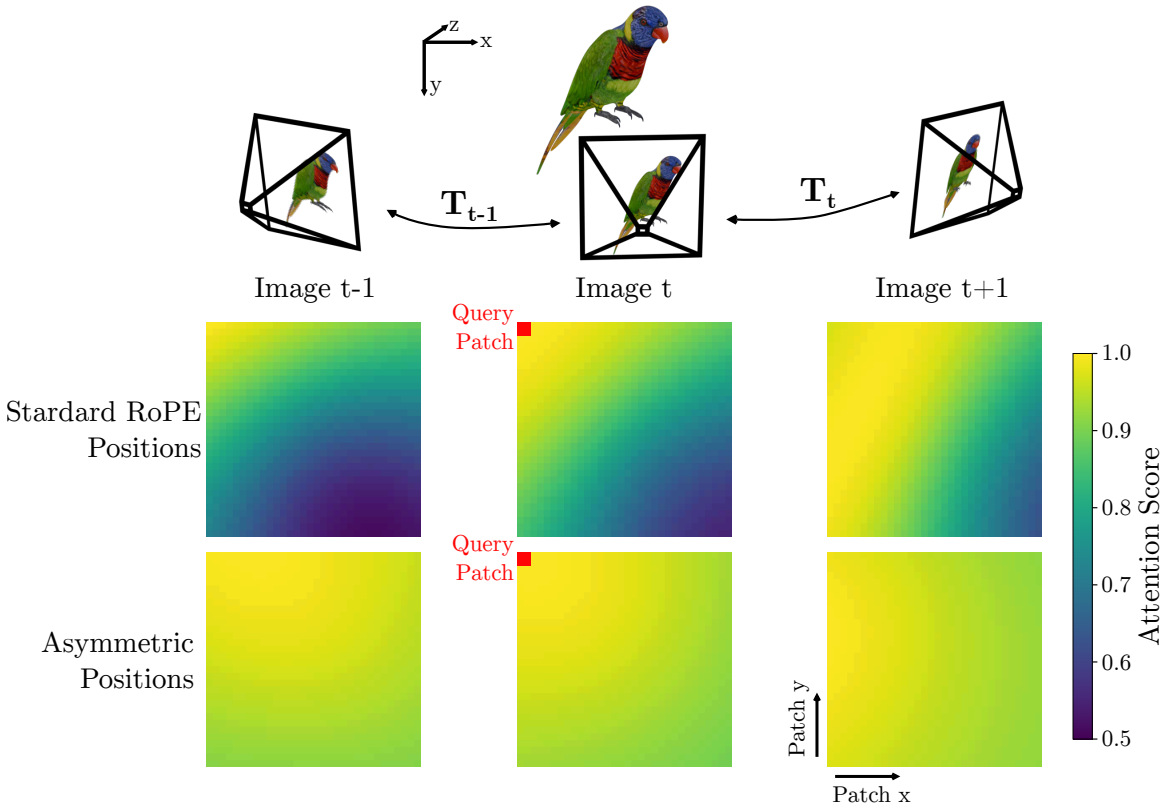

Figure 5.5 – Attention Comparison. Attention between frames at different positions using the Plücker parametrisation. The attention score between a query patch, identified in red, and all other patches is shown. Unit query and key vectors are used for this demonstration. The standard RoPE position values bias attention to rays near the query ray. This is problematic because geometric correspondences need not be spatially local. The asymmetric approach removes this bias providing a more uniform attention across possible position values.

The learned $\boldsymbol{\theta}$ parameter is randomly initialised using uniform initialisation between 0 to 0.5. It is left to future research to look into alternative initialisation strategies and how this effects performance.

Fig. 5.4 shows the learned rotation frequencies. It is encouraging that the model discovers a decay structure similar to the handcrafted schedule, despite being trained without any explicit constraints. This behaviour aligns with the established research: representing ray geometry requires a spectrum of frequencies, with higher-frequency components capturing fine-grained variations and lower-frequency components capturing broader spatial trends. The resulting learned decay therefore mirrors the in-

tended multi-scale behaviour of classical positional embedding, providing evidence that the learnt embedding parameters can autonomously recover a meaningful and interpretable frequency structure.

There are also clear differences between the position (moment) and direction dimensions, both in frequency decay and scale. This is expected, as the two quantities encode different geometric information. Positional components span a broader normalised range $[0, 1]$, whereas direction vectors vary more subtly due to camera motion constraints and overlapping fields of view. Consequently, the model allocates higher effective frequencies to direction channels and lower ones to position channels. The slightly sharper decay for direction likely reflects the finer rotational relationships between neighbouring rays.

A key benefit of this method is it removes the manual hyper-parameter selection process that is required for the handcraft method. We note that the ablation study (Tab. 5.4) shows very similar performance between the method outlined in (5.7) and the learnt method in (5.9). Due to the benefits of the learning-based method outlined above we use this method for our proposed approach.

5.3.2 Asymmetric Rotations

Standard RoPE, originally developed for NLP, is designed to emphasise local interactions by causing attention magnitudes to decay as the positional distance between tokens increases [97]. While beneficial for sequence modelling, this behaviour is undesirable in 3D vision, where rays that are far apart in image space may still hold important geometric relationships. To remove this distance-dependent attenuation, we extend an approach taken in VRoPE [75]. We perform a shifted negative counterpart of each positional component, ensuring that encoded magnitudes remain consistent across the ray domain. This modification preserves RoPE's rotational properties while preventing the unintended decay in attention, making the embedding better suited to ray-based scene representation. This can be expressed as:

$$\mathbf{V}^{patch} = \begin{pmatrix} \mathbf{t}^{+} \\ \mathbf{t}^{-} \\ \mathbf{d}^{+} \\ \mathbf{d}^{-} \end{pmatrix} = \begin{pmatrix} \mathbf{t} \\ -\mathbf{t} \\ \mathbf{d} \\ -\mathbf{d} \end{pmatrix} + \begin{pmatrix} 0 \\ b_{shift} \\ 0 \\ b_{shift} \end{pmatrix}, \tag{5.12}$$

where $\mathbf{P}$ is the position vector for a given patch, with $\boldsymbol{t}$ and $\boldsymbol{d}$ being the translation and direction components of the ray respectively. The b_{shift} is equal to 1 in our case, as the position $\boldsymbol{t}$ and direction $\boldsymbol{d}$ values are normalised to have a maximum value of 1. In practice in the $\boldsymbol{\theta}_{p\times d}$, the p dimension is actually twice the size of the original position vector.

Fig. 5.5 shows a visual comparison of embedding rays with and without the asymmetric positioning. For this comparison we take unit vectors for query and key tokens and compute the attention score between them after the rotary ray embedding has been applied. This is calculated for 3 images with different poses, with the attention scores being normalised. Without the asymmetric positioning the attention score is not uniform across the patches meaning it is biased toward rays near the query ray. While the asymmetric positions provide a much closer to uniform attention across the frames.

5.3.3 Architecture

The proposed model is based on the transformer architecture that has emerged as a highly effective structure used in many recent works [112, 60, 109]. As seen in Figure 5.6 our architecture is made up of an encoder and decoder. The encoder shown in green in the figure, is a transformer that performs self attention across all input patches from all cameras and modalities. The encoder performs patch embedding, this embedding can be performed in a number of ways from a linear layer [60], to a convolutional layer [112], or even a full transformer [109]. This patch embedding is modality specific, as in each modality has it own specific embedder. The Ray

Direction box calculates the ray that represents the centre of each patch, that is required for RoRE.

The decoder starts with a set of query patches that are a learnt patch embedding based of the modality of the patch that is being rendered. These patches are embedded through RoRE to represent a specific patch of the rendered output from a given ray. Cross-attention is performed between these patches and the output scene encoding from the encoder. The decoder has a head specific for each of the output renders. There are a few common choices for these heads, either a simple linear layer [60] or a dense prediction transformer (DPT) head [89]. For our work we use DPT heads for all modalities.

5.3.4 Masked Inputs

Masked input strategies have proven effective in recent vision research, particularly in encouraging models to develop generalisable and semantically rich internal representations [49]. Inspired by this, we extend the idea further by applying both geometric and modality masking within our framework. This allows the model to reason over missing spatial regions and unseen modalities simultaneously, promoting a deeper understanding of scene structure and cross-modal relationships.

We conceptualise our approach as a form of masked plenoptic autoencoding, where input patches from multiple viewpoints and modalities are selectively masked during training. The network is then tasked with reconstructing the missing information, effectively learning the underlying plenoptic function that describes the scene across views and sensor types. This formulation enables the model to interpolate spatially and semantically across both viewpoints and modalities.

Due to the nature of ViT, which operate on discrete tokens which represent patches rather than continuous rays, the model does not reason over individual rays directly. Instead, each token represents a bundle of rays corresponding to the image patch being encoded or rendered. Thus, masking a patch removes an entire bundle of directional observations from a given viewpoint.

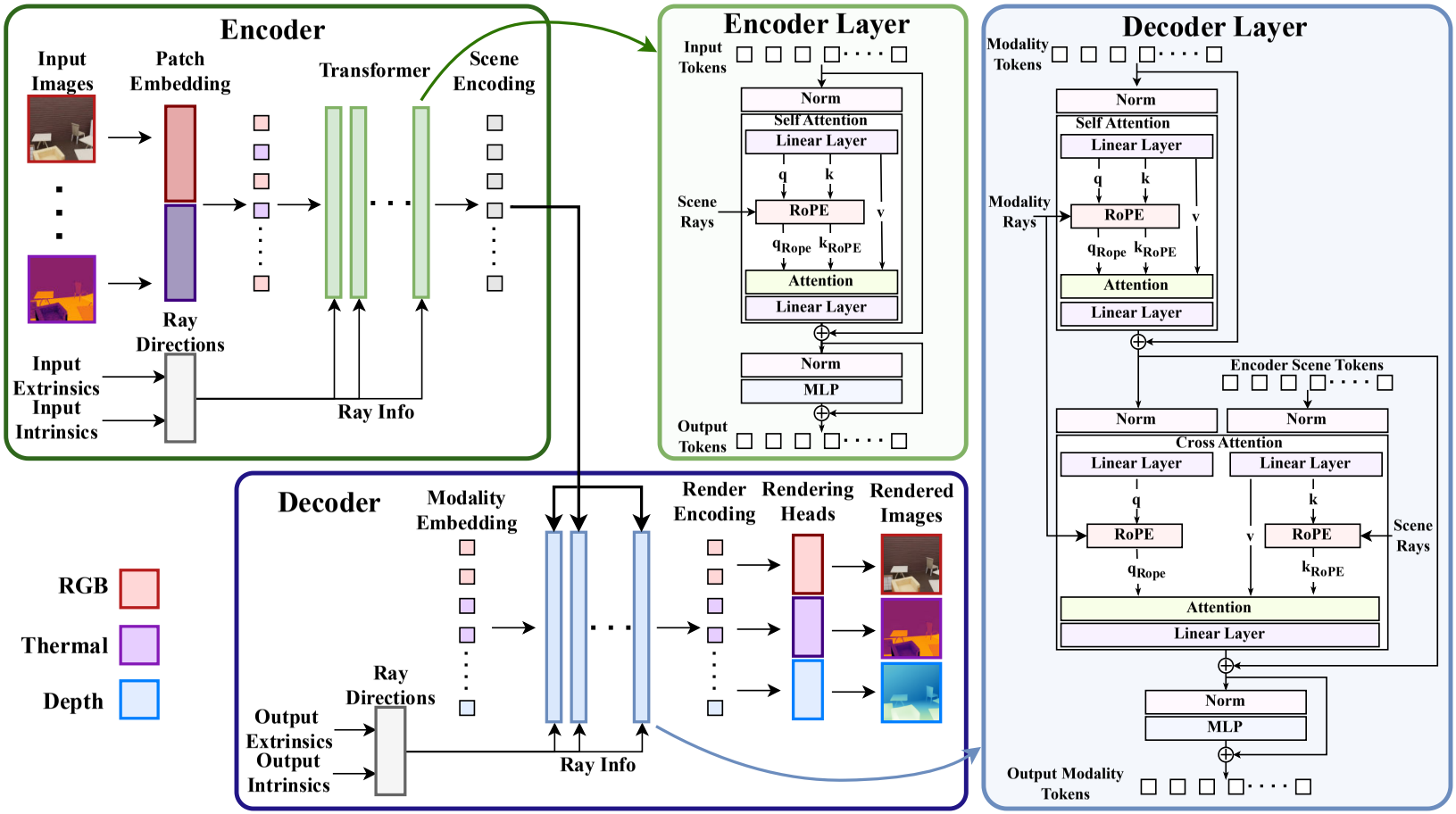

Figure 5.6 – Architecture diagram. The architecture has two stages, an encoder where images are patchified and encoded into a multimodal scene encoding. This encoding can then be used by the decoder to render novel views. This architecture makes use of our ray based RoPE to enable the embedding of camera intrinsics and extrinsics. This embedding is used in both the Encoder and Decoder layers. Otherwise these encoder and decoder layers are conventional where the encoder performs only self attention between its input patches, the decoder performs alternating self attention between the rendering patches and cross attention between the query rendering patches and the input images. Depending on outputs their respective modalities are passed through their corresponding modality head.

Importantly, we do not impose explicit constraints or losses to enforce cross-modal alignment. Rather, we rely on the reconstruction objective to implicitly encourage the network to learn correspondences across modalities. When a modality or viewpoint is masked, the model must infer the missing content using information from the remaining views and sensor types, leading to emergent cross-modal understanding.

