Neural field multi-view shape-from-polarisation

Rapee Wanaset

PhD
University of York
Computer Science

June 2025

Abstract

In this thesis, we provide three novel contributions towards 3D reconstruction by leveraging polarimetric information. First, we modify NeRF to work with the input obtained from a polarisation camera. In particular, we extend NeRF to cover 12 channels of the camera sensor. Unlike previous works, this is the first time that the model is fitted directly to raw polarisation sensor data, bypassing the need for demosaicing. Since the polarisation state of reflected light encodes the surface normal used for reconstructing 3D geometry, our method provides richer information about surface orientation than RawNeRF which uses conventional raw RGB images. This form of input is challenging for the model training due to input sparsity. Nonetheless, we show that this setup works reasonably well with a synthetic dataset, while requiring additional constraints for real-world capture. Secondly, we link surface geometry with polarised radiance through a mixed polarisation model and then inject the physical insights into the training pipeline - significantly improving the geometry prediction of the object in the scene. Rather than guessing the relationship between captured data and surface orientation (as in a 12-channel black box model), the physics-based model could follow the physical rule given by the mixed polarisation model. Nevertheless, despite its physical understanding, this model neglects practical limitations. Therefore, our last contribution is to investigate the reasons why the model did not behave as expected and tackle the issues related to noise and saturation, which greatly improve the quality of 3D reconstruction - achieving state-of-the-art performance on the PANDORA benchmark.

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for a degree or other qualification at this University or elsewhere. All sources are acknowledged as references.

Contents

List of Figures

List of Tables

1 Introduction

> The beginning is the most important part of the work.
>
> *Plato*

For the last decade, we have seen a massive success in image understanding (e.g. 2D segmentation, 2D bounding box and 2D object detection) primarily due to the availability of large-scale datasets in addition to storage space and computational power becoming more affordable to general research institutions (rather than only well-funded research labs). As the field becomes mature, models have achieved an impressive performance, for example Florence [91] yielding over 99% top-5 accuracy on the ImageNet benchmark. Nevertheless, real-world applications, such as autonomous vehicles and robotics, have struggled to meaningfully perform in daily life (self-driving car is an ongoing experiment in a few cities in the US and China). This implies that only image understanding is not sufficient to inform machines about the physical world and 3D understanding could give an additional cue to unlock these sophisticated tasks.

There are various techniques that provide geometric information of the scene such as shape-from-shading, multi-view stereo (MVS) and photometric stereo. In the same way that artists exploit different colour tone to create an illusion of depth in a 2D image, shape-from-shading primarily deals with the recovery of shape from a gradual variation in an image. Without explicit assumptions about environment lighting and surface materials, shading cues are ambiguous and could be interpreted in various ways. Multi-view stereo leverages consistency across multiple viewpoints. The core idea be-

hind this technique is to extract image features, match the features from a scene overlap within an image pair and then estimate depth information by triangulation. While being robust to an extent, MVS could be fooled when dealing with shiny objects whose specularities vary across different views. Among methods discussed in this paragraph, photometric stereo is the only one relying on active lighting conditions. The key is to maintain camera position while changing the lighting direction. By analysing the variation in reflectance, the surface orientation could be obtained. Because lighting needs to be varied, photometric stereo usually has to be done in studio/laboratory and does not work with moving objects.

In contrast to the above techniques, shape from polarisation (SfP) is a powerful technique that allows us to extract 3D information from the way light interacts with the surface. When unpolarised light reflects from a surface it becomes partially polarised. This applies to both specular reflection [66] and diffuse reflection [2] that arises from transmission out of the surface after subsurface scattering. The degree and angle of polarisation are related to the local surface normal direction and view vector and, hence, their measurement provides constraints for the reconstruction of surface geometry.

However, these methods have seen limited adoption, partly due to the challenges of capturing polarimetric images. Recent advancements, such as commodity division-of-focal-plane (DoFP) sensors that capture polarisation images in a single shot, have mitigated this issue. Nonetheless, polarisation alone is a weak shape cue, providing strong signals only at occluding boundaries for diffuse regions or within sparse specularities.

Multi-view polarisation measurements potentially overcome this restriction. As few as two multi-view measurements of the same point uniquely determine the surface normal direction from polarisation constraints alone (see Chapter 4). The challenge is to choose a representation that is amenable to optimisation while integrating information from multiple views. The recent rise of neural fields [86] and their use for implicit surface representation [89, 92, 47, 35] provides a compact and adaptive parameterisation that can be rendered differentiably, e.g. NeRF [55] and NeuS [84]. It is worth noting that while requiring input from at least two views (in the same way as MVS does), our method does not perform any feature extraction from images and

the information across viewpoints is used to supervise the training process - hence being classified as a shape-from-polarisation method rather than an MVS one.

Recent works have begun to explore the factorisation, i.e. inverse rendering, of neurally-modelled radiance into underlying physical quantities, including illumination, geometry and material properties via the bidirectional reflectance distribution function (BRDF). Capturing and modelling *polarised* radiance offers the potential for higher accuracy, requirement for fewer input views and the resolution of ambiguities that arise when decomposing RGB radiance alone. This provides us a motivation to work with polarised radiance (see Chapter 3). A recent line of work integrates polarisation into neural radiance models [19, 43]. However, they require the full Stokes vector at each pixel and perform a full inverse rendering, entailing estimation of the incident illumination and modelling of a polarised BRDF. Instead, we directly exploit the shape-from-polarisation cue in a way that is independent of the illumination environment and make very limited assumptions about material reflectance models. Moreover, we fit our model directly to raw polarisation sensor data, bypassing the need for demosaicing, which is more complex for a Colour Polarisation Filter Array (CPFA) compared to conventional RGB demosaicing. In contrast to a classic Bayer Filter Array which has 3 colour channels (Red, Green and Blue) inside the 2×2 grid, the CPFA has 12 channels for the combination of 3 colours and 4 polariser orientations inside a larger grid of size 4×4.

As a result of the unique data distribution captured directly from camera sensor, we propose a loss function that is adapted specifically to work with raw images (see Chapter 5).

1.1 Research objectives

1.1.1 How could we effectively utilise polarimetric cues inside coordinate-based neural networks in order to extract scene geometry?

This is the main research topic we have explored throughout this thesis. We naively begin with using polarisation images as input of a black box pipeline (Chapter 3), then explicitly establish how polarised radiance is linked to local surface normal after realising that black box model alone could not perform well (Chapter 4); and lastly, we modify color loss to match a wide range of HDR data, add a theta loss to encourage networks to produce meaningful results and introduce a smoothness loss to reduce the effect of noises on the object geometry (Chapter 5).

1.1.2 Ambiguities naturally arise when a shape-from-polarisation method is applied; how could we disambiguate the ambiguities without relying on additional cues *i.e.* shape-from-X?

When extracting geometry from polarimetric cues, two types of ambiguity arise: azimuthal ambiguity and azimuthal model mismatch. The azimuthal ambiguity is caused by two azimuth angles being indistinguishable, leading to either convex or concave geometry. This is a classic problem in shape-from-polarisation, which usually requires extra information (e.g. shading cues and space carving) to disambiguate. Under an assumption of Lambertian model for diffuse reflection, our work could resolve the ambiguity without relying on additional information from other techniques, by utilising polarimetric cues from at least 2 views. The second type of ambiguity is azimuthal model mismatch: different polarisation model giving different azimuth angle. After carefully investigating natural phenomenon, we realise a duality of diffuse and specular reflections, and develop a mixed polarisation model to avoid the model mismatch.

1.1.3 A benefit of multi-view methods is having access to information from different views; Usually, this provides observations which exceed the degrees of freedom; how could we utilise the exceeding observations for a result fidelity?

Not only does the multi-view information provide sufficient constraints for a unique local surface normal (as briefly mentioned in section 1.1.2), making the problem well-posed, but the information from multiple viewpoints also becomes redundant as the same 3D point being observed many times. The latter notion inspires us to investigate ways to reduce degrees of freedom in the experimental system. We have identified raw image input as the way to do so without sacrificing other benefits offered by the multi-view setup. Intuitively, a 3D point would be observed in red, green, and blue channels from different angles. This paired with an interpolation of neural networks, gives us the ability to predict colors in unobserved channels or even unobserved viewpoints. The ablation study shown in Figure 5.16 confirms the validity of this technique.

1.1.4 Under a multi-view setting, how could we effectively decompose a mixed radiance into diffuse and specular components?

To distinguish between diffuse and specular components, we set an extra assumption that the diffuse reflection follows Lambertian reflectance. That being said, we do not restrict ourselves to a matte surface. Instead, our model composes of diffuse and specular components, thus being capable of handling a variety of materials.

The reason for Lambertian assumption is to remove viewing dependency from diffuse reflection. As a result of this assumption, we know that diffuse radiance is a function of position only, whereas specular radiance is a function of both position and viewing direction. Following this rationale, we set input to diffuse and specular networks accordingly and the radiance decomposition

can be accomplished (Chapter 4).

1.1.5 Generally, HDR data covers a wider range of data when compared with LDR data, leading to a different data distribution; even though neural networks are a universal approximator, in practice, an appropriate gradient descent is crucial for the networks to successfully learn; what is an appropriate loss function which suits HDR data distribution?

In the realm of neural network training, it is fairly common to employ L1 or L2 as a training loss. However, doing so for HDR data, would make the networks biased towards bright image areas. Logarithmic loss is found to be more robust when training network with HDR data, as having a good balance of steep gradient in low-value region (dim pixels) and shallow gradient in high-value region (bright pixels) - leading to a stable training without gradient exploding.

1.1.6 3D reconstruction infers geometry from 2D images which consist of noise, unless being synthesised; how could we reduce the noise in geometry, inherited from the image noise?

Noise is an undesired property of collected data. In the real world, noise always exists, often beyond the control of experimentators. Ideally, in the case that noise is controlable, it is a good practice that the experiment is set up in such a way that has a minimal noise. In the context of computer vision, for instance, the images could be collected in a lab to reduce a lighting variation.

Nonetheless, there is always a capturing noise, which affects the quality of results. In our work, the noises from captured images get baked into

our 3D reconstruction. As a solution, we introduce a smoothness loss into our training, which encourages piecewise-coherent geometry. It is noted that piecewise coherence could be viewed as another assumption in our experiment but the smoothness loss has worked so far with all datasets. In an extreme case where we have complicated geometry, an edge-aware smoother could be an option to eliminate the inherited noises.