5.3.5 Training Strategy

The training procedure is designed to encourage the model to develop generalisable representations across diverse viewpoints and sensor types. Each training sample consists of a fixed number of context (input) images and a fixed number of target (output) views, both sampled randomly from the same scene. For each target view, the model is tasked with rendering all available modalities, such as RGB and thermal, as well as predicting a corresponding depth map. This setup promotes consistency across both spatial viewpoints and sensory channels.

To improve robustness and encourage modality-invariant learning, the context images are randomly drawn from different combinations of modalities. During training, the samples may be RGB-only, thermal-only, or mixed RGB and thermal inputs. This randomised sampling strategy ensures the network is regularly exposed to asymmetric input conditions, fostering the emergence of unified cross-modal representations and improving its ability to reason jointly across heterogeneous inputs.

The spatial configuration of the input views is also varied, with different levels of viewpoint overlap. This enables the model to learn from overlapping regions where both modalities are present and to propagate this knowledge into regions where only one modality is available. As a result, the model learns to infer correspondences across modalities based on geometric cues and learnt correspondences.

5.3.6 Loss Functions

To train the network, we employ a combination of photometric and geometric supervision. Specifically, the total loss function consists of two appearance-based losses and one depth-based loss:

$$\mathcal{L} = \lambda_{\text{mse}}\mathcal{L}_{\text{mse}} + \lambda_{\text{lpips}}\mathcal{L}_{\text{lpips}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}}. \tag{5.13}$$

where $\mathcal{L}_{\text{mse}}$ is the MSE loss between predicted and ground-truth RGB or thermal reconstructions, and $\mathcal{L}_{\text{lpips}}$ is the LPIPS, which captures higher-level structural and semantic differences between the reconstructed and reference images. Both of these losses are detailed in Equation 2.19 and Equation 2.23, respectively.

To enforce geometric consistency, we also include a depth loss, $\mathcal{L}_{\text{depth}}$, defined as:

$$\mathcal{L}_{\text{depth}} = \frac{1}{N}\sum_{i=1}^{N} |d_i - \hat{d}_i| + |\nabla d_i - \nabla \hat{d}_i|, \tag{5.14}$$

where d_i and $\hat{d}_i$ represent the ground-truth and predicted depths at pixel i, and ∇ denotes the spatial gradient operator. The loss combines both absolute depth error and gradient-based smoothness, a formulation commonly used in monocular depth estimation encourage accurate relative depth. This loss works well for bounded scene e.g. indoors. If the method was being applied in scenes with large depth values e.g. outside, there are other losses that could be applied which are resilient to the reduced accuracy associated with larger depths.

The scalar weights λ_{mse}, λ_{lpips}, and λ_{depth} balance the contributions of the respective terms. This multi-term loss encourages the model to produce reconstructions that are both photometrically accurate and geometrically consistent, which is critical for effective multi-view, multi-modal synthesis.

5.4 Results

5.4.1 Datasets

This work makes use of a combination of real-world and synthetic datasets to evaluate performance across single-modality and multi-modality scenarios. The selected datasets enable both comparison to existing state-of-the-art methods and controlled analysis of cross-modal fusion.

To assess performance on real-world RGB-only data, we utilise two large-scale datasets: RealEstate10K [130] and DL3DV-10K [74]. RealEstate10K consists of 10,000 video sequences sourced from online real-estate listings, featuring a wide range of indoor and outdoor environments. The dataset provides camera parameters and multi-view RGB imagery, making it a widely adopted benchmark for novel view synthesis tasks. DL3DV-10K is a similar dataset comprising 10,000 diverse scenes, with a broader range of viewpoints and environmental variability. These datasets are employed exclusively for RGB-based experiments. RealEstate10K is used for the quantitative comparison in Table 5.1 and for qualitative analysis in Figure 5.10, while DL3DV-10K contributes additional qualitative examples in the same figure.

For evaluating multi-modal performance, we use a custom Blender-based simulated dataset that includes aligned RGB, thermal, depth and normal modalities. The dataset was generated using BlenderProc [26], with additional modifications to simulate thermal imaging. In total, the dataset contains 3915 training scenes and 100 test scenes. Each scene is rendered from 40 distinct camera poses, capturing all modalities, resulting in 120 images per scene (40 per modality). Complete intrinsic and extrinsic calibration is available for every view.

Scenes are procedurally generated to ensure diversity. Each room is assigned a random layout and populated with 3D assets from the ShapeNet dataset [10]. Surface appearances are randomly textured using CC0-licensed materials [1] for floors, walls, and objects, resulting in realistic and diverse visual variation across the dataset.

[1] `https://cc0-textures.com/`

Simulated data is employed for several practical and methodological reasons. First, it removes the burden of manual annotation while providing precise ground truth for depth and geometry, which is essential for evaluating cross-modal learning. Second, the procedural generation process enables the creation of large-scale, diverse datasets without the need for extensive physical data collection. While simulation does not perfectly replicate all real-world effects, the dataset serves as an effective and controlled environment for validating the proposed framework's ability to learn multi-modal correspondences and geometry. This dataset is used throughout all multi-modal experiments presented in this chapter.

5.4.2 Baselines

For validating our relative ray-based embedding method we compare to methods of novel view synthesis that perform positional embedding in different ways with different information: LVSM [60] which is an absolute positional embedding only method. GTA [81] which using both ray based absolute embedding and their relative encoding of camera extrinsics; and finally concurrent work PRoPE [71], which embeds their own camera base absolute embedding and a modified GTA relative embedding that also embe camera intrinsics using a projection matrix. All methods including ours use the exact same model architecture keeping all parameters the same except for varying the embedding method. These baselines are used for the RGB-only results.

We are unaware of any existing alternative feedforward multi-modal models, as such we do not perform direct comparisons to existing methods. For our multi-modal approach instead we show different operating modes to characterise its performance. We evaluate reconstruction quality using PSNR, SSIM, and LPIPS for both RGB and thermal outputs, noting that perceptual metrics such as LPIPS were not originally designed for thermal imagery and therefore cannot be directly compared to their RGB counterparts.

5.4.3 Implementation Details

Two distinct model configurations are used throughout our experiments to accommodate the different evaluation settings: one for benchmarking against the state-of-the-art LVSM [60], and another for the proposed multi-camera, multi-modal framework.

RGB-Only Comparison Configuration: For experiments involving direct comparison to LVSM, we adopt the standard configuration used in the original implementation. Specifically, we replicate their use of Plücker ray representations and maintain architectural parity to ensure a fair evaluation. All hyperparameters and training protocols follow those reported in the LVSM public code repository. This setup is used exclusively in the experiments presented in Table 5.1.

Multimodal Configuration: For all other experiments, including the evaluation of our proposed approach on multi-modal, multi-camera inputs, we use a dedicated configuration tailored for generalised scene understanding. The scene encoder is based on a ViT-Large backbone, while the decoder uses a ViT-Base architecture. Ray geometry is embedded using the plenoptic representation, as described in Section 5.3.1, patches are embedded using a single convolutional layer and predictions are generated using DPT heads [88, 89] for all output modalities.

A full overview of the hyperparameters and architectural settings used in the multi-modal configuration is provided in Table A.1.

5.4.4 RoRE Embedding

To validate the impact of our embedding, we replace the original pose encoding in LVSM with our proposed Relative Ray-based embedding and assess both convergence behaviour and qualitative reconstruction performance. The convergence is plotted in Figure 5.7. It is observed adding that adding our RoRE embedding improves the PSNR performance.

Tab. 5.1 reports novel view synthesis results on RealEstate10K and DL3DV. All models are trained on RealEstate10K, with evaluation on the same dataset reflecting in-domain performance, and DL3DV providing an unseen but closely related test set. Across both datasets, the methods achieve broadly comparable results: PRoPE performs slightly better on the training domain, while LVSM is marginally lower. While these results do not highlight a clear advantage for our method, they establish that RoRE remains competitive on standard benchmarks, with its benefits becoming more evident in settings that require greater generalisation, as demonstrated in subsequent experiments.

Varying Intrinsics. We evaluate robustness to changes in camera intrinsics by varying the focal length of target and query images through cropping, with randomly chosen magnification of up to 3. This experiment was conducted without retraining the models. As shown in Tab. 5.2 and Fig. 5.8, methods with stronger representation constraints, such as GTA and PRoPE, fail to adapt. In contrast, LVSM and our RoRE handle these variations effectively, with RoRE consistently outperforming LVSM. While PRoPE can address this case with additional training, our results highlight the inherent advantage of ray-based embeddings, which are naturally invariant to changes in intrinsics.

Distorted and Fisheye Inputs. We next test robustness to non-perspective inputs, using (i) perspective images from RealEstate10K with added barrel distortion and (ii) native fisheye images from FIORD as shown in Tab. 5.3 and Fig. 5.9. Distorted inputs are paired with perspective queries, and all evaluations are conducted without retraining.

Our method demonstrates consistently stronger generalisation than competing approaches, outperforming LVSM by over 1 dB in PSNR, while GTA and PRoPE fail due to the absence of explicit ray-direction encoding. The fisheye case is particularly relevant, since rectification would reduce field of view; RoRE can handle these inputs directly, capturing the global distortion, though some local inaccuracies remain.

Discussion. These results reflect the differing representational biases of the methods. PRoPE uses a constrained, camera-specific formulation based on projection-

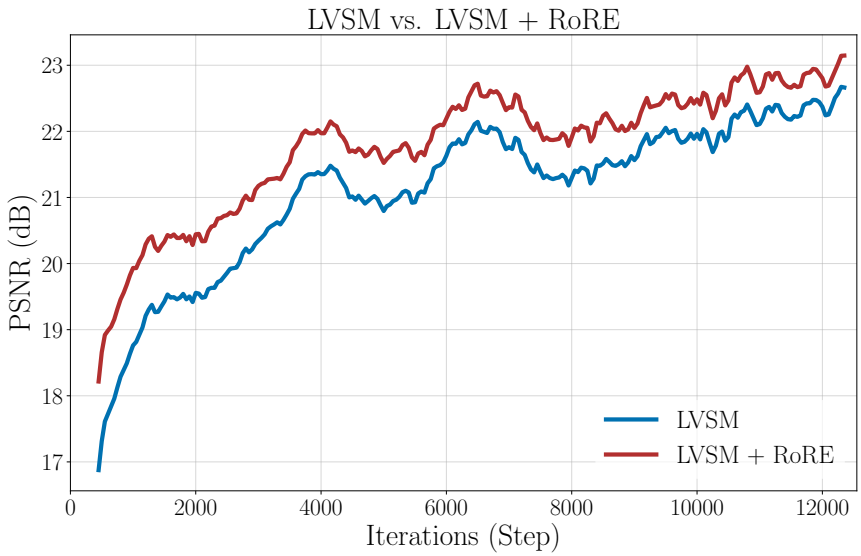

Figure 5.7 – Comparing PSNR performance during training for LVSM and LVSM+RoRE. This supports that adding the RoRE embedding to LVSM improves reconstruction performance.

Table 5.1 – Novel view synthesis results. Results from RealEstate10K (training domain) and DL3DV (unseen but similar domain). All methods perform comparably, with LVSM performing marginally worse and PRoPE performing marginally better. Method marked with † represents concurrent work.

Method	RealEstate10k			DL3DV			Iteration Time
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	Seconds
LVSM	26.18	0.834	0.076	19.48	0.604	0.281	**1.287**
GTA	26.74	0.846	0.069	19.55	0.614	0.281	1.647
PRoPE†	**26.81**	**0.848**	**0.068**	19.68	**0.620**	**0.278**	1.454
RoRE (ours)	26.65	0.845	0.070	**19.77**	0.619	0.279	1.326

matrix relative encodings, which aligns well with conventional perspective data such as RE10K. GTA imposes an even more restricted variant based in the extrinsics. RoRE, by contrast, encodes full rays and learns multi-dimensional frequency interactions, resulting in a more expressive and geometry-agnostic representation. This flexibility leads to substantially stronger generalisation under intrinsics changes, distortion, and fisheye inputs (Tables 5.2, 5.3). However, its less constrained embedding space can yield slightly lower performance on tightly scoped perspective datasets such as RE10K and DL3DV (Table 5.1).

Table 5.2 – Quantitative evaluation under varying focal lengths. Models are tested without retraining by cropping target and query images to simulate changes in camera intrinsics. Methods with stronger representation constraints (GTA and PRoPE) fail to adapt, while LVSM and RoRE remain robust. RoRE consistently outperforms LVSM, demonstrating the advantage of a relative ray-based embeddings for generalisation across intrinsics variations.

Method	RealEstate10k			DL3DV		
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
LVSM	21.95	0.744	0.219	19.86	0.653	0.349
GTA	14.81	0.523	0.459	14.47	0.469	0.564
PRoPE	14.71	0.516	0.486	14.28	0.454	0.617
RoRE (ours)	**22.66**	**0.770**	**0.211**	**20.31**	**0.678**	**0.335**

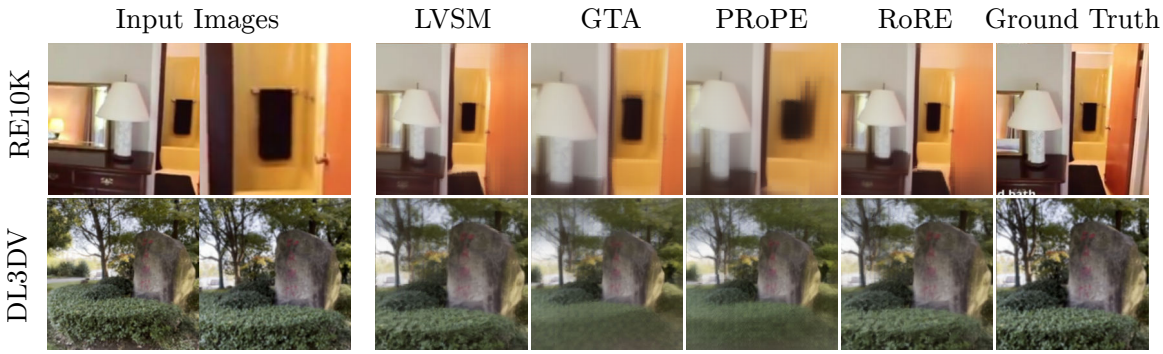

Figure 5.8 – Varying intrinsics in scene. When the camera intrinsics vary within a scene without any additional training, we see both GTA and PRoPE fail to interpret the new cameras. The authors of PRoPE show that with training PRoPE is capable, however RoRE natively understands this, without additional training.

Although we were unable to train either model to full convergence due to computational resource constraints, we observe that our 6D-Ray RoPE variant demonstrates faster convergence during the first 12,000 iterations. This suggests that our embedding enables the model to more effectively learn spatial relationships early in training, likely due to the relative embedding nature of RoPE. Given prior evidence of improved performance from similar embeddings in 2D applications, and based on these early-stage trends, we have strong reason to believe that with sufficient compute our method would also yield higher absolute performance upon full training.

In addition to the convergence analysis, we present qualitative results on masked inputs for the RealEstate10K and DL3DV-10K datasets using RGB-only to demonstrate is capabilities degraded inputs. These are shown in Figure 5.10. The examples shown

Table 5.3 – Quantitative evaluation on distorted and fisheye inputs. Barrel-distorted RealEstate10K images and native FIORD fisheye images are used as inputs without retraining. RoRE generalises robustly to both cases, while GTA and PRoPE fail due to the absence of explicit ray-direction encoding.

Method	Distorted RE10K			Fisheye FIORD		
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
LVSM	21.99	0.725	0.142	22.52	0.732	0.310
GTA	18.58	0.605	0.188	11.64	0.456	0.596
PRoPE	18.57	0.605	0.188	11.90	0.408	0.673
RoRE (ours)	**23.96**	**0.802**	**0.124**	**23.55**	**0.746**	**0.284**

Figure 5.9 – Qualitative results on distorted and fisheye inputs. RoRE preserves scene structure under both barrel-distorted perspective images (top) and native fisheye images (bottom), whereas competing methods produce severe artefacts or fail to reconstruct meaningful views.

here are typical of the training inputs where input images are partially masked to encourage cross-view completion. The reconstructions demonstrate high visual fidelity across input scenes, confirming the practical applicability of our embedding even in standard monocular RGB settings. These results further support the case that our ray-based encoding provides a general and robust representation for capturing camera pose and geometry in transformer-based view synthesis frameworks.

Ablation Study: We ablate different components of our method, shown in Tab. 5.4. Firstly, the learnt frequencies refers to the method outlined in Sec. 5.3.1. The method without it refers to the process outlined in Eqn. 5.7. Asymmetric refers to the method outlined in Sec. 5.3.2. Including asymmetric positioning provides a modest increase to performance. Using learned frequencies yields performance comparable to the hand-

Table 5.4 – Ablation study on RE10K. The study shows adding the relative embedding in any form improves the performance. The asymmetric positioning provides another modest improvement to performance. Applying the learnt frequency produces identical results but as stated provides a more general approach. RoRE performs similarly whether the additional absolute embedding is applied or not.

Absolute Emb.	Relative Emb.	Learnt Frequencies	Asymmetric	PSNR(↑)	SSIM(↑)	LPIPS(↓)
✓	✗	✗	✗	26.18	0.834	0.076
✓	✓	✗	✗	26.56	0.843	0.071
✓	✓	✗	✓	**26.65**	**0.845**	**0.070**
✓	✓	✓	✗	26.57	0.842	0.072
✗	✓	✓	✓	**26.65**	0.843	**0.070**
✓	✓	✓	✓	**26.65**	**0.845**	**0.070**

crafted schedule. However, we note that this formulation is a more general solution that removes the need for additional hand tune parameters and handcrafted elements, for this reason our proposed method utilises these learnt frequencies, as they do not impact performance or inference time.

Since the RPE is separate and complements the APE we can use any combination of the embedding methods. The study demonstrates that the performance boost comes from using the RPE compared to the APE. Including both the performance is roughly the same, this is an indication that the network relies on the most appropriate embedding information. Our relative embedding method could work without the absolute embedding, however we do include it in the other experiments as it does show a slightly higher SSIM score.

5.4.5 Multi-Modal Reconstruction

Having validated the effectiveness of our ray-based positional embedding on single modality inputs, we now evaluate the full multimodal network using RGB and thermal imagery. While our architecture differs slightly from the state-of-the-art LVSM model, it maintains a similar high-level design, adapted to handle multi-camera, multi-modal input directly within a unified framework.

Our method produces a single model capable of operating with three distinct input configurations: RGB-only, thermal-only, and combined RGB-thermal. This flexibil-

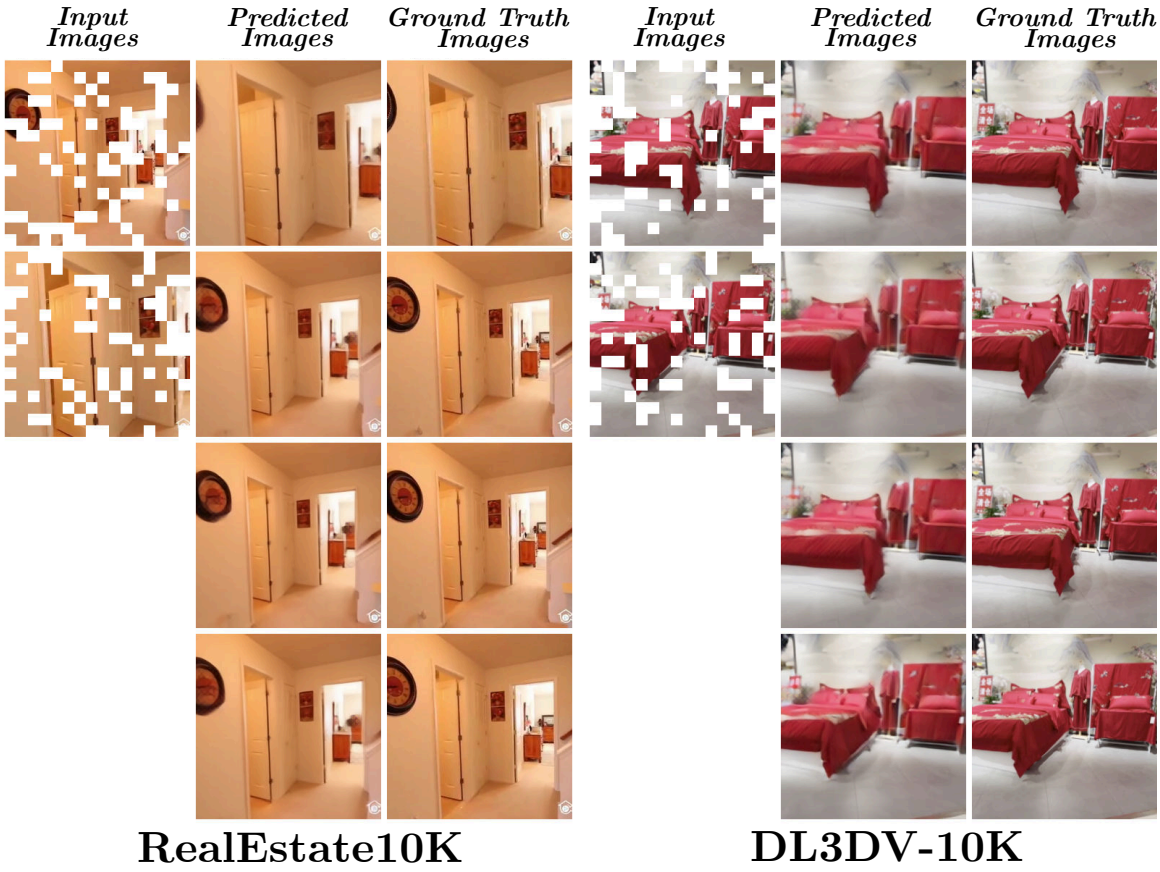

Figure 5.10 – Reconstruction results on the RealEstate10K and the DL3DV-10K Datasets. Here is a typical training sample where input images are masked and multiple frames are rendered from different view points. The rendering results show high quality reconstruction on both datasets.

ity enables deployment across a wide range of sensing scenarios without requiring separate models or retraining for each modality combination.

Quantitative and qualitative results are presented in Table 5.5 and Figure 5.11, respectively. The network demonstrates stable performance across all modalities, with depth estimates remaining consistent regardless of input type, demonstrating its ability to generalise across different sensing configurations. In particular, reconstructions from RGB-only and RGB+thermal inputs show slightly improved quality over thermal-only, which is expected given the higher information density in RGB images compared to thermal. For single-modality inputs, the network renders novel views using only the provided modality.

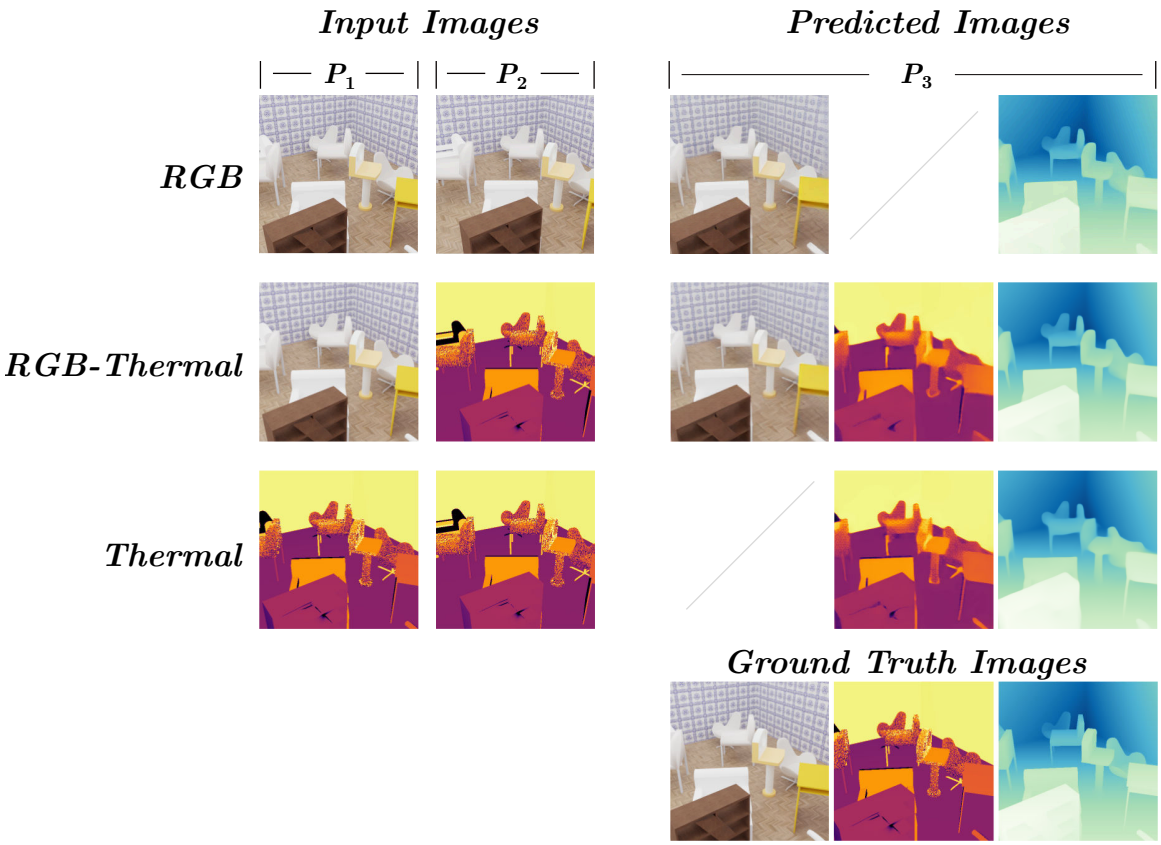

Figure 5.11 – Qualitative results from the multimodal network. Inputs are from two poses, P_1 and P_2, while rendering is from a third pose P_3. Showing three different modes of operation, RGB-only, both RGB and thermal, and thermal-only. This shows that the model is able to output high quality reconstructions in all three operating modes, importantly the depthmaps shown are consistent across the different input modalities.

As shown in Figure 5.11, the model produces high-fidelity reconstructions and depth maps in all three operating modes. The consistency of the depth outputs across input configurations indicates that the network has successfully learnt cross-modal correspondences and is capable of leveraging both shared and modality-specific features in a coherent spatial representation.

While we report LPIPS scores for thermal image reconstructions, it is important to acknowledge that the LPIPS metric was originally developed for evaluating perceptual similarity in RGB imagery. As such, its application to thermal images which lack colour channels and have significantly different texture and contrast characteristics

is not fully aligned with the metric's intended use. Consequently, LPIPS values for thermal images should not be interpreted or compared directly with those from RGB reconstructions. Nevertheless, we include these scores as they remain useful for assessing relative performance between thermal reconstructions across different experimental conditions.

To further assess the capabilities of the proposed multimodal network, we evaluate its performance across three distinct input configurations involving two cameras—one RGB and one thermal. Specifically, we explore: (i) partial overlap between the fields of view of the two modalities, (ii) no overlap between them, and (iii) rendering in regions completely outside the field of view of both cameras. These test cases are designed to investigate the model's ability to learn cross-modal correspondences, reason spatially beyond direct observation, and assess the limits of generalisation in the absence of visual information.