1.2 Outline

Other than Chapter 2 which provides an overall picture of the related fields, this thesis is designed to be read in chronological order where the earlier chapter is the foundation of the next one. In this thesis, we cover materials as follow:

- **Chapter 2:** We show works in related fields including shape from polarisation, diffuse-specular separation, and neural inverse rendering. Since our work could be easily extended to performing in a controlled lighting, an overview of light stage methods is also covered. We close the chapter by identifying research gap within the fields and reviewing available datasets.

- **Chapter 3:** Since NeRF [55] is the foundation of our work, we explore related concepts such as volume rendering and Signed Distance Function (SDF). Then we toss a hypothetical question about Shape-from-Polarisation in the context of coordinate-based scene representation i.e. NeRF on polarised images. According to this, we extend NeRF to fit 12 channels offered by CPFA camera sensor. The model demonstrates a promise that the concept is working to some extent, and in some image regions, our model is being able to produce finer details than PANDORA which employs full inverse rendering - hence providing an inspiration for the next chapter.

- **Chapter 4:** The result in Chapter 3 is reasonably good when considering synthetic dataset which is in the ideal context, i.e. perfect focus without capturing noise, whereas the model trained on real datasets

gives a meaningful result but still missing some object parts. These results hint us that the model need a better constraint rather than completely guessing as done in Chapter 3. Therefore, we add physical insights from polarisation models. In particular, we develop a mixed polarisation model showing the link between surface normal direction and camera measurement. New results show an improvement over the ones obtained from the black box model i.e. the model used in Chapter 3. Nonetheless, a small portion of results show undesirable artifacts especially in the real-world scenes. We suspect that the artifacts could be baked into the model due to an imperfect capture which usually occurs in real-world scenarios.

- **Chapter 5:** To conclude this trilogy, we address training practicalities from image capturing to noise handling. We begin with showing two edge cases of captured image, one being too dim and one being saturated, before proposing a new training loss that works well with HDR images used to train our model. Then we carefully investigate how each point in space contributes to final appearance in the image and impose a condition which restricts non-physical points to contribute in volume rendering. Finally, we introduce a smoothing prior to tackle various sources of noise. We observe a great improvement on the results, and as a result, our model achieves state-of-the-art performance on the PANDORA benchmark.

- **Chapter 6:** We wrap up the work with a conclusion. During the process of producing this set of works, we have encountered limitations, seen concurrently emerging works which are promising, and different applications to what we originally intended to work for. We, therefore, make a few suggestions for future researchers who are interested in this line of works.

A paper with content in Chapters 4 and 5 is accepted to 36th Eurographics Symposium on Rendering (EGSR 2025) under title: Neural field multi-view shape-from-polarisation.

2 Related work

> If I have seen further, it is by standing on the shoulders of giants.
>
> *Sir Isaac Newton*

This chapter will briefly show the prior works that have been done in the field. Our aim is to cover a wider picture of the research theme. The detailed discussion on how we develop our methodology can be found in the chapters 3, 4, and 5].

2.1 Shape from polarisation

In 1828, Augustin-Jean Fresnel derived the formulae that relates reflection from and transmission into a surface to the polarisation state of the incident light. This provides the quantitative model of polarisation and can be used to relate the local surface normal orientation and the polarisation state of reflected light.

The Fresnel equations describe the link between surface normal orientation and polarisation properties of light reflected off a surface. Such link is exploited by SfP techniques, aimed at estimating surface normal from polarimetric measurements. While multi-view stereo typically does not work well with smooth, featureless and glossy surfaces, polarisation can be used on a wide range of materials, such as metals [60], dielectrics [2, 27], dark and shiny surfaces [58, 59], as well as transparent ones [57]. Furthermore, *polarisation cameras* are able to record the polarisation state in a single shot,

thus providing a dense cue, only limited by camera resolution, and enabling surface orientation estimation at each pixel.

Many polarisation-based methods either deal with specular or diffuse materials, due to the different reflection phenomena. Atkinson and Hancock [2] assumed diffuse reflectance to estimate the depth map of an object. Miyazaki *et al.* [56] proposed a framework [78] to separate reflection components, from which the object's shape can be inferred. Morel *et al.* [60] developed a SfP method aimed at metals, using a specular polarisation model.

However, most real-world objects exhibit a mixture of both diffuse and specular polarisation, causing model mismatch [77]. Smith *et al.* [74] relaxed the classic assumption and classify each pixel as diffuse dominant or specular dominant. Taamazyan *et al.* [77] used both viewpoint and polarisation information to recover shape of an object, relying on a mixed polarisation model. Polarisation data from at least 2 viewpoints constrains surface normal estimation, posed as a non-linear least square problem. Cui *et al.* [14] used polarimetric multi-view stereo to handle a variety of objects with mixed polarisation reflection, using iso-depth contours to propagate depth from sparse points.

An additional challenge for SfP methods is the azimuthal ambiguity, i.e. two azimuthal angles shifted by π radians cannot be distinguished from polarisation information alone. Miyazaki *et al.* [59] used space carving to estimate the rough structure of an object, before integrating priors to the SfP pipeline. Similarly, Zhu and Smith [97] used multi-view information and a coarse depth map obtained from stereo cues as a guide surface for disambiguation. Kadambi *et al.* [30] combined a single polarisation image with the depth map obtained from an RGBD camera, the latter used to disambiguate normal direction. Proposing PMVIR, Zhao *et al.* [94, 95] address the multi-view SfP problem using a mesh-based representation and do not fully resolve the ambiguity, relying instead on the most plausible azimuth angle. Moreover, being a refinement method, PMVIR requires a reasonable initial shape rather than estimating shape directly from polarimetric cues as generally done in SfP.

By establishing consistency of tangent space among multiple viewpoints, MVAS [11] manages to reconstruct textureless 3D objects which have been

challenging for conventional stereo methods. Leveraging a circularly polarised light, MVAS provides a number of benefits over traditional multi-view photometric stereo (MVPS). First, MVAS works with a wider range of materials i.e. not specific to particular reflectance models such as Lambertian [12] or a microfacet model [88]. Second, MVAS does not need an active capture environment which is a critical requirement for MVPS. In a way, MVAS could be treated as a circular-based SfP integrated with Tangent Space Consistency. PolarPMS [96] exploits photometric and polarimetric consistencies. While being able to reconstruct a 3D object up to pixel-level resolution, in contrast to our methods (see Chapters 3, 4, 5) which employs SDF to implicitly infer geometry, PolarPMS iteratively generates several pairs of depth and normal hypotheses and picks the one that minimises inconsistency between views.

Ba *et al.* [3] and Lei *et al.* [42] tackle the monocular SfP problem, recovering only a single normal map from one view. Both methods train networks using data from many scenes, learning general priors that generalise to unseen novel input. While requiring only a single view for each scene, the networks is a result of averaging multiple views to generate geometric priors and thus could not give the same level of accuracy as the methods that dedicate to a single scene (e.g. NeuS[84]).

2.2 Diffuse-specular separation

Reflected radiance can be accurately modelled as a combination of specular and diffuse reflections. Specular reflection occurs when light is reflected off a smooth surface, whereas diffuse reflection could be due to either sub-surface scattering or reflection from a rough surface. Various methods in computer vision and graphics are simplified under the assumption that reflection from the object is either solely diffuse or solely specular [2, 56, 60]. For this reason, many approaches were developed to separate diffuse and specular components of a surface reflection.

While some methods rely on pixel intensity [72, 51, 76] to separate reflectance components, polarisation-based reflectance separation has been widely exploited. Riviere *et al.* [68] proposed a passive method for uncontrolled

environments, that estimates the reflectance of a planar surface using polarisation measurements from 3 views, one along the normal and two from viewpoints near the Brewster angle. Such measurements are used to fit a sinusoid. Nogué *et al.* [64] proposed planar surface reflectometry using a near-field display, which requires 3 linear polarisation measurements under a fixed display illumination.

Several recent techniques employ neural networks for radiance separation. Inspired by real-time graphics, Boss *et al.* [9] presented a pre-integrated lighting network that converts illumination integration process into a query network, the latter resulting in efficient rendering and radiance decomposition. PhySG [92] tackles radiance separation by using Spherical Gaussians and data-driven embedding to model reflectance and lighting respectively. Dave *et al.* [18] proposed a 2-step method to decompose specular and diffuse reflectances from a single polarimetric image. The initial separation is done by analysing the relationship among polarisation cues and reflected radiance, then refined by the network trained on synthetic scenes.

2.3 Neural inverse rendering

Inverse rendering aims at estimating geometry, material properties and lighting from images. The problem is inherently ill-posed, due to the large space of plausible solutions that can explain acquired images. Recently, neural approaches have grasped the attention of vision and graphics communities [80]. NeRF [55] takes in input a set of photographs, or even raw images [52], to synthesise novel views, by encoding the volume density and colour of a scene within the weights of a coordinate-based Multi-Layer Perceptron (MLP). The quality of rendered results demonstrates the ability of the network to learn material properties, occlusions and specularities. Neural representations have been successfully exploited for relighting [7, 93, 8], shape estimation [8, 84], material editing [93, 8, 38], and object decomposition [87, 38]. Han et al. [29] focus on polarimetric cues, splitting them into geometric and photometric cues derived from Stokes vectors. Cao et al. [10] address 3D reconstruction in textureless areas, deriving depth priors from neural graphics primitives and using a graph-based energy function to resolve and scale normal maps into

depth. While most of the works assume perfect camera pose, BARF [46] applies coarse-to-fine camera registration making reasonable training from inaccurate camera poses.

Explicit representations, such as voxels [85, 32], point clouds [23] and meshes [83, 33], can be used to describe the 3D structure. In recent years, implicit representations have gained increasing attention for their differentiability and ability to achieve high fidelity with similar network size. Signed Distance Function (SDF) [84, 92] could be used to represent object surfaces.

PANDORA [19] incorporates polarisation properties into a neural inverse rendering pipeline. The approach uses multi-view polarisation images, COLMAP [70, 69] camera poses, and binary masks. However, PANDORA performs full inverse rendering, modelling the specular BRDF and incident illumination environment, passing this through a polarised BRDF model and rendering Stokes vectors that are compared to those recorded by the camera. This requires first demosaicing the raw images to provide full Stokes vectors at every pixel.

GNeRP [44] uses a Gaussian representation for surface normals, with the mean indicating overall orientation and covariance capturing high-frequency variation. It employs DoLP (Degree of Linear Polarisation) reweighting to balance higher DoLP in specular regions. The method uses implicit neural BRDF which can handle high-frequency details. NeRSP [29] splits polarimetric cues into geometric and photometric cues. A camera measurement is demosaiced and processed to obtain the Stokes vector (used as photometric cues), from which the geometric cues are also derived. Consequently, both cues in NeRSP are not fully independent, making their proposed losses redundant for constraining scene geometry.

While many works assume single-bounce illumination, polarised rays contain rich information (e.g. as modeled by [37]). NeISF [43] relaxes this assumption, using coordinate-based MLPs to capture the Stokes field of the second-last bounce. Diffuse and specular reflections are modelled separately: diffuse Mueller matrices depend on surface normals, while specular ones depend on microfacet normals.

NPMVS [10] tackles 3D reconstruction in textureless areas. Depth priors are derived from neural graphics primitives [62] using only photometric cues.

An energy function in a graph-based model [97] is minimised to resolve the normal map, which is then converted to a depth map and scaled to fit the scene.

2.4 Lightstage-based capture

In general, lightstage-based methods use spherical illumination patterns to perform a version of photometric stereo. Many light-stage appearance-capture pipelines aim to recover not only geometry but also spatially varying material parameters (see [26] for a survey). For example, Ma et al. [49] introduced the spherical gradient illumination patterns while Lattas et al. [40] use spherical binary patterns. This provides a per-pixel estimate of the surface normal and material properties including diffuse albedo and specular intensity. This cue can be integrated across views using a multiview stereo type approach [25]. Lattas et al. [41] use less-constrained, at desk-based illumination constructed from a panel of LCDs.

Some of these methods use properties of polarisation for the purposes of separating diffuse and specular reflection. This is based on a simplistic model in which specular reflection is assumed to perfectly preserve the plane of linear polarisation while diffuse reflection completely depolarises the reflected light. In fact, diffuse reflection caused by subsurface scattering (as in human skin) partially polarises the light that is transmitted out of the surface (regardless of whether the incident light was polarised or not) and specular reflection similarly partially polarises unpolarised incident light when it is reflected. It is this shape-from-polarisation cue that our method exploits, negating the need for varying illumination patterns.

In addition, the diffuse/specular separation technique used by [49] requires each light source in the lightstage to be polarised with a particular plane of polarisation. The orientation of the polarisation filters can only be tuned for a single viewpoint. This means that such a design cannot be used to provide multiview information. This reduces coverage of the face but also means that geometric, multiview shape cues cannot be combined with the photometric cues. [25] proposed an alternative in which polariser orientations on the lights were arranged such that an approximate separation was

possible from any viewpoint on the equator of the dome. This allows the use of multiview information but at the cost of losing exact diffuse/specular separation.

2.5 Research gap

So far, we have seen an overall picture of what has been done before. This section will justify why our research direction is worth pursuing.

- **Duality of diffuse and specular reflectances.** At the time we started shaping our research direction, most of the SfP works assumed either diffuse or specular reflection which, we realised, is not how real-world objects behave. With that being said, a few exceptions did exist. For instance, Smith et al. [74] and Cui et al. [14] allow both types of reflections but have to classify which one is dominant within the particular pixel, and Taamazyan et al. [77] approximate the degree of polarisation to obtain the expression for diffuse radiance which co-exists with specular radiance. These methods rely either on an approximation or dominant reflection labelling, whose accuracy is sacrificed. When properly seeing the behavior of light, the reflection comes from both diffuse and specular phenomena. Hence, to achieve a more accurate 3D reconstruction, we need a model where both types of reflections co-exist.

- **Coordinate-based scene representation with polarimetric cues.** Even though NeRF is designed to synthesise a novel view, NeRF is also capable of 3D reconstruction. However, none of the follow-up works tried to incorporate polarimetric cues into the pipeline - not until recently. This provides an inspiration for us to include polarisation information in the framework and Chapter 3 serves as our first attempt to do so.

- **Less processed input assumes less about inferred geometry - thus potentially higher accuracy.** Most SfP techniques work with Stokes vectors, if not DoLP and AoLP (Angle of Linear Polarisation).

This requires a full demosaicing which is either done inside the camera software or manually calculated under an assumption. Bilinear interpolation is a common method for demosaicing, which assumes a linear transition between pixels. This effect would be inherited to the geometry, smoothing high-frequency details. Therefore, directly working with what the camera actually captured, i.e. raw image, could avoid this pitfall. RawNeRF [52] is a good example that deals with raw RGB images (instead of demosaiced images). Since the polarisation filter array is 4-times larger than a conventional color filter array, the benefit of working directly with raw images for SfP methods could be significantly higher.

- **High-quality 3D geometry from passive capture in real time.** Since inverse rendering is generally an ill-posed problem with various unknowns (e.g. object geometry, material properties and lighting environment), unless imposing a strong assumption about the scene or capturing the object with special technique, the estimated geometry is usually in poor conditions - providing an unmeaningful result, at worst, when a key assumption is broken or being full of artifacts due to unrealistic assumptions. For example, with light stage methods [20, 49, 24] which usually require many lighting patterns for an object i.e. special capture, the environmental lighting becomes solidly defined, material properties could be analysed from different incident lighting and thus a high-quality geometry could be estimated. However, as requiring different lighting patterns, these methods fail to capture the subject of interest in real time. There would be no problem if the subject was a static object, but unfortunately, this group of methods often has a movable human as the subject of interest. Polarisation is a good tool to distinguish different lighting states thus having a potential to capture a moving scene/object within a single shot. Therefore, real-time-possible polarimetric capture in the context of 3D reconstruction is worth exploring to ease human capturing process (particularly in film production).

2.6 Polarimetric dataset review

In this section, we provide an overview of available datasets. Since our work solely relies on polarisation images, *i.e.* our method could not perform on traditional color images, we will exclude RGB datasets in the following discussion.

Publicly available polarisation image datasets have seen a limited presence due to 3 main reasons. An obvious one is that polarisation images have no direct meaning to general public nevertheless being applied in end-user applications such as 3D glasses in cinemas and polaroid eyewear. Therefore, researchers could not easily bootstrap polarisation work in the same way as traditional vision research employing RGB images on internet. Secondly, polarisation is only a small sub-category in a larger research theme. Geometry extraction, for example, could be done by various techniques such as shape from shading, multi-view stereo, photometric stereo, shape from polarisation or even LiDAR sensing technology. This effectively reduces the need for polarisation in research landscape, thus lowering the number of available datasets. Requiring special equipment is another reason why we have seen a limited adoption in both academic and industry. Without polarisation camera, to obtain enough information to fit a sinusoidal curve, a scene has to be captured at least 3 times by applying polariser on top of a traditional camera, thus relatively increasing a capture time when compared to non-polarisation methods.

Beyond the availability of datasets, it is noted that different datasets are collected to fit different research methodologies. Therefore, varied specifications are observed. There is a group of methods [3, 30, 42] which leverage a single view - restricting multi-view techniques like ours to exploit their collected data. Although datasets provided by in-the-wild methods [42] are rich in scene variety, some datasets [30] require a specific setup in capturing process while some [3] provide a structured variety aiming at a certain number of object orientations and lighting conditions in the scene. RMVP3D is the first real-world multi-view polarised image dataset with ground-truth shape, nonetheless providing limited viewpoints, only suiting for sparse-view methods [29]. NeISF [43] is a multi-view method which makes their dataset

publicly available. However, due to a security measure, the access right seems to be strictly given. Because of similar assumptions and methodology, PANDORA [19] is the dataset we use widely in this thesis, giving a good mix of object shapes and materials.

The PANDORA dataset consists of 2 sub-categories: rendered polarimetric dataset and real polarimetric dataset. Bust and globe datasets, belonged to rendered datasets, are generated using Mitsuba 2 [63] by applying pBRDF [4] on both objects and then rendering 45 views each under a realistic lighting environment. While the globe represents a simple geometry of a sphere, the bust is full of facial features inheriting a complex structure of a human face. Due to being rendered, we have an access to ground-truth color and geometry, and are therefore being able to conduct quantitative evaluation as reported in Table 3.1, 4.1 and 5.1. On the other hand, vase, owl and gnome datasets are real datasets. Each real dataset, composing of 35 views horizontally rotated around the object, is captured under unstructured lighting conditions *e.g.* office hall. The real datasets well represent the real-world complexity composing of different materials such as ceramics, glass, resin and plastic. Even though both real and rendered datasets do not cover every possible angle of the object, there is an intersecting area between each frame, satisfying our requirements discussed in Section 4.2.3.

3 Neural Polarised Radiance Fields

> Every brilliant experiment, like every great work of art, starts with an act of imagination.
>
> *Jonah Lehrer*

In 2020, just a year before I started my PhD, a scene-representation paper came out and quickly became a classic recipe for thousands of follow-up works. The quality of reproduced scene, competitive metrics and the scene versatility - how could someone in vision/graphics ignore this piece of work? Neural Radiance Fields or NeRF [55] represents a scene using 5D coordinates - spatial location (x, y, z) and viewing direction (θ, ϕ) - achieving state-of-the-art results at that time.

This chapter aims to reconstruct an object in the scene. We will start from NeRF, gradually improve different aspects of the pipeline, toss a hypothetical question and then attempt to answer it.

3.1 What is NeRF, really?

Since NeRF will be thoroughly referred to in this chapter and be the foundation in the following chapters, we will begin with explaining what NeRF is to refresh its concept in readers' mind.

NeRF synthesises realistic renderings of scenes by encoding volumetric density σ and colour $\mathbf{c}_{rgb}$ of a scene within the weights of a coordinate-based multi-layer perceptron (MLP). In particular, the MLP network $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}_{rgb}, \sigma)$ maps 3D location $\mathbf{x} = (x, y, z)$ and 2D viewing direction (θ, ϕ) which

is expressed in the form of 3D Cartesian unit vector $\mathbf{d}$, to its corresponding volume density and direction emitted colour. NeRF achieves multiview consistency by restricting the network to predict the volume density as a function of only location $\sigma(\mathbf{x})$ while allowing the predicted colour to be a function of both location and viewing direction $\mathbf{c}_{\text{rgb}}(\mathbf{x}, \mathbf{d})$.

To composite these values into an image, NeRF applies volume rendering [31] to the colour at each point on the ray. The resulting colour $\hat{C}(\mathbf{r})$ of camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with near and far bounds t_n and t_f is:

$$\hat{C}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}_{\text{rgb}}(\mathbf{r}(t), \mathbf{d})dt, \text{ where } T(t) = \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds)). \tag{3.1}$$

The above integral in Equation 3.1 could be approximated as the sum of stratified samples:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i (\mathbf{c}_{\text{rgb}})_i, \text{ where } T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j) \tag{3.2}$$

and volume density gets reduced to alpha compositing with alpha values $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$. The parameter δ represents the quantised distance between adjacent samples.