In the first case, where partial overlap exists between RGB and thermal inputs, the network is tasked with completing scene content that is only partially observed in one modality but fully captured in the other. Results for this condition are shown in Figure 5.12. We observe that the network successfully completes objects that are truncated in one view by using information from the other. For instance, in the RGB view, the green chair is only partially visible, but the predicted RGB output reconstructs it in full by borrowing geometric cues from the corresponding thermal input. Similarly, in the thermal prediction, the chair structure is completed using information from the RGB modality. When objects are observed by only a single modality, the geometry remains structurally plausible; however, photometric consistency degrades, for example, the red chair is reconstructed with incorrect colouration in the rendered RGB image. Despite these photometric inconsistencies, depth estimation remains stable and structurally coherent and accurate.

One point of interest is Figure 5.12 (b) where the floor has been filled with the brick texture observed in the input image, however it has been applied at the incorrect angle. The other point of interest is the transparency of the green chair. The chair is slightly transparent, however this transparency is more pronounced in the rendering.

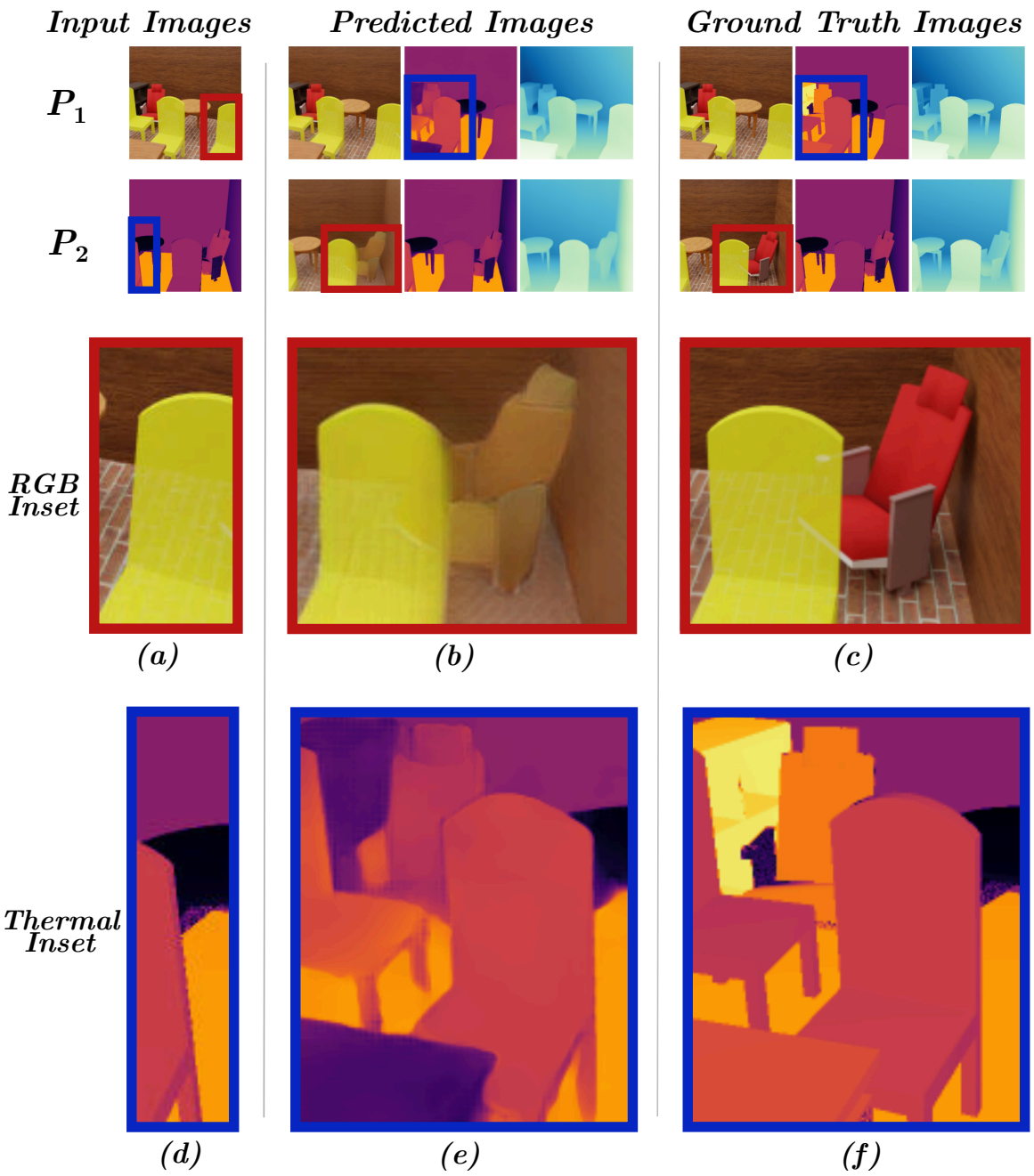

Figure 5.12 – Partial overlap of input images at two positions P_1 and P_2. (a) RGB Input with green chair is only partially in frame, (b) Predicted RGB shows a completed green chair using the shape from thermal, (c) RGB ground truth shows the chair is constructed well, however the red chair is the wrong colour. (d) Thermal input shows chair mostly out of frame, (e) thermal predicted image with a completed chair using information from RGB, (f) thermal ground truth shows the chair has been completed well, the dresser in the background has been predicted with plausible structure but incorrect temperature similar to the chair in RGB.

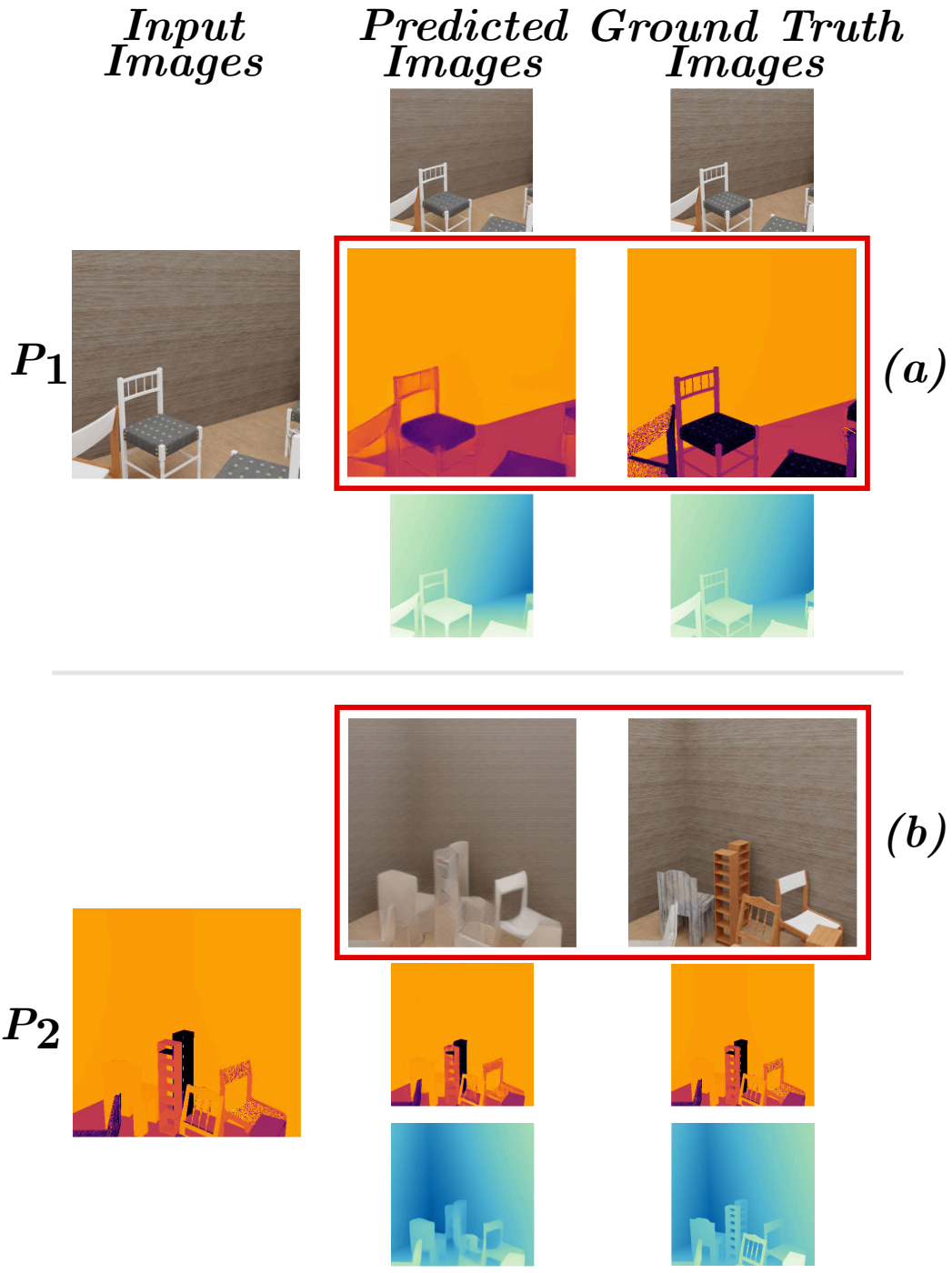

Figure 5.13 – No overlap between inputs, a RGB image at P_1 and thermal image at P_2. (a) shows the predicted thermal image from the view of P_1, due to no thermal information being provided here the photometrics are incorrect, however the structure is correct. The floor and walls have been correctly identified from the thermal input image. (b) shows the predicted RGB images from the view of P_2, the same situation is observed, the photometrics are generally wrong except the walls and floor.

The second scenario considers the more extreme case of no overlap between the input modalities, with RGB and thermal images captured from entirely separate viewpoints. The results are presented in Figure 5.13. Interestingly, the network still reconstructs plausible scene geometry. Large scale background elements such as walls and floors are inferred correctly, likely because these surfaces exhibit consistent appearance and layout throughout the scene. This suggests that the model has learnt to generalise spatial priors even across modality boundaries. As in the partial overlap case, photometric predictions are unreliable when the modality is not observed at the rendering viewpoint—for example, the predicted thermal image from an RGB-only viewpoint lacks accurate intensity vales. However, structural predictions remain valid, and the depth output continues to perform robustly by falling back to monocular depth cues.

Finally, we evaluate the out-of-scope case, where neither modality observes the rendered viewpoint. As shown in Figure 5.14, the model degrades under this condition. While photometric reconstructions become blurry and semantically nonsensical, the predictions still retain coarse structural elements such as the general colour and shape of walls and floors. However, fine-grained detail and texture are entirely absent, as expected in the complete absence of visual input. This behaviour indicates that while the model can interpolate between known views, it lacks the capacity to hallucinate plausible content beyond the visual hull defined by the input. This may or may not be desirable depending on application.

A note on the SSIM, it is observed that RGB consistently has a lower scores than the thermal, we postulate this is due to RGB having more complex texture to estimate compared to thermal.

5.4.6 Multi-Camera Reconstruction

To evaluate the scalability and generalisability of our multimodal transformer framework, we examine its performance under varying numbers and combinations of input images. Unlike previous models that assume a fixed number or modality of inputs, our architecture is designed to accept any number of camera views, comprised of any

Table 5.5 – Comparison of different input modalities. We see that this single model is able to accept a range of input modalities configurations without any adjustments.

Input Images	RGB			Thermal			Depth		
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	AbsRel(↓)	RMSE(↓)	δ_1(↑)
rgb-rgb	22.995	0.514	0.218	-	-	-	0.060	0.024	0.965
rgb-thermal	21.494	0.457	0.254	20.481	0.744	0.172	0.060	0.025	0.964
thermal-thermal	-	-	-	21.662	0.770	0.153	0.065	0.027	0.959

Table 5.6 – Different configurations of input images. Increasing the number of input frames provides the network with more information which leads to improved performance, across all metrics. Additionally the model is flexible to the modality configuration of the input images.