Nonetheless, volume rendering alone could not yield a clear image with high-frequency details as the deep networks are biased towards learning low-frequency function [65]. NeRF employs positional encoding:

$$\gamma(p) = (\sin(2^0 \pi p), \cos(2^0 \pi p), ..., \sin(2^{L-1} \pi p), \cos(2^{L-1} \pi p)), \tag{3.3}$$

where L is an arbitrary integer, being respectively chosen to be 10 and 4 for 3D coordinate $\mathbf{x}$ and viewing direction $\mathbf{d}$ in NeRF.

During optimisation, the loss is calculated as the total squared error between rendered pixel colour $\hat{C}$ from the sampled ray and true pixel colour C in the training images:

$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} (\hat{C}(\mathbf{r}) - C(\mathbf{r}))^2, \tag{3.4}$$

where $\mathcal{R}$ is the set of rays in each batch.

'How does it perform?', you may wonder. The Figure 3.1 shows the comparison between NeRF and prior methods. NeRF manages to produce reasonably high quality scenes. The next question is, once we get a plausible scene reproduction, do we automatically get the geometry of the object inside the scene as a by-product? The short answer is 'NO'.

3.2 Modification for surface reconstruction

As NeRF can reproduce the whole scene including the object we want, why cannot we directly extract object geometry out of the reproduction? NeRF only aims to synthesize novel views. While being able to reproduce the object geometry, NeRF inherits a geometric error during the volume rendering process (Equation 3.1).

On a different note, there is another line of works which are specifically designed to reconstruct 3D surfaces. Implicit Differentiable Renderer or IDR [89], for example, represents a surface as the intersection points between a ray and the surface. Due to inefficient back propagation from only one single point along the ray, the method struggles with objects that have complex structures or abrupt depth changes.

NeuS or neural rendering scheme [84] comes in to bridge the gap of geometric bias while allowing the back propagation from all points on the ray. Specifically, NeuS uses Signed Distance Function (SDF) to represent a surface and modify standard volume rendering to accurately learn the surface representation. Figure 3.2 shows 3D reconstructions from 3 methods. As can be seen, IDR struggles to produce a meaningful result in the hole region and NeRF gives a noisy result due to geometric bias, while NeuS produces a high quality result.

3.2.1 SDF in surface reconstruction

Before going to see how NeuS transforms the traditional volume rendering, we want to arm our readers with SDF concept and its application in surface reconstruction.

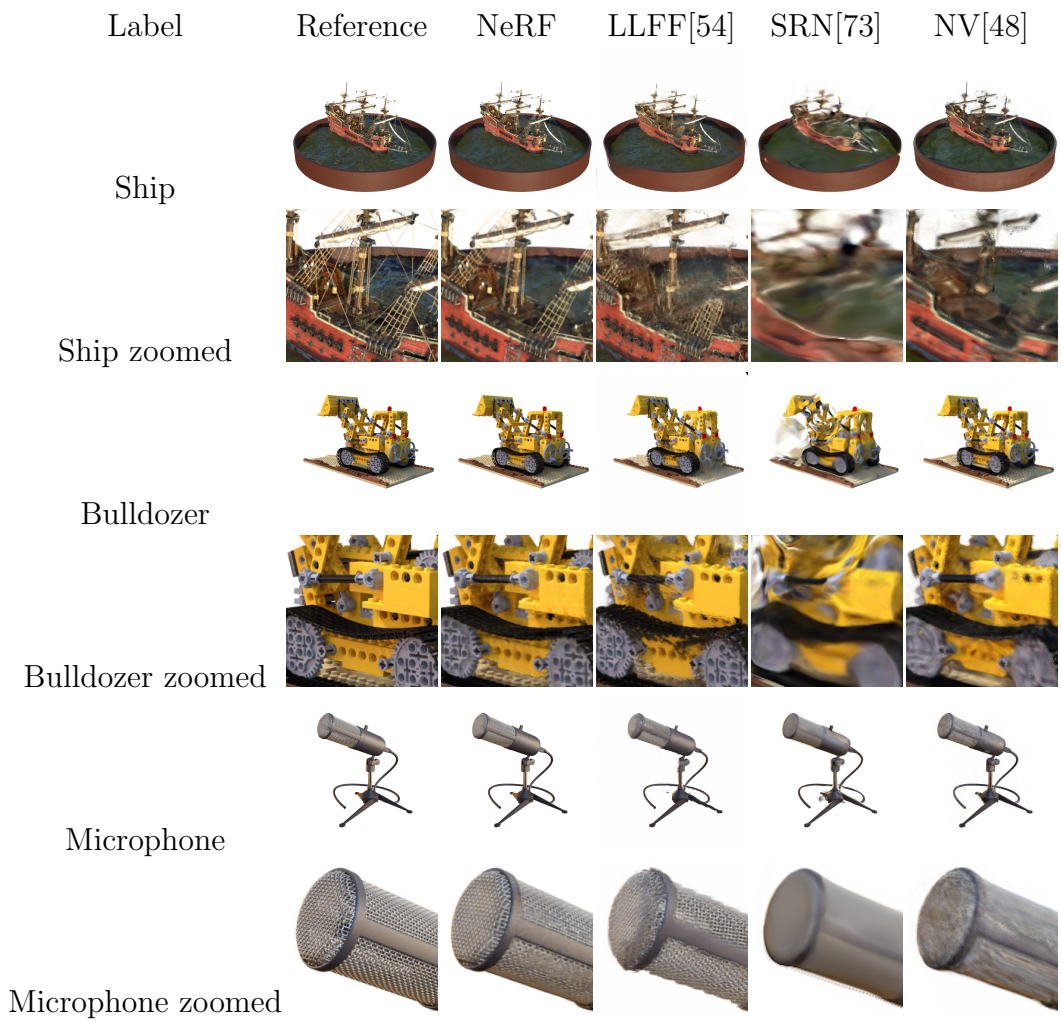
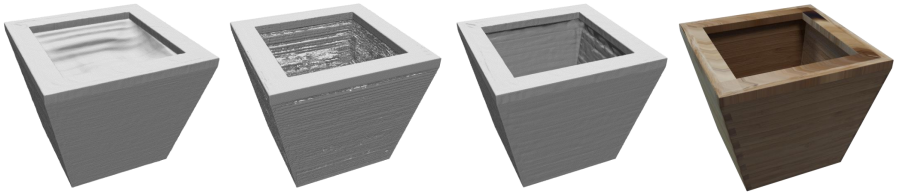


Figure 3.1. **Rendering comparisons** among NeRF and prior methods. NeRF is versatile and could reproduce the appearance of different object such as the ship body, bulldozer lego structure, and microphone grille. LLFF, while managing to produce the rough object appearance, could not recover the full length of rope net clung to the ship body and produces a broken chain in bulldozer. SRN gives blurry and distorted renderings in every case. NV fails to capture the details inside the bulldozer wheel and blurs all the fine structure in microphone grille. Images from [55].



IDR [89] NeRF [55] NeuS [84] Reference

Figure 3.2. **NeuS improvement over the surface- and volumetric methods.** NeuS is able to reproduce a clean surface of the planter whose depth change abruptly. Images from [84].

Signed distance function, or sometimes called signed distance field, is defined as the shortest distance of a point $\mathbf{x}$ to a surface $\mathbf{S}$, with the sign determined by whether or not the point $\mathbf{x}$ is inside the surface $\mathbf{S}$. The sign could be arbitrarily assigned by author/context. For instance, Figure 3.3 illustrates a SDF of a circle where outside distance is chosen to be positive. Mathematically, $f(\mathbf{x}) > 0$ outside the circle, $f(\mathbf{x}) < 0$ inside the circle, and $f(\mathbf{x}) = 0$ on the circle. Thus, by having all the points $\mathbf{x}$ that make $f(\mathbf{x}) = 0$, we simply have the circle.

SDF property. By definition, the gradient vector of a function is the vector pointing in the direction that the function fastest increases. So, in the case of SDF, its gradient is the normal vector of the local surface. As we move along the normal vector 1 infinitesimal unit, the SDF value also increases/decreases the same 1 unit. Therefore, the magnitude of SDF gradient is always 1 $|\nabla f(\mathbf{x})| = 1$ or being known as Eikonal Equation in mathematics.

3.2.2 Volume-rendering revision

What is it that causes the geometric error in traditional volume rendering? We are now going back to volume rendering equation to investigate this issue. We can simplify Equation 3.1 by rewriting the product of accumulated transmittance T and volume density σ into the function called weight w:

$$w(t) = T(t)\sigma(t), \tag{3.5}$$

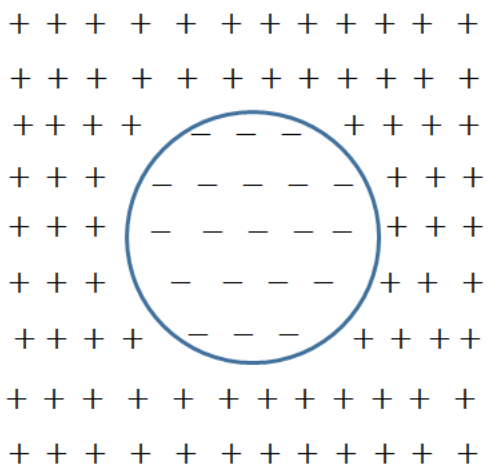


Figure 3.3. **Signed Distance Function (SDF)** where the distance inside the circle is defined as being negative.

and hence the rendering equation becomes:

$$\hat{C}(\mathbf{r}) = \int_0^\infty w(t)\mathbf{c}_{\text{rgb}}(\mathbf{r}(t), \mathbf{d})dt, \tag{3.6}$$

when we consider the whole space in front of the camera. Assuming we have an opaque object that is put in a transparent medium, our weight function w would ideally be zero everywhere except at the surface where the peak is, but in reality, one can show that the weight function determined by transmittance and volume density (as in Equation 3.5) has the peak at a point before the ray hits the object's surface - hence introducing the bias.

The bias could be avoided by establishing the right relationship between the the output colors and SDF. In other words, we have to derive a new 'weight function' based on the SDF of the scene. Let's start from listing the desired properties of the weight function:

- **Unbiased.** At a medium intersection point $t = t^*$, the new weight function hits the local maxima. Mathematically we have $\frac{dw}{dt}(t^*) = 0$ and $\frac{d^2w}{dt^2}(t^*) < 0$. This guarantees that the point where the camera ray hits the zero-level set of SDF, contributes most to the pixel value;

- **Occlusion-aware.** Given any two points on the ray t_0 and t_1 such that $f(t_0) = f(t_1)$, $w(t_0) > 0$, $w(t_1) > 0$ and $t_0 < t_1$, the weight

functions must obey $w(t_0) > w(t_1)$. This relationship ensures that the rendering process gives output color dominated by the surface nearest to the camera when the camera ray passes multiple surfaces.