	RGB			Thermal			Depth			Encode Time	Decode Time
Number Images	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	AbsRel(↓)	RMSE(↓)	δ_1(↑)	ms	ms
1 (1-RGB)	19.738	0.396	0.301	-	-	-	0.084	0.033	0.929	35.413	35.513
1 (1-thermal)	-	-	-	18.610	0.700	0.215	0.100	0.036	0.909	34.312	34.902
2 (1-RGB, 1-thermal)	20.470	0.418	0.280	20.068	0.738	0.181	0.062	0.026	0.958	37.493	40.662
4 (2-RGB, 2-thermal)	21.751	0.463	0.246	21.008	0.758	0.161	0.053	0.022	0.966	47.489	44.182
8 (4-RGB, 4-thermal)	22.767	0.490	0.223	22.104	0.783	0.140	0.044	0.019	0.979	95.566	54.249
6 (1-RGB, 5-thermal)	20.927	0.431	0.269	22.082	0.783	0.143	0.051	0.022	0.971	71.211	49.486
6 (2-RGB, 4-thermal)	22.013	0.469	0.241	21.829	0.777	0.146	0.050	0.021	0.972	71.451	49.569
6 (3-RGB, 3-thermal)	22.409	0.480	0.231	21.580	0.771	0.151	0.049	0.021	0.975	70.531	49.441
6 (4-RGB, 2-thermal)	22.730	0.490	0.223	21.209	0.764	0.156	0.048	0.021	0.975	70.524	49.403
6 (5-RGB, 1-thermal)	22.916	0.497	0.218	20.534	0.746	0.168	0.049	0.021	0.975	71.594	49.488

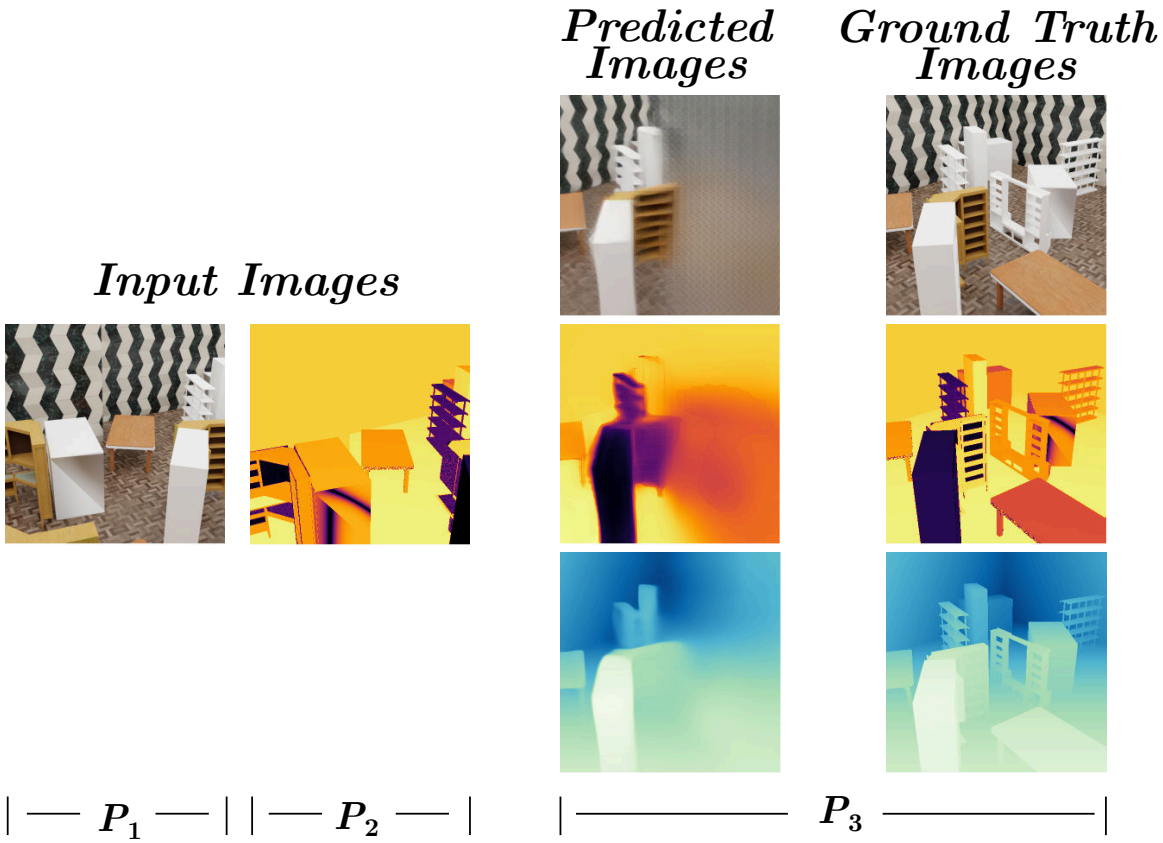

Figure 5.14 – Rendering out of the scope of the input images with RGB input at pose P_1 and Thermal input at pose P_2. The Predicted images at pose P_3 which is out of frame of the both input images shows a significant degradation and smoothing as it goes out of frame. The walls are partially guessed correctly.

mix of RGB and thermal modalities, only maximally constrained by available memory and compute at inference time.

We conduct two main experiments to explore this flexibility, with results summarised in Table 5.6. In the first experiment, we vary the number of input images, using an equal split of RGB and thermal modalities (e.g., 1 RGB and 1 thermal up to 4 RGB and 4 thermal images). In the second experiment, we fix the total number of input images to six and vary the modality ratio, altering the number of RGB versus thermal images to investigate how modality dominance affects reconstruction quality.

We observe that increasing the number of input images improves both the quality of rendered photometric views and the accuracy of depth estimates. This is consistent with expectations, more viewpoints provide the network with greater coverage of the

scene and improved geometric constraints. As the number of inputs increases, we see more complete and coherent reconstructions, with reduced artefacts and improved consistency in depth predictions.

When varying the modality composition while keeping the total number of images constant, the results follow understanding, increasing the number of thermal inputs improves the fidelity of thermal reconstructions but slightly degrades RGB predictions, and vice versa. This reflects the model's ability to use the additional information of a given modality. Interestingly, depth estimates remain relatively consistent across modality mixes, although slightly better performance is observed when RGB images dominate. This is likely due to the higher spatial resolution and texture information in RGB inputs, which provide stronger cues for depth inference, a trend also reflected in Table 5.5.

These results confirm the model's ability to flexibly integrate information across arbitrary camera configurations. This allows for investigation of how camera numbers and modality balance can influence performance when deployed in specific sensing environments. See Section A.2 for additional figures showing qualitative results of multi-camera inputs.

5.4.7 Masked Inputs

The use of masked input tokens during training enables the model to develop resilience to partial occlusions and missing data at inference time. By learning to reconstruct scenes from incomplete information, the network becomes inherently robust to scenarios where certain portions of the input imagery are unavailable or corrupted. This design is particularly valuable in real-world deployments, where sensors may be partially obscured by environmental factors such as dirt, water, or glare.

To evaluate this robustness, we progressively increase the percentage of masked input tokens during inference and observe the model's reconstruction and depth estimation performance. Quantitative metrics are reported in Table 5.7, and qualitative results

Table 5.7 – Masking multi-modal input images progressively from 0% to 90%. This table shows that the model is resilient to the loss of information. Importantly the depth prediction is still reliable even after a significant level of masking, compared to the 0% masking case.

Mask	RGB			Thermal			Depth		
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	AbsRel(↓)	RMSE(↓)	δ_1(↑)
0%	21.494	0.457	0.254	20.481	0.744	0.172	0.060	0.025	0.964
10%	20.875	0.435	0.269	20.021	0.734	0.181	0.063	0.025	0.962
30%	19.612	0.379	0.303	18.933	0.707	0.205	0.067	0.027	0.957
50%	18.287	0.312	0.350	17.755	0.674	0.239	0.076	0.031	0.945
70%	16.913	0.249	0.409	16.367	0.632	0.286	0.105	0.042	0.884
90%	14.902	0.192	0.496	14.183	0.560	0.366	0.207	0.082	0.537

are shown in Figure 5.15. As expected, performance degrades with increasing levels of masking; however, the degradation is gradual and surprisingly modest up to significant levels of occlusion.

At 10–50% masking, the model retains high-quality reconstructions and stable depth predictions. For example, even with 50% of the input patches masked, key scene elements, such as the black box on the bottom shelf remain accurately reconstructed. A more noticeable drop in performance occurs at 70% masking, where some finer scene details disappear from the reconstructions, and quantitative metrics show a clear reduction in image fidelity and depth accuracy. Nonetheless, the network manages to predict plausible global structure and generate depth maps that remain broadly consistent with the ground truth.

These findings highlight the model's strong capacity for geometric reasoning under partial observability. We believe its ability to perform scene completion using sparse visual input is a consequence of operating in ray or patch space, where each token carries spatial and directional information that facilitates structural inference. This stands in contrast to models that rely on full image input, which may degrade more rapidly under occlusion. In practice, this capability suggests strong resilience in adverse conditions, such as cameras partially blocked by mud, debris, or from glare or lens flares, so long as a mechanism exists to identify the visible patches. The network naturally handles such cases without modification, making it particularly well-suited for deployment in unstructured or unpredictable environments.

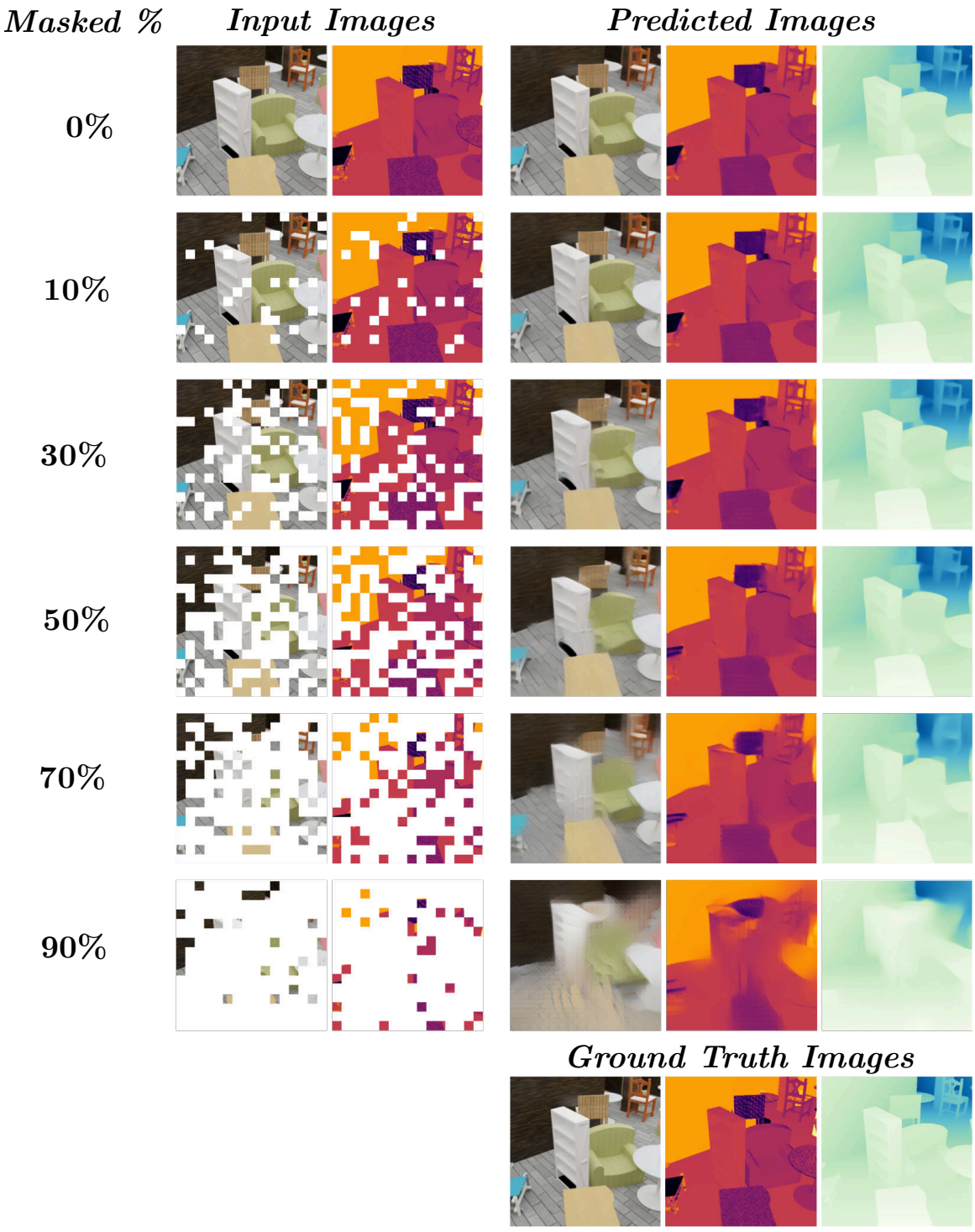

Figure 5.15 – Reconstruction results with increasing levels of masking. Visually the reconstruction maintains a high quality even at 50% masking. The network has an impressive ability to predict scene geometry using heavily masked inputs.

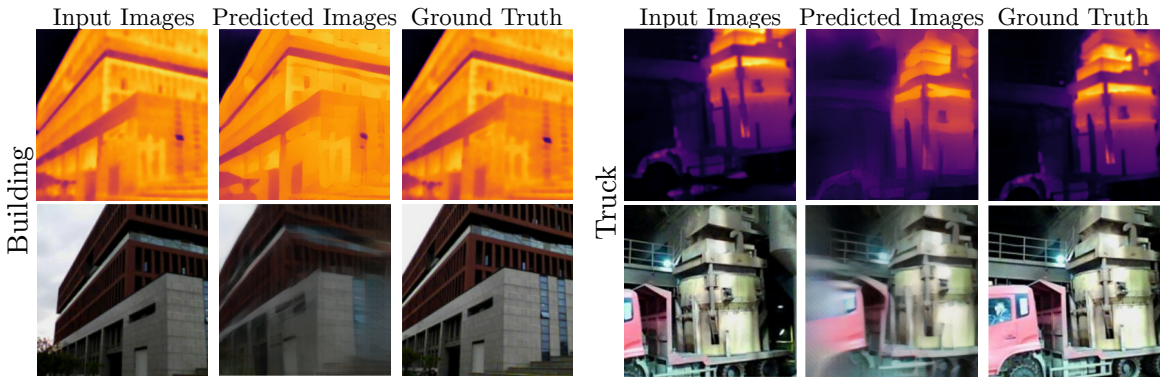

Figure 5.16 – Qualitative results on the ThermalGaussian dataset [79]**.** The model generates consistent RGB-thermal renderings without additional training.

5.4.8 Real-World Thermal Results

Fig. 5.16 shows qualitative renderings on the ThermalGaussian dataset [79]. Due to the dataset's limited size, we evaluate via inference only. The results demonstrate that the model can process real-world RGB-thermal inputs and produce accurate renderings, even in environments different from the training domain, highlighting its potential for real-world deployment. Some edge effects are present when rendering beyond the spatial extent of inputs; for instance, the truck cabin is extrapolated using nearby visual information. While performance is promising, a simulation-to-real gap remains, we hypothesise, due to differences in scene type, motion, and simulated thermal fidelity, representing an important avenue for future work.