While there are many possible functions that could satisfy the above requirements, we will follow NeuS choice of logistic density distribution $\phi_s(x) = se^{-sx}/(1+e^{-sx})^2$, or commonly known as the derivative of the Sigmoid function $\Phi_s(x) = (1+e^{-sx})^{-1}$ *i.e.* $\phi_s(x) = \Phi_s'(x)$, to avoid unnecessary complexity; the full proof can be found in NeuS supplementary material. So one of the corrected functions, we are looking for, is:

$$\omega(t) = \tau(t)\rho(t), \tag{3.7}$$

where we use opaque density function $\rho(t)$ instead of volume density $\sigma(t)$ in the standard volume rendering, and consequently the accumulated transmittance $\tau(t)$ is determined by:

$$\tau(t) = \exp(-\int_0^t \rho(u)du), \tag{3.8}$$

in which $\rho(t)$ is defined as:

$$\rho(t) = \max(\frac{-\frac{d\Phi_s}{dt}(f(\mathbf{r}(t)))}{\Phi_s(f(\mathbf{r}(t)))}, 0). \tag{3.9}$$

We modify Equation 3.2 accordingly to work with the newly proposed weight function $\omega(t)$. Specifically, the accumulated transmittance is approximated as

$$\tau_i = \Pi_{j=1}^{i-1}(1-\alpha_j), \tag{3.10}$$

and opacity values become:

$$(\alpha_i)_{\text{new}} = \max(\frac{\Phi_s(f(\mathbf{r}(t_i))) - \Phi_s(f(\mathbf{r}(t_{i+1})))}{\Phi_s(f(\mathbf{r}(t_i)))}, 0). \tag{3.11}$$

To train NeuS, we optimise the neural networks by minimising 3 losses:

$$\mathcal{L} = \mathcal{L}_{\text{color}} + \lambda\mathcal{L}_{\text{reg}} + \beta\mathcal{L}_{\text{mask}}. \tag{3.12}$$

The color loss $\mathcal{L}_{\text{color}}$ is the mean value of color L1:

$$\mathcal{L}_{\text{color}} = |\hat{C} - C|, \tag{3.13}$$

the regularising term helps ensuring that the magnitude of SDF gradient is 1 (as we have seen in section 3.2.1):

$$\mathcal{L}_{\text{reg}} = (|\nabla f(\mathbf{x})| - 1)^2, \tag{3.14}$$

and lastly, mask loss signifies the networks where the object is:

$$\mathcal{L}_{\text{mask}} = \text{BCE}(M, \hat{O}), \tag{3.15}$$

where BCE is the binary cross entropy loss, $M \in \{0, 1\}$ is binary mask, and $\hat{O} = \sum_{i=1}^{n} \tau_i(\alpha_i)_{\text{new}}$ is the sum of new weights.

3.3 It's all about efficiency!

So far, we have not talked about the time taken to train the networks. Using a large batch size of 4096 rays, NeRF takes around 100-300k iterations to converge on a high performance GPU (NVIDIA V100) or being equivalent to 1-2 days. With a smaller batch size of 512 rays, NeuS needs around 15 hours to train 300k iterations for the networks to converge on NVIDIA RTX2080Ti GPU. The time taken to train these 2 models is considerably long. Fine tuning a set of parameters could easily take up to a week, given a limited resource an individual or a small organisation has. Thus the slow training speed has inspired many researchers to come up with the model that has a small training time.

One of the bottlenecks in the NeRF framework is coarse-to-fine networks which first predict coarse ray intervals and then fine ray intervals. While this architecture helps navigating where the object surface is, there is a redundancy to back propagate to both coarse and fine networks. Instead, Mip-NeRF 360 [6], employs only one colour network dubbed 'NeRF MLP' with an addition weight network called 'proposal MLP', and altogether streamlines the gradient propagation to only flow back to NeRF MLP, rather than

coarse and fine networks as done in NeRF. The process of having proposal MLP to guide NeRF MLP the range of possible surfaces, could be thought of as a kind of 'knowledge distillation'. It is worth noting that there is nothing being special about proposal MLP - both MLPs are randomly initialized. This modification simplifies the output of 'teacher network' and thus we can reduce its size, further boosting the training speed in addition to effective gradient flow.

Instant neural graphic primitives [62], or instant NGP in short, reduces the amount of time to train NeRF from days to seconds and won the best inventions of 2022 given by TIME Magazine. That is serveral orders of magnitude faster than the original NeRF - what is the technique they are using? It is hash encoding.

Hashing is a method where the data is converted to hash by a hashing algorithm. Different algorithm has their own use cases such as SHA-2 usually being used in security applications, while the one used by instant NGP is mainly to shrink the number of parameters - hence smaller network required. Nonetheless there is a possibility that 2 values give the same hashing value or so called hash collision, which is addressed by multi-resolution hash table. Instant NGP could achieve a similar image quality, measured with PSNR, by using 8x smaller amount of time using to train NeRF, and 20x fewer parameters used to train dense grid [13, 28].

3.4 What if we add polarisation?

So now, we have an efficient scene representation which are unbiased and occlusion-aware. We modify the weight function to accurately learn object surface via SDF. We use proposal network as a teacher network to guide color network. We encode network input with hash encoding, reducing the number of parameters that are needed to train. What else could we do to improve 3D object reconstruction?

There is a whole range of Shape-from-Polarisation (SfP) literature showing the correlation between surface normal and radiance reflected from surface. We wonder if polarisation could fit the NeRF-like achitechture we have been dealing with.

3.4.1 Data acquisition

Typically, polarisation-based reconstruction would work in either Stokes-vector space:

$$S = [S_0, S_1, S_2, S_3], \tag{3.16}$$
$$S_0 =< E_x^2 > + < E_y^2 >, \tag{3.17}$$
$$S_1 =< E_x^2 > - < E_y^2 >, \tag{3.18}$$
$$S_2 =< E_a^2 > - < E_b^2 >, \tag{3.19}$$
$$S_3 =< E_r^2 > - < E_l^2 >, \tag{3.20}$$

where $< E >$ represents the expected value of electric field in the Cartesian basis $(\hat{x}, \hat{y})$, Cartesian basis rotated anti-clockwise by 45^o $(\hat{a}, \hat{b})$, and circular basis $(\hat{l}, \hat{r})$; or a group of unpolarized intensity I_{un}, degree of linear polarisation (DoLP) ρ, and angle of linear polarisation (AoLP) ϕ:

$$I_{un} = S_0, \tag{3.21}$$
$$\rho = \frac{\sqrt{S_1^2 + S_2^2}}{S_0}, \tag{3.22}$$
$$\phi = \frac{1}{2} \arctan \frac{S_2}{S_1}. \tag{3.23}$$

While these parameters could be obtained from camera or derived from raw image, they are processed by camera software or distorted by algorithm (e.g. bilinear interpolation). Instead of relying on secondary data, we will follow RawNeRF approach [53] using raw data from camera sensor.

Traditionally, to acquire polarisation information, a polarising filter is put on top of the camera before a standard capture. This process has to be repeated to acquire the information at different filter angles, thus hindering a fast acquisition. Recently, commercial-grade polarisation camera becomes available as another off-the-shelf equipment. The mosaic pattern is placed on top of camera sensor and the polarisation data could be captured within one single shot. In a way, this could be thought as gaining extra polarisa-

tion information at the expense of spatial resolution. We will use the latter method for the sake of our convenience. Not only does this choice of image capturing shrinks our acquisition time, but the method also makes capturing a moving object possible with multiple cameras.

3.4.2 NeRF on polarised images

Now the key question is - how do we modify NeRF to work with the input from polarisation camera? What is the assumption in NeRF that becomes invalid in this new setting?

Assumption: NeRF has assumed 3 colour channels (RGB) and perfect spatial resolution *i.e.* the result of demosaicing, when rendering an image. The colour at each point in the object/medium and its transparency/opacity, which are physical property, stay unchanged. Even if the world has millions of color spaces, the NeRF framework will also work perfectly fine as the volume rendering principle is the same. In fact, our world does have an infinity set of possible color spaces (the wavelength of light is continuous), only we as human beings limit them to standard RGB due to our inability to distinguish fine colors. There is one caveat though: light at different wavelength also have different penetration (this is why an x-ray works). Similar to NeRF and other following works, we will not take that into account and throughout this thesis we will assume that all colour channels have the same penetrating property.

Before moving to the NeRF modification, we need to understand the raw image being captured by polarisation camera. Being directly linked to material property, the polarisation cue alone is versatile to various applications such as scratch detection in transparent material while our application, 3D reconstruction, is better to have both polarisation and color information. Thus we employ a color polarised filter shown on the right of Figure 3.4. So, instead of getting the standard Bayer pattern RGGB as shown on Figure 3.5, we gain extra polarisation information. Inside the red pixel group, for example, there are 4 polarisation filters. The pattern is camera-specific and ours, Triton camera from LUCID, is 90^o, 45^o, 135^o and 0^o. We will represent the combination of color channel and polarisation orientation as 12 channels including R0, R45, R90, R135, G0, G45, G90, G135, B0, B45, B90 and B135.

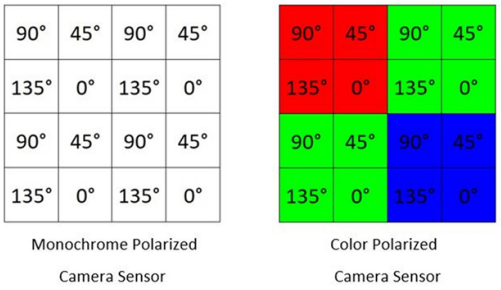


Figure 3.4. **Mosaic pattern.** Monochrome filter (left) and color filter (right).

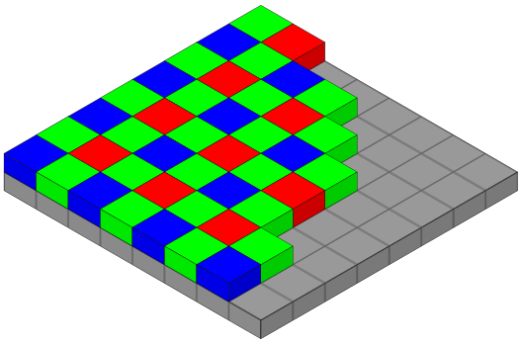


Figure 3.5. **Bayer pattern** is a color filter array (CFA) for arranging RGB color filters on a square grid of photosensors.