5.4.9 Training and Inference Time

As with most transformer-based architectures, training our multimodal reconstruction model demands substantial computational resources and time. The model used to produce the results presented in this chapter was trained over the course of three days using two NVIDIA RTX 6000 Ada GPUs. While additional training time and larger-scale datasets could further refine performance, the current setup was sufficient to validate the model's key capabilities and draw the conclusions reported above.

Further scaling is unlikely to significantly alter the observed trends in modality fusion, generalisation, or robustness.

Inference timing results are presented in Table 5.6. We observe that encoding time increases non-linearly with the number of input images, owing to the quadratic complexity of self-attention. As the number of input views and thus tokens increases, the number of attention comparisons grows rapidly due to the fully connected nature of self-attention operations. In contrast, rendering time increases only marginally with additional inputs.

Despite the resource demands, the system is capable of performing real-time rendering at approximately 20 frames per second on a single RTX 6000 Ada GPU. This makes the model practical for a range of interactive or time-sensitive applications, particularly in robotics, where frame-rate responsiveness is often critical.

The ray-based positional embedding mechanism employed in this model relies on a RoPE process to be applied multiple times per transformer layer. The 2D version of RoPE used in DUSt3R for instance, was implemented in CUDA for efficiency, while for our RoRE, it was implemented in Python. A CUDA reimplementation of our version RoRE would likely offer substantial training and inference speed improvements as approximately 30% of the time per iteration is currently used for RoRE layers.

5.4.10 Energy Usage

The development and training of large-scale machine learning models require significant computational resources, which in turn consume substantial amounts of energy. In light of increasing concerns about the environmental impact of artificial intelligence development, it is important to acknowledge and transparently report the energy footprint of such work. This is particularly relevant when research is conducted in regions where the energy grid remains predominantly powered by fossil fuels, as is the case in Australia.

To support transparency and promote awareness around sustainability in the field, we tracked the total energy consumption associated with the experiments and model

Table 5.8 – Energy consumption during model development and estimated equivalent emissions.

Metric	Estimate
Total Energy Used	2554 kWh
CO_2-equivalent emissions	1402 kg
Equivalent vehicle distance driven	11,685 km
Equivalent household energy usage	200 days

development presented in this chapter. Energy usage was monitored using the open-source `CodeCarbon` Python package [22], which estimates both energy consumption and associated carbon emissions based on hardware usage and local grid intensity.

Table 5.8 summarises the estimated energy usage during the course of this work. A total of 2554 kilowatt-hours (kWh) were consumed, resulting in an estimated 1402 kilograms of carbon dioxide (CO_2) emissions. To contextualise these figures, we provide equivalent estimates: this is roughly equal to driving an average petrol vehicle for 11,685 kilometres[2], a distance nearly equivalent to the diameter of the Earth at the equator (approximately 12,742 km). It is also comparable to around 200 days of typical household electricity usage based on personal consumption data.

We encourage the broader research community to adopt similar practices for tracking and reporting energy use. Doing so is a first step toward more ethically and environmentally responsible AI research, and contributes to ongoing discussions around sustainability, efficiency, and the societal costs of large-scale machine learning.

5.5 Discussion and Future Work

In this chapter, we presented a transformer-based framework for multi-camera, multi-modal scene understanding, capable of integrating information from heterogeneous sensors in a geometrically consistent and spatially aligned manner. The system operates in a feedforward fashion and generalises across varying numbers and combinations of input cameras, modalities, and viewpoints. By working in ray space and leveraging

[2]Estimated using a value of 0.12 kgCO_2 per kilometer, which is the average car according to the European Environment Agency

a novel positional embedding scheme, the model constructs a unified scene representation without requiring handcrafted fusion strategies or modality-specific design.

We evaluated the quality of fusion indirectly through novel view synthesis and depth estimation tasks, using reconstruction accuracy and depth consistency as proxies for the coherence of the internal representation. While not an explicit metric of fusion quality, this approach provides a strong indication that the network has successfully aligned information across modalities into a shared understanding of the scene.

This work represents a significant step toward the long-term goal of plug-and-play, camera-agnostic vision systems, where diverse sensor arrays can be deployed without the need for specialised retraining or fusion pipelines. Although our experiments were limited to simulation and focused on just two modalities, RGB and thermal, this configuration already offers practical benefits, such as enhanced robustness in scenes with occlusion or reduced visibility. The framework is inherently extensible, and could be adapted to incorporate additional modalities (e.g. depth, event, or hyperspectral cameras) depending on the application domain.

Consider, for example, a field-deployed agricultural robot that uses RGB and thermal cameras to monitor crop health. A new low-cost multispectral sensor becomes available, promising improved detection of early-stage plant stress. Traditionally, integrating this sensor would require significant engineering effort: building a new dataset, calibrating the new modality, designing a fusion pipeline accommodate the new data stream. With our framework, the process could be far simpler. The robot could be equipped with the new sensor, its pose and intrinsics measured, or inferred, and data from it immediately integrated into the existing model using the same self-supervised masking and ray-based reasoning strategy. No paired ground truth or fusion heuristics would be required. This flexibility lowers the barrier to sensor innovation, enabling more rapid deployment of emerging hardware in real-world systems.

The ability to reason jointly over multiple camera views and sensor types has important implications for autonomous systems operating in complex environments, such as underwater inspection, firefighting and infrastructure monitoring. These domains often require multi-modal sensing for safety, reliability, or performance, and

our method offers a pathway for integrating such sensors into a common spatial representation. Furthermore, the scene representations produced by the model can serve as inputs to downstream tasks such as cross-modal change detection, semantic classification, or navigation, particularly in cases where multimodal context is essential for disambiguating scene content.

Looking forward, several avenues for future research emerge:

- Transition from simulation to real-world multimodal datasets, to assess generalisation and robustness under real sensor noise and hardware constraints.
- Expand the set of supported modalities, including depth, event, or hyperspectral sensors, to enable richer environmental understanding.
- Evaluate performance on downstream tasks, such as classification, semantic segmentation, or change detection in multi-modal domains.
- Integrate with the NOCaL framework (Chapter 3) to enable joint learning of camera intrinsics and extrinsics, providing a complete end-to-end system from calibration to scene understanding.

Overall, this chapter marks a meaningful advance toward automated, cross-modal camera integration, furthering the goal of removing barriers and making it easier to deploy cameras with less human intervention, which leads to robust vision systems that operate in all environments, from a robotic vacuum in your home to a lunar rover. By fusing information across modalities and viewpoints into a single representation, this work contributes to the broader goal of enabling intelligent systems that can perceive and reason reliably today, while future-proofing for integration with the cameras of tomorrow.

Chapter 6

Conclusions and Future Directions

"The important thing is not to stop questioning."

— Albert Einstein

In the preceding chapters, we introduced three complementary methods that address distinct challenges in the deployment of camera systems, spanning calibration, geometric adaptation, and multi-modal integration. This final chapter provides a concise summary of the thesis contributions in context with the broader goals and explores several promising directions for future research.

6.1 Summary

In this thesis, we addressed key challenges in deploying computer vision systems across diverse and often unconventional camera configurations. Specifically, we targeted three major barriers to flexible and robust vision system deployment: camera calibration, architectural adaptation to new sensor geometries, and the integration of heterogeneous multi-camera sensor data. These challenges are pervasive in robotics and autonomous systems where the sensing platform may vary dramatically in terms of optical properties, geometric layout, and environmental constraints.

We began in Chapter 3 by addressing the problem of monocular camera self-calibration in unconstrained environments. Recognising the burden of traditional calibration routines and the scarcity of labelled data in real-world scenarios. We introduced NOCaL, a semi-supervised framework that jointly learns intrinsic and extrinsic camera parameters alongside odometry. Leveraging the geometric structure of light fields and a pretrained hypernetwork-based renderer, NOCaL enables learning from unlabelled video sequences with minimal motion labels. At the time of publication, this work represented a step forward in using LFNs for real-world task, enabling self-calibration, it showed improved performance compared to the state-of-the-art unsupervised alternative [129]. This contribution showed that self-supervision from light field rendering could produce competitive odometry and calibration results without requiring known camera models. This represents a small step towards the vision of online calibration of deployed sensors where recalibration is difficult or impossible.

Building on the need for geometric adaptability, Chapter 4 introduced RectConv, a novel convolutional layer that modifies the sampling behaviour of CNNs to account for image distortion directly during inference. Rather than retraining networks or rectifying input images, which can be computationally expensive and informationally destructive, RectConv adjusts the convolutional sampling grid based on the input camera's calibration model. We demonstrated the effectiveness of this approach across segmentation and detection tasks on wide-FOV imagery, showing improved performance with minimal overhead. This contribution provides a practical tool for

deploying existing, pretrained models across diverse imaging geometries without the cost of retraining.

Finally, Chapter 5 explored the integration of multi-camera, multi-modal vision systems. These systems promise richer scene understanding by combining sensors such as RGB and thermal, but they introduce significant fusion challenges. We presented a transformer-based architecture that uses a novel ray-based rotary embedding to encode both viewpoint and modality information into a unified geometric space. Through masked, cross-view rendering tasks, the model learns correspondences across sensor types without requiring explicit alignment or manual fusion strategies. We validated this design in simulations across multiple fusion scenarios, demonstrating its ability to generalise across modality combinations and spatial configurations. This work lays the foundation for robust, modality-agnostic visual understanding in complex environments.

Across these three contributions, we demonstrated a consistent philosophy: that camera-aware geometric reasoning in ray space, coupled with modern machine learning architectures, enables more flexible, adaptable, and robust computer vision pipelines. By shifting the focus away from handcrafted designs and towards learnt representations that directly incorporate imaging geometry, we reduce the engineering burden associated with new hardware while improving adaptability and generalisability. A further unifying theme is the use of self-supervision wherever possible. This approach alleviates the reliance on labelled data, a major bottleneck in deploying custom vision systems, and promotes broader generalisation across tasks, sensors, and environments. It also reflects a more natural mode of learning: much like how humans acquire an understanding of depth and geometry through exploration and experience, rather than being shown curated examples, our methods learn to perceive structure and appearance from unlabelled observations using intrinsic cues such as multi-view consistency and cross-modal reconstruction.

Collectively, the methods presented in this thesis advance the field towards realising truly plug-and-play vision systems. These are systems capable of accommodating unknown or evolving camera parameters, operating seamlessly across modalities, and

generalising across diverse environments with minimal calibration or human intervention.

A critical advantage of such systems is their potential for future-proofing. In many real-world deployments, including long-duration autonomous missions, fixed infrastructure monitoring, or industrial robotics, sensor configurations may change over time due to temperature, physical movement, or vibrations [96] as well as the integration of new technologies. Currently, adapting to such changes often requires expensive and time-consuming recalibration, bespoke retraining, or complete system redesign. The ray-based, self-supervised approaches introduced in this thesis offer a pathway towards vision systems that maintain functionality and accuracy even as sensor hardware evolves, supporting more modular, resilient, and scalable deployments.

We anticipate that these methods will benefit researchers and practitioners aiming to deploy adaptable vision systems in challenging conditions. Application areas include autonomous inspection in industrial environments, robotic navigation in outdoor or unstructured spaces, and exploration in remote or extreme settings such as the deep ocean or planetary surfaces, where pre-calibration or recalibration is impractical and sensor behaviour may change unpredictably.

The principles explored in this thesis have broader significance across other fields that depend on reliable and adaptable vision systems. In cinematography and mixed-reality production, rapid adaptation to varying camera setups could streamline creative workflows. In augmented and virtual reality, robust fusion of RGB, depth, and thermal data could support more immersive and dependable user experiences. Environmental monitoring platforms may benefit from the ability to integrate diverse and opportunistically deployed sensors. In the medical imaging domain, techniques that generalise across different imaging geometries or modalities could support more consistent and accessible diagnostics, particularly in resource-constrained scenarios.

6.2 Future Work

While this thesis has made significant strides towards plug-and-play camera deployment, it also opens the door to a range of new research directions. In this section, we outline a set of potential future avenues that may further expand the applicability, efficiency, and robustness of these systems in real-world settings.

Generalising to More Modalities and Camera Types. Across Chapters. 3, 4, and 5 there is clear potential to extend the presented methods to a wider range of camera types, projection models, and sensing modalities. Chapter 3 used monocular RGB cameras with lens distortion, but the NOCaL framework could be applied to non-central projection models such as fisheye or catadioptric cameras. Chapter 4 focused on wide- FOV imagery, though RectConv is geometry-agnostic and could extend to omnidirectional, light field, inputs, given known or estimable ray geometry. Chapter 5 addressed RGB and thermal imagery, but the framework is well-suited to incorporating additional modalities such as polarised light, hyperspectral, or, potentially, event cameras. Future work could explore how to embed and interpret these diverse sensor types in a shared ray-space representation, and how to adapt the presented methods to handle sensors with fundamentally different spatial, spectral, or temporal properties. This would enable a more universal framework for perception, capable of supporting a wide variety of application-specific sensor suites.

Looking further there are computational imaging techniques, such as coded aperture or single pixel cameras, which are not well represented as a single ray per pixel. Instead, these cameras integrate multiple rays of the incoming light onto a pixel. Meaning the ray based philosophy could not be directly applied to these cameras. However, we believe that with additional modelling of the camera to rays relationship to allow for multiple rays to be mapped to a single pixel, the approaches taken in this work could potentially be extended to these camera types.