Implementation: We extend NeRF to render the 12 channels, filter out 11 channels that do not correspond to the camera filter, and compare the rendering with ground truth pixel. We incorporate the unbiased rendering scheme (as done in NeuS), proposal network and hash encoding in our pipeline, with the L1 color loss (Equation 3.13). The Figure 3.6 shows the overall architecture of our framework.

3.4.3 Results

Training details: We implement our method on Nerfstudio [79] where we inherit most of the hyperparameters. We test our method on both synthetic and real datasets. Due to the difference in spatial image dimension, we apply 2 and 6 layers of color network, with 256 hidden units for each layer, on synthetic and real datasets respectively. For the activation function, ReLU is applied throughout, except the final layer where we use exponential function

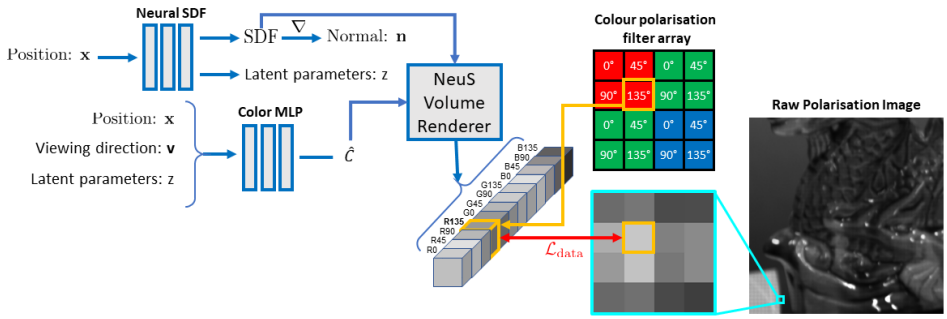


Figure 3.6. **NeRF in 12 channels.** We use a neural signed distance function to represent the surface and derive the surface normal via differentiation. Another MLP learn polarised radiance as a blackbox where no physical knowledge is given to the network. We select the appropriate channel for each measured pixel and compute a data loss. While it may not be obvious in the diagram, we want to emphasise the following link: the latent parameters obtained from Neural SDF Network are used as an input of color network. The link is omitted from the diagram for a compact visualisation.

to reflect the nature of raw image input. The geometric network is set to have 256 units per layer, and 2 layers in total; softplus is used as the activation function.

Figures 3.7, 3.8, 3.9 and 3.10 illustrate the rendering results of the reflectance and normal estimation whilst Table 3.1 shows a quantitative evaluation of synthetic datasets. While our method could reproduce the overall 3D structure of bust and its corresponding appearance, our method struggles with noise hence leading to to rough surface of nose and surrounded area of neck. The reconstructed geometry of globe is full of artifacts affecting the rendering. When applied on real datasets, our method severely suffers from incomplete reconstruction, lacking a few object parts *e.g.* gnome's left leg and mug handle. In the case where the model is able to reconstruct the whole object, we observe that the model fails to distinguish between texture and geometric variation *e.g.* owl geometry. Nevertheless, as working directly on raw images, our model is able to recreate the fine geometric details which never appear on PANDORA reconstruction.

The last obersavation made on the rendering results gives us an affirmation that our method could preserve object geometry, in some cases, even

		Diffuse		Specular		Mixed		Normals
Scene	Method	PSNR ↑ (dB)	SSIM ↑	PSNR ↑ (dB)	SSIM ↑	PSNR ↑ (dB)	SSIM ↑	MAE ↓ (°)
bust	NeuralPIL*	23.90	0.87	18.04	0.87	26.71	0.87	15.36
	PhySG*	22.64	0.94	23.00	0.94	19.94	0.72	9.81
	PANDORA†	25.82	0.81	22.96	0.75	22.79	0.79	3.91
	NeuS*	N/A	N/A	N/A	N/A	28.09	0.85	8.53
	12-channel NeRF	N/A	N/A	N/A	N/A	30.08	0.95	4.25
globe	NeuralPIL*	13.09	0.55	12.92	0.55	20.04	0.66	38.73
	PhySG*	21.76	0.76	18.90	0.76	17.93	0.70	8.42
	PANDORA†	24.33	0.77	22.70	0.89	21.76	0.81	1.41
	NeuS*	N/A	N/A	N/A	N/A	23.57	0.81	3.72
	12-channel NeRF	N/A	N/A	N/A	N/A	18.88	0.82	1.95

Table 3.1. **Quantitative evaluation** on PANDORA [19] synthetic image benchmark. * = method is given access to ground truth demosaiced RGB images. † = method is given access ground truth demosaiced 12 channel RGB/polarisation images.

better than PANDORA which explicitly utilises the polarisation cues in the reconstructing pipeline.

In this chapter, we have seen the gradual development of the coordinate-based neural networks which are applied to scene reconstruction and proposed how we could introduce polarimetric cues into the pipeline; in the next chapter, we will show the theory underpinning polarisation principles and how we put the related knowledge together into our framework.

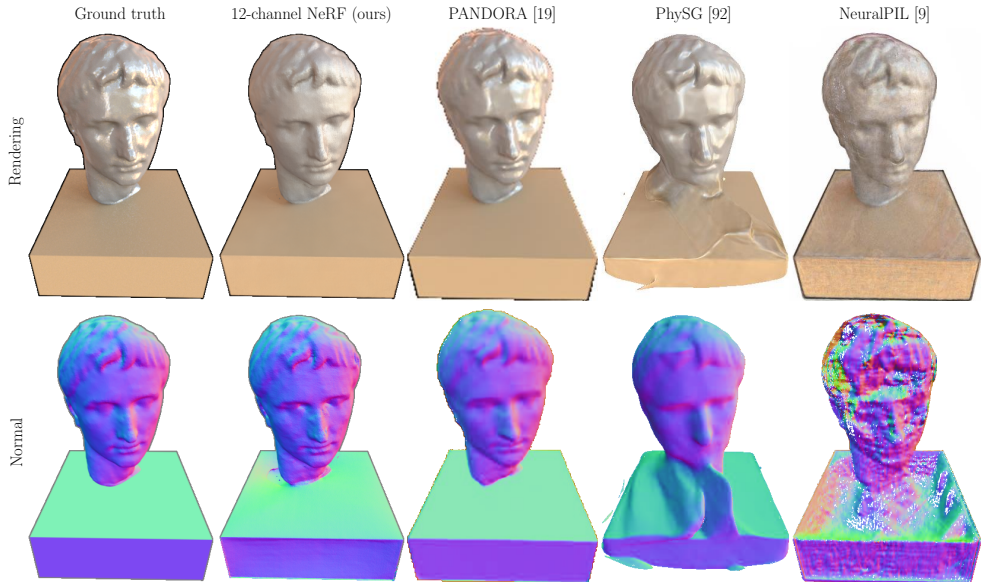


Figure 3.7. **Comparison of reflectance rendering and surface normals with baselines on (synthetic) bust dataset.** Our method performs reasonably well, giving the right overall geometry and capturing the bust appearance. Nonetheless, compared to PANDORA which explicitly utilises the polarisation information, the reconstructed normal map is prone to noise especially around the nose and neck regions, and the rendering image lacks specularities at cheek and forehead.

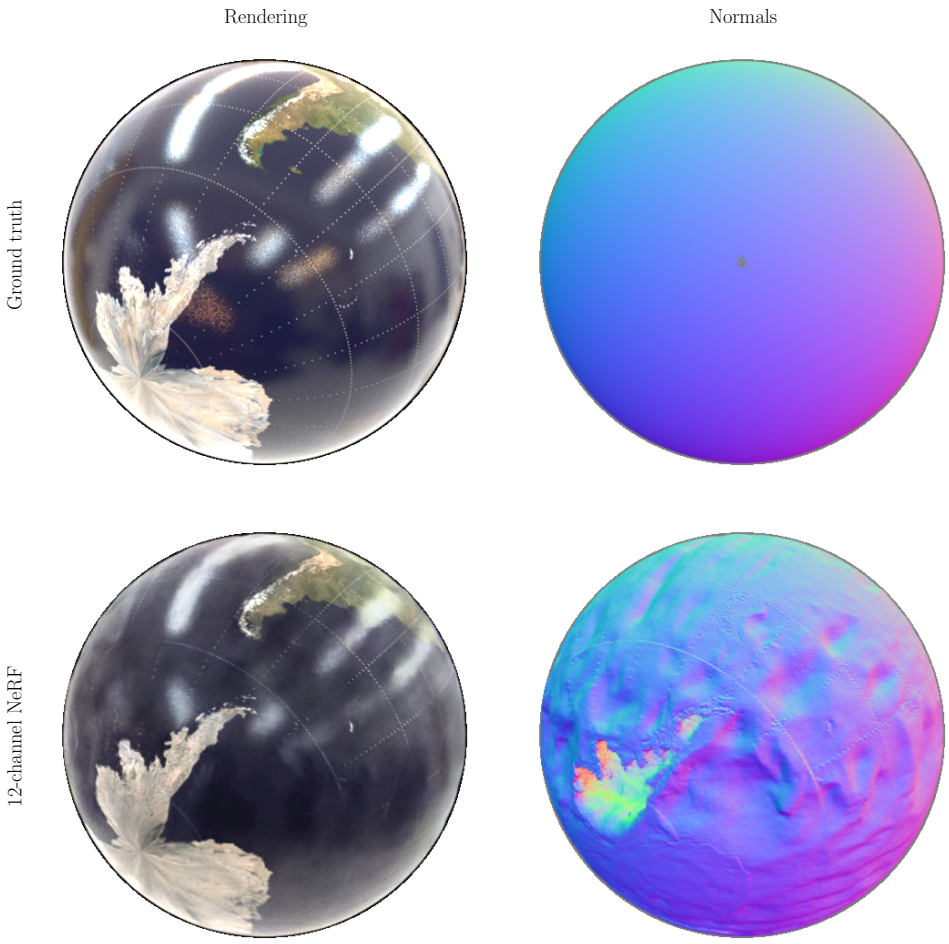


Figure 3.8. **Reflectance rendering and geometry estimation on (synthetic) globe dataset.** While giving a reasonable result on color rendering, our method shows foggy artifacts. The model struggles to produce a smooth surface of the sphere.

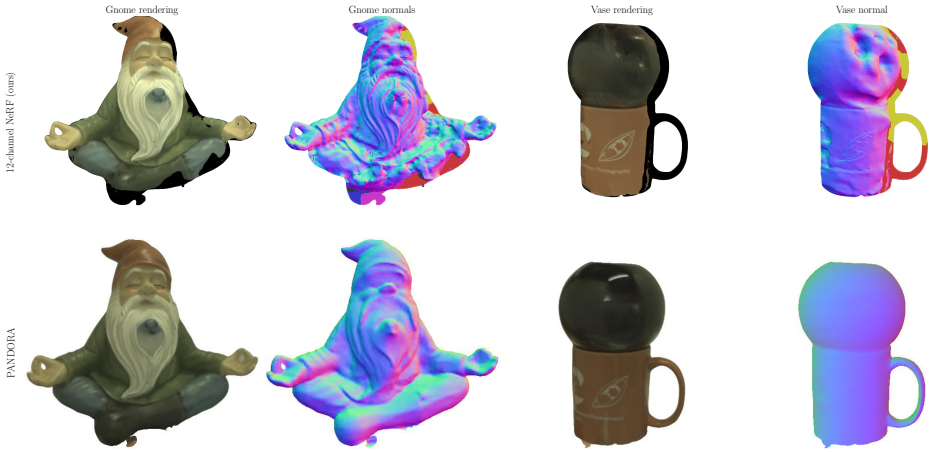


Figure 3.9. **Reflectance rendering and geometry estimation on (real-world) gnome and vase datasets.** Our method could not reconstruct the whole object and produce artifacts on normal maps. However, it is important to note the high-frequency geometric details on gnome dataset, which don't appear on reconstruction from PANDORA.

Figure 3.10. **More 12-channel results of (real-world) Shakespeare and owl data.**

4 Polarised Neural Radiance Fields

> Everything should be made as simple as possible, but not simpler.
>
> *Albert Einstein*

In the previous chapter, Neural Polarised Radiance Fields, we leverage the multi-view consistency and the power of neural networks to predict polarised radiance. The radiance is then fitted to the ground truth images captured directly from camera sensor, *i.e.* raw images. By independently treating polarisation information captured with differently oriented filter (as we referred to 12 channels), we provide no physical insight to the networks, how each channel correlates to the other. In a way, we just ask the networks to guess the object geometry from extra polarisation information (in addition to color) and hope the networks would understand the physics.

Instead of relying on implicit understanding of physics, in this chapter, we will explicitly inject the physical insight into the pipeline. This problem is known as Shape from Polarisation.

Polarisation state of light describes the direction in which the light oscillates (light is a transverse wave by nature). The key idea behind SfP is that the state of light changes after being reflected at a surface. A simple way to observe the polarisation state is to put a polarising filter between our eye and the light source, rotate the filter, and the image we see would become dark and bright assuming the light source gives (partially) polarised light. If we plot the intensity against the polariser angle, we will obtain a sinusoid (see Figure 4.1); the maximum intensity occurs when the oscillating light passes

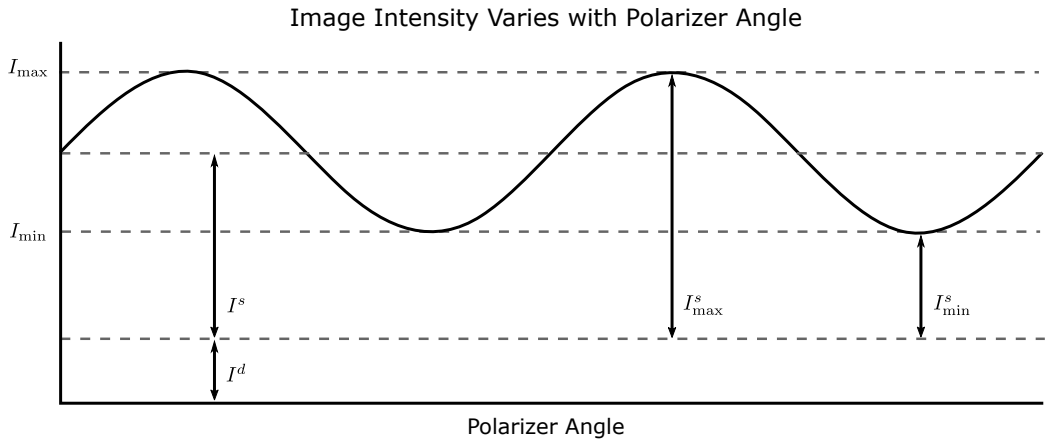


Figure 4.1. **Intensity variation** as the polarisation filer (polariser) is rotated, assuming diffuse radiance is unpolarised. Image from [77]

the filter most while the minimum intensity does when the light passes least. Mathematically, we have:

$$I(\vartheta) = \frac{I_{max} + I_{min}}{2} + \frac{I_{max} - I_{min}}{2} \cos(2(\vartheta - \phi)), \tag{4.1}$$

where I_{max}, I_{min} are the maximum and minimum intensities respectively, ϑ is the polariser angle and ϕ is the phase angle which can be derived from the azimuth angle φ:

$$\phi = \varphi \mod \pi. \tag{4.2}$$

4.1 Diffuse vs specular radiances

Actually, when concerning the underlying physical phenomena, the above intensity (Equation 4.1) is the combination of diffuse and specular radiances. When light hits a surface, specular reflection is the effect of the light directly bouncing off the surface into the same medium; whereas diffuse reflection occurs from the light partially transmitting into another medium, scattering inside that medium before transmitting back to the medium where the light is coming from (see Figure 4.2 for the illustration). In daily life, mirror and water surface are a perfect example for specular surface while diffuse surface could be represented by clothing and concrete road.

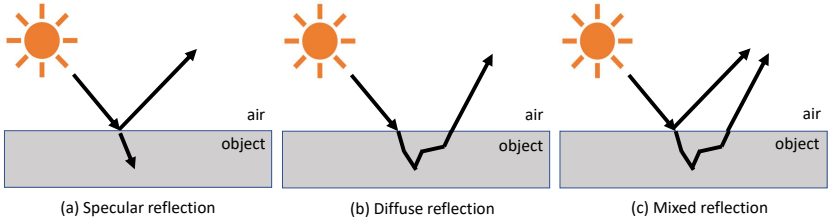


Figure 4.2. **Reflection mechanism.** Unpolarised light incident on a surface can be specularly reflected (a), thus acquiring partial polarisation, in agreement with the Fresnel Equations; (b) diffuse reflection is the result of light scattering multiple times inside the object before being refracted at the surface, again acquiring partial polarisation due to transmission, out of phase with specular polarisation; (c) mixed reflection, the model used in this work, is the combination of both types of reflection.

While the Equation 4.1 describes the behaviour of polarised light, to fully understand this physical phenomenon, we need to trace the history back to 19th century when Augustin-Jean Fresnel derived reflectivity and transmissivity - or usually being called Fresnel Equations. To keep the thesis concise, we will not repeat the whole derivation of polarisation model [1] here but will only quote the result. The diffuse radiance that arises from internal scattering followed by transmission out of the surface, could be described by diffuse polarisation model:

$$I_{\vartheta}^{\text{diffuse}} = I_d + I_d \rho_d \cos(2\vartheta - 2\phi), \tag{4.3}$$

where I_d is the unpolarised RGB diffuse radiance in the direction of the viewer, and ρ_d is the diffuse degree of polarisation being equal to:

$$\rho_d = \frac{(\sin(\theta))^2(\eta - \frac{1}{\eta})^2}{4\cos(\theta)\sqrt{\eta^2 - (\sin(\theta))^2} - (\sin(\theta))^2(\eta + \frac{1}{\eta})^2 + 2\eta^2 + 2}, \tag{4.4}$$

where θ is zenith angle *i.e.* the angle between the normal direction and the light ray, and η represents the refractive index of the material underneath the surface. A similar expression exists for specular reflection:

$$I_{\vartheta}^{\text{specular}} = I_s + I_s \rho_s \cos(2\vartheta - 2\phi + \pi), \tag{4.5}$$

where we replace ρ_d with ρ_s and add π inside the cosine to account for the radiance being out of phase with diffuse radiance; the specular degree of polarisation ρ_s could be calculated as follows:

$$\rho_s = \frac{2(\sin(\theta))^2 \cos(\theta) \sqrt{\eta^2 - (\sin(\theta))^2}}{\eta^2 - (\sin(\theta))^2 - \eta^2(\sin(\theta))^2 + 2(\sin(\theta))^4}. \tag{4.6}$$

4.2 SfP ambiguities

Now we are equipped with the basic polarisation knowledge showing how the surface orientation relates to the measurement from the camera; the next step is to measure and calculate. Is it that simple?

4.2.1 Convex or concave - we cannot say!

We know that the relationship between measured intensity and the rotation angle of the polariser is sinusoidal. The simplest setup is to align object, polariser and camera together; and keep rotating the polariser until we find the darkest/brightest image. Assuming we know the object nature (diffuse or specular) by observation and attempt for darkest image, to minimise the intensity in the polarisation model, the obvious solution is to get the cosine value being equal to -1 as other variables are positive.

The problem immediately arises. As we turn the full round of polariser, we would observe 2 darkest images. Which one is the one we want? How do we distinguish these 2 images?

This phenomenon comes from the fact that polariser allows electric field in the specified direction passing through. For example, having vertical filter on the polariser would allow both upwards and downwards electric fields to pass through. This is equivalent to the factor of 2 in front of ϑ and ϕ in the polarisation models (Equations 4.3, 4.5). As a result of this, distinguishing between 2 darkest images is impossible without further information.

So for every darkest image we manage to capture, there will be 2 phase

angles that satisfy the polarisation model (either diffuse or specular one we are dealing with). As phase angle is directly linked to azimuth angle which is the intrinsic property of a surface, we end up having 2 possible surfaces as the solution. This issue is generally known as **convex/concave ambiguity** or **azimuthal ambiguity**.

4.2.2 Diffuse or specular - we do not care!

While simplifying the material behaviour, either being diffuse or specular, could pave a long way for solving SfP problem, real-world surfaces exhibit mixed reflections. By applying diffuse assumption on specular surface or the other way round, we would yield an inaccurate result namely **model mismatch** which could happen to both zenith and azimuth angles.