Towards Online Calibration and Dynamic Camera Arrays. Chapters 4 and 5 focused on systems where camera intrinsics and extrinsics are assumed to be known, allowing explicit ray computation for accurate geometric reasoning. However, in many

real-world scenarios, camera parameters may be unknown or may vary over time due to mechanical flex, thermal drift, or changes in the sensor configuration. One promising future direction is to combine the self-calibration capabilities introduced in Chapter 3 with the network adaptation strategy from Chapter 4. This combination could enable models to automatically adjust their processing to account for changes in camera geometry, maintaining performance even as the underlying imaging conditions shift. A further extension would be to integrate the self-calibration method from Chapter 3 with the multi-camera fusion framework developed in Chapter 5. This would result in a fully learnable system capable of adapting to both unknown modalities and variable imaging geometries. Such an approach could support the development of plug-and-play camera arrays, where new or repositioned sensors can be added to a system without the need for manual recalibration. This capability would be particularly valuable for autonomous systems operating in complex, dynamic, or remote environments.

Real-World Deployment. While our evaluation in Chapter 5 relied on simulated RGB-thermal data, real-world deployment presents additional complexities including sensor noise, varying frame rates, thermal drift, and imperfect alignment. Future work could focus on deploying the model on real hardware, such as autonomous ground or aerial vehicles, to validate its effectiveness under operational conditions. This transition from simulation to reality also opens questions about sim-to-real transfer, robustness to domain shift, and the role of fine-tuning in adapting pre-trained multimodal models.

Continual Learning. Another future direction involves enabling these models to adapt continuously during deployment. Real-world vision systems often encounter new environments, sensor configurations, or operating conditions not seen during training. Future research could investigate continual learning strategies that allow models to incorporate new information incrementally, without suffering from catastrophic forgetting or requiring full retraining. Such methods would support long-term deployment of vision systems in dynamic scenarios, enabling them to improve over time as new data becomes available.

Downstream Integration. The methods developed in this thesis could be extended to support a broader range of tasks beyond view synthesis, odometry, depth estimation and segmentation. Downstream applications such as multi-modal object detection, cross-modal tracking, or scene segmentation could benefit from the representations produced by our frameworks. One promising direction is to adapt our architectures to support multi-task learning where a single network jointly infers geometry, semantics, and dynamics across modalities. This would be particularly valuable in fields such as search-and-rescue robotics, planetary science, and agriculture, where systems must operate in complex, unstructured environments with limited human supervision.

Upstream Integration. A promising avenue for future work involves integrating the insights developed in this thesis with research on upstream camera hardware design. There is an increasing interest in systems that are optimised end to end, from sensor configuration through to task-specific performance [119]. The approaches presented in this thesis, particularly the use of ray-based representations and self-supervised learning, are well aligned with this goal. These strategies provide a flexible foundation that could be incorporated into sensor design workflows, enabling co-development of hardware and algorithms for improved robustness, adaptability, and efficiency in practical vision systems.

Looking ahead, we envision a future where perception systems are no longer constrained by fixed optics, sensor designs, or tightly engineered pipelines. Instead, learning-based systems will flexibly adapt to the constraints and opportunities of their environment and hardware, much like biological vision systems do. The methods developed in this thesis represent a step in that direction, and we are optimistic about the research opportunities they help to unlock.

List of References

[1] Michał Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 2022.

[2] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[3] Benjamin Attal, Jia-Bin Huang, Michael Zollhöfer, Johannes Kopf, and Changil Kim. Learning Neural Light Fields with Ray-Space Embedding Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[4] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. MultiMAE: Multi-modal multi-task masked autoencoders. In *Proceedings of the European Conference on Computer Vision*, 2022.

[5] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[6] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[7] Mark Boss, Andreas Engelhardt, Abhishek Kar, Yuanzhen Li, Deqing Sun, Jonathan T. Barron, Hendrik P. A. Lensch, and Varun Jampani. SAMURAI: Shape and material from unconstrained real-world arbitrary image collections. In *Advances in Neural Information Processing Systems*, 2022.

[8] Duane Brown. Decentering distortion of lenses. *Photogrammetric engineering*, 1996.

[9] Neil A. Campbell, Jane B. Reece, Lisa A. Urry, Michael L. Cain, Steven A. Wasserman, Peter V. Minorsky, and Robert B. Jackson. *Biology*. Pearson Benjamin Cummings, San Francisco, CA, 8th edition, 2008. ISBN 978-0805368444.

[10] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012, 2015.

[11] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian Splats from Image Pairs for Scalable Generalizable 3D Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[12] Kani Chen, Shaojun Guo, Yuanyuan Lin, and Zhiliang Ying. Least absolute relative error estimation. *Journal of the American Statistical Association*, 2010.

[13] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. In *arXiv preprint arXiv:1706.05587*, 2017.

[14] Ping-Yang Chen, Jun-Wei Hsieh, Ming-Ching Chang, Munkhjargal Gochoo, Fang-Pang Lin, and Yong-Sheng Chen. Fisheye Multiple Object Tracking by Learning Distortions Without Dewarping. In *IEEE International Conference on Image Processing*. IEEE, 2023.

[15] Qian Chen, Shihao Shu, and Xiangzhi Bai. Thermal3D-GS: Physics-induced 3d gaussians for thermal infrared novel-view synthesis. In *European Conference on Computer Vision*. Springer, 2024.

[16] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-PoseNet: Absolute pose regression with photometric consistency. In *International Conference on 3D Vision*, 2021.

[17] Peter Christen, David J Hand, and Nishadi Kirielle. A review of the f-measure: its history, properties, criticism, and alternatives. *ACM Computing Surveys*, 2023.

[18] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018.

[19] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. SphereNet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision*, 2018.

[20] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[21] Jonathan Courbon, Youcef Mezouar, Laurent Eckt, and Philippe Martinet. A generic fisheye camera model for robotic applications. In *Intelligent Robots and Systems*. IEEE, 2007.

[22] Benoit Courty, Victor Schmidt, Sasha Luccioni, Goyal-Kamal, MarionCoutarel, Boris Feld, Jérémy Lecourt, LiamConnell, Amine Saboni, Inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, Marc Alencon, Michał Stęchły, Christian Bauer, Lucas Otávio N. de Araújo, JPW, and MinervaBooks. mlco2/codecarbon: v2.4.1. 2024.

[23] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[24] Carlos R. Del-Blanco, Pablo Carballeira, Fernando Jaureguizar, and Narciso García. Robust people indoor localization with omnidirectional cameras using a grid of spatial-aware classifiers. *Signal Processing: Image Communication*, 2021.

[25] Youming Deng, Wenqi Xian, Guandao Yang, Leonidas Guibas, Gordon Wetzstein, Steve Marschner, and Paul Debevec. Self-calibrating gaussian splatting for large field-of-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

[26] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 2023.

[27] Sundara Tejaswi Digumarti, Joseph Daniel, Ahalya Ravendran, Ryan Griffiths, and Donald G. Dansereau. Unsupervised learning of depth estimation and visual odometry for sparse light field cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[28] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[29] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning SO(3) equivariant representations with spherical CNNs. In *Proceedings of the European Conference on Computer Vision*, 2018.

[30] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 2010.

[31] Jose M. Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. CAM-Convs: Camera-aware multi-scale convolutions for single-view depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[32] Jiading Fang, Igor Vasiljevic, Vitor Guizilini, Rares Ambrus, Greg Shakhnarovich, Adrien Gaidon, and Matthew R. Walter. Self-supervised camera self-calibration from video. In *IEEE International Conference on Robotics and Automation*, 2022.

[33] FLIR. Free - FLIR Thermal Dataset for Algorithm Training | Teledyne FLIR. `https://www.flir.com.au/oem/adas/adas-dataset-form/`. 2018.

[34] Georg Glaeser and Hannes F Paulus. *The evolution of the eye*. Springer, 2015.

[35] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010.

[36] Clément Godard, Oisín Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[37] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[38] Ryan Griffiths and Donald G Dansereau. Adapting CNNs for Fisheye Cameras without Retraining. In *IEEE International Joint Conference on Neural Networks*. IEEE, 2025.

[39] Ryan Griffiths and Donald G. Dansereau. RoRE: Rotary Ray Embedding for Generalised Multi-Modal Scene Understanding. In *International Conference on Learning Representations*, 2026.

[40] Ryan Griffiths, Jack Naylor, and Donald G Dansereau. NOCaL: Calibration-free semi-supervised learning of odometry and camera intrinsics. In *IEEE International Conference on Robotics and Automation*, 2023.

[41] Michael D Grossberg and Shree K Nayar. The raxel imaging model and ray-based calibration. *International Journal of Computer Vision*, 2005.

[42] Michael Grupp. evo: Python package for the evaluation of odometry and SLAM. `https://github.com/MichaelGrupp/evo`. 2017.

[43] Xianfeng Gu, Steven J Gortler, and Michael F Cohen. Polyhedral geometry and the two-plane parameterization. In *Rendering Techniques' 97: Proceedings of the Eurographics Workshop in St. Etienne, France, June 16–18, 1997 8*. Springer, 1997.

[44] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *Proceedings of the European Conference on Computer Vision*, 2018.

[45] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017.

[46] Liming Han, Yimin Lin, Guoguang Du, and Shiguo Lian. DeepVIO: Self-supervised Deep Learning of Monocular Visual Inertial Odometry using 3D Geometric Constraints. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[47] Mariam Hassan, Florent Forest, Olga Fink, and Malcolm Mielle. ThermoNeRF: Joint RGB and thermal novel view synthesis for building facades using multimodal neural radiance fields. *arXiv preprint arXiv:2403.12154*, 2024.

[48] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[49] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[50] Xingyi He, Hao Yu, Sida Peng, Dongli Tan, Zehong Shen, Hujun Bao, and Xiaowei Zhou. Matchanything: Universal cross-modality image matching with large-scale pre-training. *arXiv preprint arXiv:2501.07556*, 2025.

[51] Thomas J Herbert. Calibration of fisheye lenses by inversion of area projections. *Applied Optics*, 1986.

[52] Alain Hore and Djemel Ziou. Image quality metrics: PSNR vs. SSIM. In *International Conference on Pattern Recognition*. IEEE, 2010.

[53] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, and Douglas Eck. Music transformer: Generating music with long-term structure. *International Conference on Learning Representations*, 2019.

[54] Jeffrey Ichnowski, Yahav Avigal, Justin Kerr, and Ken Goldberg. Dex-NeRF: Using a neural radiance field to grasp transparent objects. In *Conference on Robot Learning*, 2021.

[55] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 1901.

[56] Max Jaderberg, Karen Simonyan, and Andrew Zisserman. Spatial transformer networks. *Advances in Neural Information Processing Systems*, 2015.

[57] Yunho Jeon and Junmo Kim. Active convolution: Learning the shape of convolution for image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[58] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[59] Jianbo Jiao, Ying Cao, Yibing Song, and Rynson Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *Proceedings of the European Conference on Computer Vision*, 2018.

[60] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: A Large View Synthesis Model with Minimal 3D Inductive Bias. In *International Conference on Learning Representations*, 2025.

[61] Ziyi Jin, Zhixue Li, Tianyuan Gan, Zuoming Fu, Chongan Zhang, Zhongyu He, Hong Zhang, Peng Wang, Jiquan Liu, and Xuesong Ye. A novel central camera calibration method recording point-to-point distortion for vision-based human activity recognition. *Sensors*, 2022.

[62] Juho Kannala and Sami S Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[63] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 2023.

[64] Songeun Kim and Soon-Yong Park. Expandable Spherical Projection and Feature Concatenation Methods for Real-Time Road Object Detection Using Fisheye Image. *Applied Sciences*, 2022.

[65] Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J Davison. Eschernet: A generative model for scalable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[66] Kana Kurata, Hitoshi Niigaki, Xiaojun Wu, and Ryuichi Tanida. MultiBARF: Integrating Imagery of Different Wavelength Regions by Using Neural Radiance Fields. *arXiv preprint arXiv:2503.15070*, 2025.

[67] Michael Land. *Photoreception*. Encyclopedia Britannica, 2020.

[68] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 2015.

[69] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3D with MASt3R. In *Proceedings of the European Conference on Computer Vision*, 2024.

[70] Peter R Lewis, Lukas Esterle, Arjun Chandra, Bernhard Rinner, Jim Torresen, and Xin Yao. Static, dynamic, and adaptive heterogeneity in distributed smart camera networks. *ACM Transactions on Autonomous and Adaptive Systems*, 2015.

[71] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. *arXiv preprint arXiv:2507.10496*, 2025.

[72] Zhong Li, Liangchen Song, Celong Liu, Junsong Yuan, and Yi Xu. NeuLF: Efficient Novel View Synthesis with Neural 4D Light Field. In *Eurographics Symposium on Rendering*, 2022.

[73] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[74] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[75] Zikang Liu, Longteng Guo, Yepeng Tang, Tongtian Yue, Junxian Cai, Kai Ma, Qingbin Liu, Xi Chen, and Jing Liu. VRoPE: Rotary Position Embedding for Video Large Language Models. 2025.

[76] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[77] Manuel Lopez, Roger Mari, Pau Gargallo, Yubin Kuang, Javier Gonzalez-Jimenez, and Gloria Haro. Deep single image camera calibration with radial distortion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[78] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Ieee, 1999.

[79] Rongfeng Lu, Hangyu Chen, Zunjie Zhu, Yuhang Qin, Ming Lu, Le Zhang, Chenggang Yan, and Anke Xue. ThermalGaussian: Thermal 3D gaussian splatting. In *International Conference on Learning Representations*, 2025.

[80] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021.

[81] Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers. In *International Conference on Learning Representations*, 2024.

[82] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. LENS: Localization enhanced by NeRF synthesis. In *Conference on Robot Learning*, 2022.

[83] Ali Mosleh, Avinash Sharma, Emmanuel Onzon, Fahim Mannan, Nicolas Robidoux, and Felix Heide. Hardware-in-the-loop end-to-end optimization of camera image processing pipelines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[84] Claus Müller. *Spherical harmonics*, volume 17. Springer, 2006.

[85] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 2022.

[86] Julius Plücker. On a new geometry of space. *Philosophical Transactions of the Royal Society of London*, 1865.

[87] Dhanesh Ramachandram and Graham W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 2017.

[88] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[89] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

[90] Hazem Rashed, Eslam Mohamed, Ganesh Sistu, Varun Ravi Kumar, Ciaran Eising, Ahmad El-Sallab, and Senthil Yogamani. Generalized object detection on fisheye cameras for autonomous driving: Dataset, representations and baseline. In *Winter Conference on Applications of Computer Vision*, 2021.

[91] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 1958.

[92] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Ieee, 2011.

[93] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[94] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[95] Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light Field Networks: Neural Scene Representations with Single-Evaluation Rendering. In *Advances in Neural Information Processing Systems*, 2021.

[96] M. J. Smith and E. Cope. The effects of temperature variation on single-lens-reflex digital camera calibration parameters. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2010.

[97] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

[98] Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360 imagery. *Advances in Neural Information Processing Systems*, 2017.

[99] Yu-Chuan Su and Kristen Grauman. Kernel transformer networks for compact spherical convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[100] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew Davison. iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[101] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light Field Neural Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[102] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

[103] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *Proceedings of the European Conference on Computer Vision*, 2018.

[104] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[105] Önder Tuzcuoğlu, Aybora Köksal, Buğra Sofu, Sinan Kalkan, and A Aydin Alatan. Xoftr: Cross-modal feature matching transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[106] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[107] Fatma Vatansever and Michael R Hamblin. Far infrared radiation (FIR): Its biological effects and medical applications: Ferne Infrarotstrahlung: Biologische Effekte und medizinische Anwendungen. *Photonics & lasers in medicine*, 2012.

[108] Johannes von Oswald, Christian Henning, Benjamin F. Grewe, and João Sacramento. Continual learning with hypernetworks. In *International Conference on Learning Representations*, 2020.

[109] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[110] Peng Wang, Yuan Liu, Guying Lin, Jiatao Gu, Lingjie Liu, Taku Komura, and Wenping Wang. Progressively-connected Light Field Network for Efficient View Synthesis. *arXiv preprint arXiv:2207.04465*, 2022.

[111] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[112] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUST3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[113] Yu-An Wang and Yun-Nung Chen. What do position embeddings learn? an empirical study of pre-trained language model positional encoding. *arXiv preprint arXiv:2010.04903*, 2020.

[114] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.

[115] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[116] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. CroCo v2: Improved Cross-view Completion Pre-training for Stereo Matching and Optical Flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[117] Weinzaepfel, Philippe and Leroy, Vincent and Lucas, Thomas and Brégier, Romain and Cabon, Yohann and Arora, Vaibhav and Antsfeld, Leonid and Chidlovskii, Boris and Csurka, Gabriela and Revaud Jérôme. CroCo: Self-Supervised Pre-training for 3D Vision Tasks by Cross-View Completion. In *Neural Information Processing Systems*, 2022.

[118] Daniel N Wood, Daniel I Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H Salesin, and Werner Stuetzle. Surface light fields for 3d photography. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 487–496. 2023.

[119] Chengyang Yan and Donald G. Dansereau. TaCOS: Task-specific camera optimization with simulation. In *Winter Conference on Applications of Computer Vision*, 2025.

[120] Kailun Yang, Xinxin Hu, Luis M Bergasa, Eduardo Romera, and Kaiwei Wang. Pass: Panoramic annular semantic segmentation. *Transactions on Intelligent Transportation Systems*, 2019.

[121] Botao Ye, Sifei Liu, Haofei Xu, Li Xueting, Marc Pollefeys, Ming-Hsuan Yang, and Peng Songyou. No Pose, No Problem: Surprisingly Simple 3D Gaussian Splats from Sparse Unposed Images. *arXiv preprint arXiv:2410.24207*, 2024.

[122] Yaozu Ye, Kailun Yang, Kaite Xiang, Juan Wang, and Kaiwei Wang. Universal semantic segmentation for fisheye urban driving images. In *Systems, Man, and Cybernetics*. IEEE, 2020.

[123] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Tsung-Yi Lin, Alberto Rodriguez, and Phillip Isola. NeRF-Supervision: Learning dense object descriptors from neural radiance fields. In *IEEE International Conference on Robotics and Automation*, 2022.

[124] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[125] Senthil Yogamani, Ciarán Hughes, Jonathan Horgan, Ganesh Sistu, Padraig Varley, Derek O'Dea, Michal Uricár, Stefan Milz, Martin Simon, and Karl Amende. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[126] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[127] Jiaming Zhang, Kailun Yang, Chaoxiang Ma, Simon Reiß, Kunyu Peng, and Rainer Stiefelhagen. Bending reality: Distortion-aware transformers for panoramic semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[128] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

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

[130] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo Magnification: Learning view synthesis using multiplane images. *ACM Transactions Graph. Proceedings Special Interest Group on Computer Graphics and Interactive Techniques*, 2018.

[131] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[132] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Appendix A

Appendix

A.1 Multi-modal Configuration

Table A.1 – Multi-modal configuration values.

Section	**Parameter**	**Value/Setting**
Model	Encoder Backbone	ViT Large
	Decoder Backbone	ViT Base
	Positional Embedding	RoRE
	Ray Parameterisation	raymap
	Head Type	dpt
Patch Embeding	Embedding Type	conv
	Patch Size	16x16
Dataset	Context Views	8
	Target Views	6
Optimizer	Learning Rate	5.00e-5
	Warm-up Steps	500
Data Loader	Batch Size (train/val/test)	2
Loss	MSE Weight (λ_{mse})	1.0
	LPIPS Weight (λ_{lpips})	0.05
	Depth Loss Weight (λ_{depth})	0.75

A.2 Multi-camera Renderings

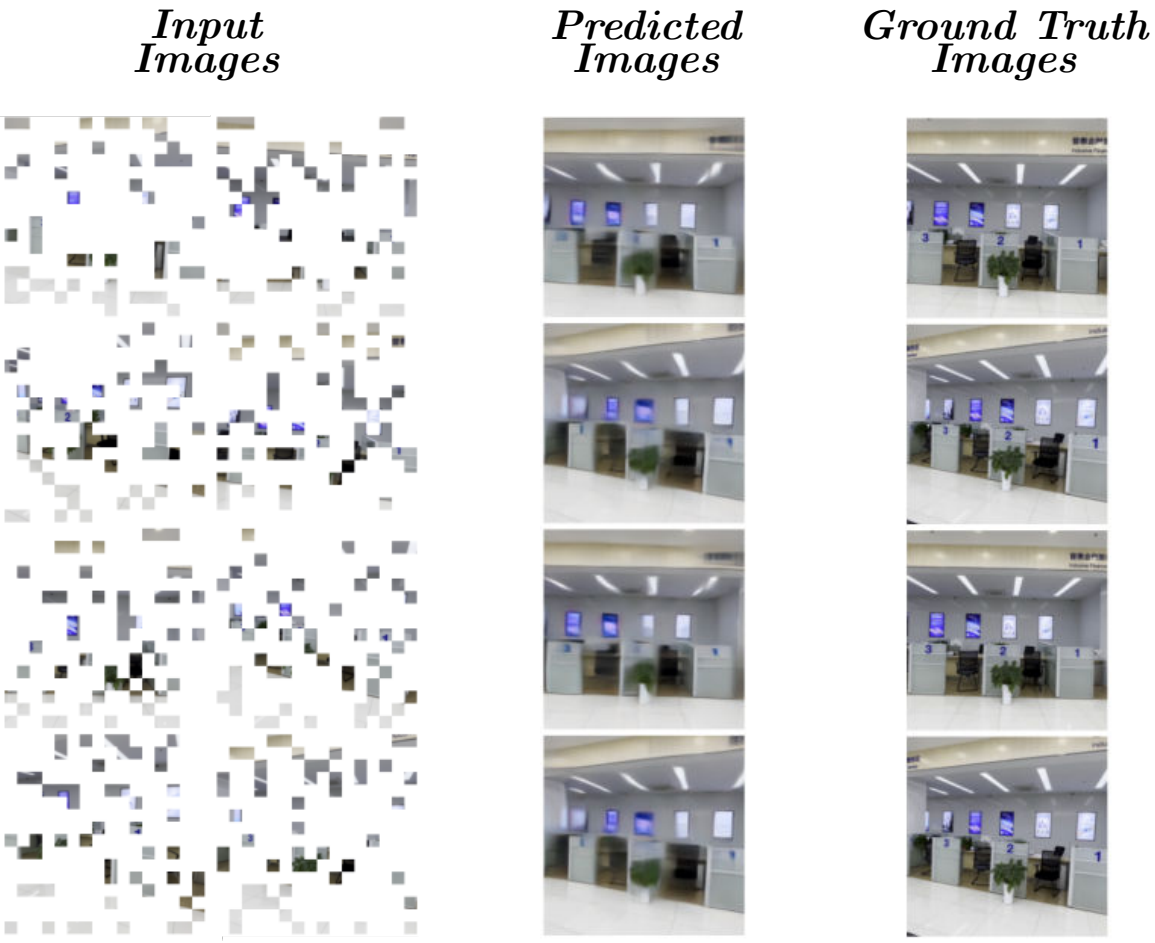

Figure A.1 – Real-world rendering on the DL3DV dataset. With 8 input images masked at 70%. This shows accurate rending in real-world environments even under high masking scenarios.

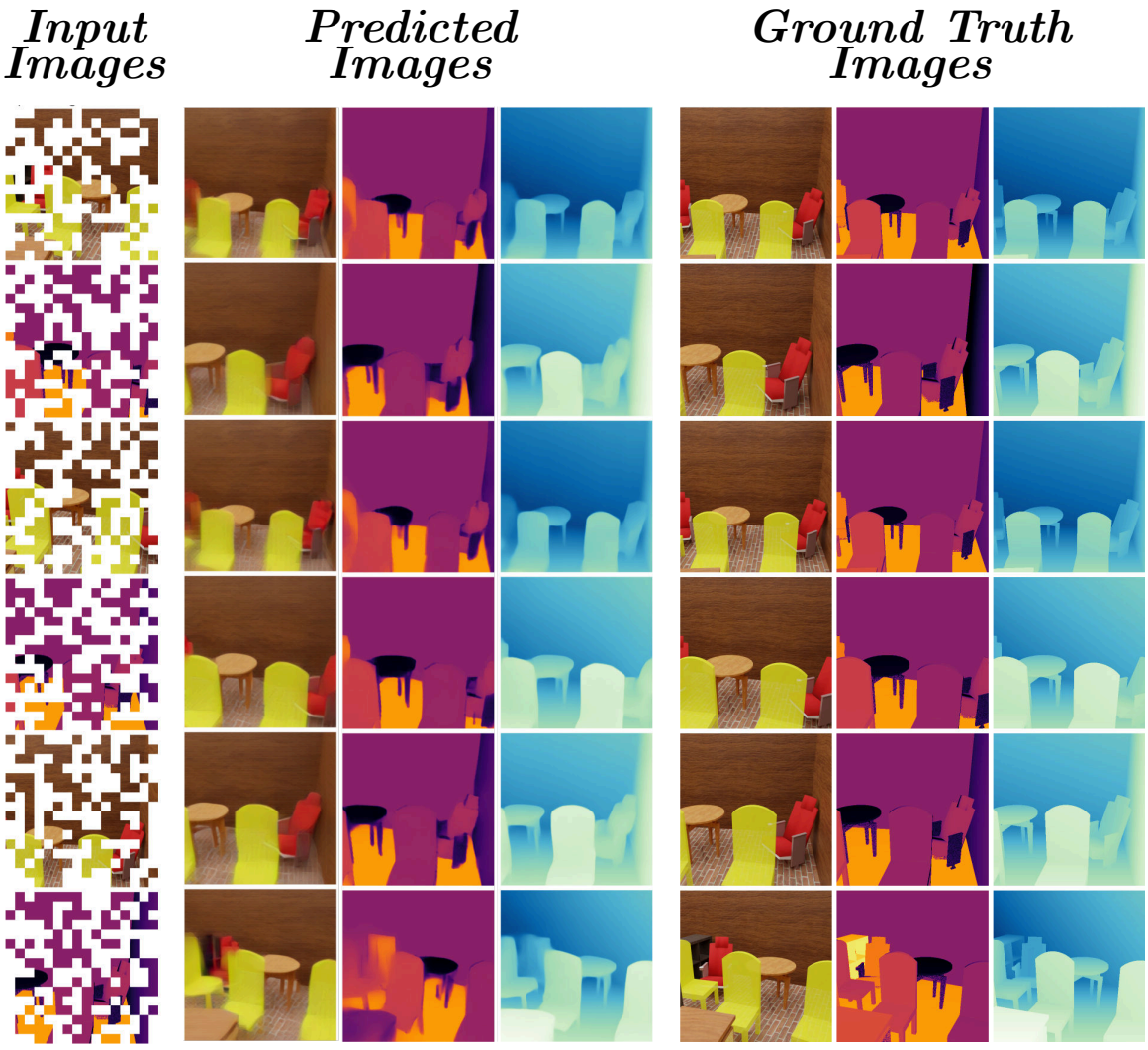

Figure A.2 – A masked, multi-modal, multi-camera rendering configuration. Showing accurate renderings.