Actually, as illustrated in Figure 4.2, both diffuse and specular reflections coexist within a surface. When light hits a surface, a portion of light is specularly reflected back to the same medium; the rest transmits to another medium and a fraction of that would go through scattering and transmit back to the medium where the light comes from. With this notion, we can have a mixed polarisation model, which solves model mismatch, by summing diffuse and specular models together:

$$\begin{aligned} I_{\vartheta}^{\text{mixed}} &= I_{\vartheta}^{\text{diffuse}} + I_{\vartheta}^{\text{specular}} \\ &= (I_d + I_s) + (I_d\rho_d - I_s\rho_s)\cos(2\vartheta - 2\phi). \end{aligned} \tag{4.7}$$

4.2.3 Extra constraints needed

To get an accurate surface reconstruction, we have to resolve azimuthal ambiguity. This could be done with an additional piece of information to identify whether the surface is convex or concave *e.g.* photometric stereo, shading cue or even depth camera. Surprisingly, the setup we use in chapter 3, gives the constraints we need for resolving the ambiguity. The key is to capture every point on the surface from at least 2 viewpoints. How does this work?

Let's go back to a single-view setup and further assume grayscale radiance (for the sake of explanation). For a single view, there are four unknowns in

the mixed model: two components of the orientation of the surface normal and I_d and I_s. A single observation with a demosaiced polarisation camera provides three observations (the three parameters of a sinusoid: unpolarised intensity, degree of polarisation and angle of polarisation). Hence, inverting the mixed model is ill-posed for one view. Nevertheless, the surface normal is independent of viewpoint and, under the Lambertian assumption, so is I_d. Thus, if we add a second view, this means we add one additional unknown (only I_s from the second view) but gain three more observations at which point the problem becomes well-posed. Hence, NeRF setting where there are many viewpoints captured for one point on the object, would lead to robust optimisation, being less sensitive to noise.

4.3 Multi-view mixed polarisation model

The mixed polarisation model shown in Equation 4.7 is still fully valid but less useful as most of the parameters are defined in the specific camera coordinate. This section will rewrite the mixed polarisation model to explicitly account for the transformation from a world coordinate to the coordinate of a given camera. Figure 4.3 shows our setup accompanying the below derivation.

Beginning with the basic properties of the scene, we denote a 3D point in world coordinates as $\mathbf{x} = (x, y, z)$ and the surface normal in world coordinates at that point as $\mathbf{n}(\mathbf{x}) = [n_x(\mathbf{x}), n_y(\mathbf{x}), n_z(\mathbf{x})]^T$ with $\|\mathbf{n}(\mathbf{x})\| = 1$. We define camera pose by a rotation matrix $\mathbf{R} \in SO(3)$ that rotates world to camera coordinates and the position of the camera centre by $\mathbf{c}$. Hence, the view direction from which a camera with centre $\mathbf{c}$ observes point $\mathbf{x}$ is given by:

$$\mathbf{v}(\mathbf{c}, \mathbf{x}) = \frac{\mathbf{c} - \mathbf{x}}{\|\mathbf{c} - \mathbf{x}\|}. \tag{4.8}$$

The spherical coordinates of the surface normal at $\mathbf{x}$ in the camera coordinate system can be obtained as follows. The zenith angle is given by the angle between $\mathbf{n}(\mathbf{x})$ and $\mathbf{v}(\mathbf{c}, \mathbf{x})$: $\theta(\mathbf{c}, \mathbf{x}, \mathbf{n}) = \arccos\left(\mathbf{n}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{c}, \mathbf{x})\right)$, while the azimuth angle is given by rotating the surface normal to camera coordinates: $\mathbf{n}_c(\mathbf{R}, \mathbf{x}, \mathbf{n}) = \mathbf{R}\mathbf{n}(\mathbf{x})$, and then converting the Cartesian representation to

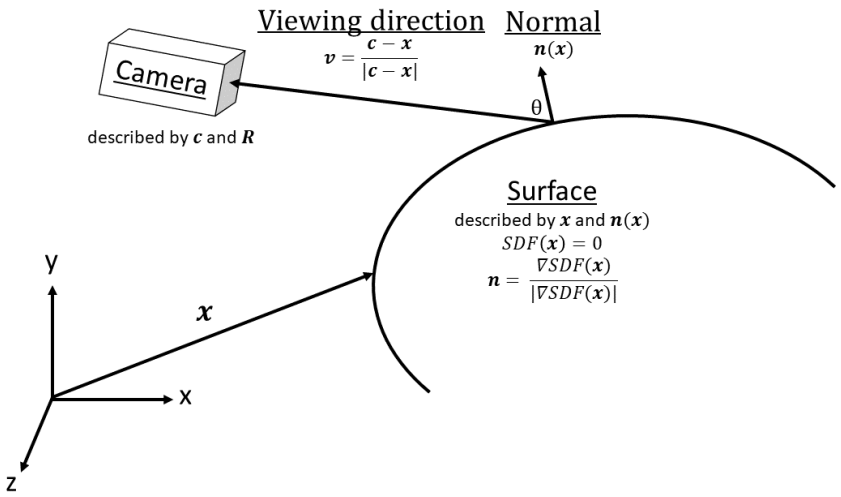


Figure 4.3. **Relationship between parameters.**

the spherical azimuth angle:

$$\varphi(\mathbf{R}, \mathbf{x}, \mathbf{n}) = \text{atan2}\left(\frac{n_{c,y}(\mathbf{R}, \mathbf{x}, \mathbf{n})}{n_{c,x}(\mathbf{R}, \mathbf{x}, \mathbf{n})}\right). \tag{4.9}$$

Assuming known refractive index η, we can now rewrite our multi-view mixed model as a function of all free parameters: position ($\mathbf{x}$), the camera pose ($\mathbf{R}, \mathbf{c}$), the surface normal ($\mathbf{n}$) and diffuse and specular unpolarised radiance (I_d, I_s):

$$\begin{aligned} I_\vartheta^{\text{mixed}}(\mathbf{x}, \mathbf{R}, \mathbf{c}, \mathbf{n}, I_d, I_s) = (I_d + I_s) + \\ [I_d \rho_d(\theta(\mathbf{c}, \mathbf{x}, \mathbf{n})) - I_s \rho_s(\theta(\mathbf{c}, \mathbf{x}, \mathbf{n}))] \times \\ \cos\left[2\vartheta - 2\phi(\varphi(\mathbf{R}, \mathbf{x}, \mathbf{n}))\right]. \end{aligned} \tag{4.10}$$

4.4 MLP re-parameterisation

NeRF, as explained in the section 3.1, pioneers scene representation using 2 MLPs: a 'spatial' MLP outputting volume density and a 'directional' MLP outputting outgoing radiance along viewing direction. To render a pixel's

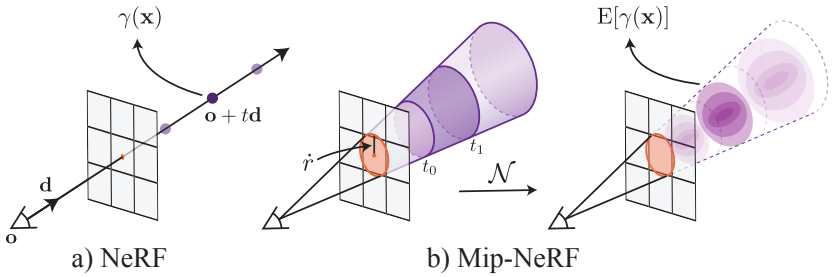
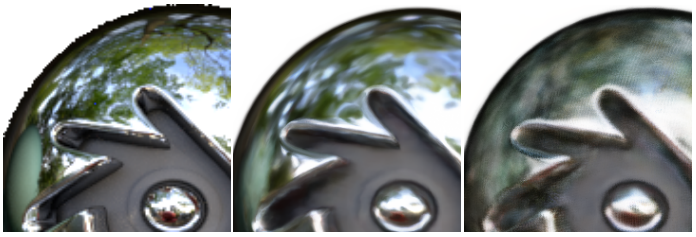


Figure 4.4. **The comparison between NeRF [55] and Mip-NeRF [5].** NeRF (a) samples points **x** along the rays that are traced from the camera centre through each pixel whereas Mip-NeRF (b) reasons about 3D conical frustum defined by a camera pixel. Image from [5].

color, NeRF casts a ray through that pixel, queries MLPs for scene properties, and composites these values into a color.

One of the reasons that help NeRF achieves a photorealistic result, is a roughly constant distance between camera and the scene content. Once this condition is broken, the rendering appears blurred in close-up views and consists of aliasing in distant views. The straightforward solution is supersampling multiple rays per pixel, which is computationally intensive and quickly becomes impractical as rendering each ray requires querying MLPs hundreds of times. The key idea behind supersampling is that there are more possible contents in a distance space and the scene thus requires more rays to represent the further contents. Inspired from mipmapping in graphics, Mip-NeRF [5] casts a cone using Gaussians that approximate the conical frustums corresponding to the pixel. Figure 4.4 shows the contrast of scene coverages using ray in NeRF and cone in Mip-NeRF. By using cone instead of ray, Mip-NeRF could be trained on a single neural network that models the scene at multiple scales.

Guaranteed by the universal approximation theorem in Mathematics, the MLP is naturally a universal approximator that could mimic behaviour of any continuous functions. The job of machine-learning engineers/scientists is to find an architecture that helps the machine learn the specified task. NeRF is such an architecture that allows a matchine to learn the scene representation. Even though NeRF could successfully reproduce specular reflections, a careful inspection shows that those reflections are faked by using isotropic emitters



Ground Truth Ref-NeRF [82] Mip-NeRF [5]

Figure 4.5. **The comparison between Ref-NeRF and Mip-NeRF renderings.** By reparameterisation, Ref-NeRF simplifies interpolating task of MLP thus gaining a clearer result.

inside semi-transparent shells. The issue here is not that the MLP cannot learn an accurate view-dependent specular reflection, but the task itself is too challenging for the machine to properly learn, leading to an undesired solution that looks similar to specular reflection.

Since an object appearance is a complex entanglement in lighting direction, object geometry and material, asking a network to predict an outgoing light by giving viewing direction as input, is therefore a very challenging task on its own. Even in a simple setup, the specular reflection could drastically change as we move along the surface. Without dense sampling, interpolation between views is almost impossible. Ref-NeRF [82], built on top of Mip-NeRF, re-parameterises the input for directional MLP. Instead of using plain viewing direction, Ref-NeRF employs a reflected view as the network input. Assuming a uniform object material, the outgoing radiance is a static function of reflected direction. Doing so simplifies the network's task of interpolation, helping the model learn an accurate specular reflection. Figure 4.5 illustrates the specular reflection from a metal surface as learned by Ref-NeRF and Mip-NeRF. By re-parameterising the network input, Ref-NeRF clearly outperforms Mip-NeRF whose interpolating task is more challenging.

4.4.1 Implicit BRDF

First thing first - what is BRDF? BRDF, a short form of Bidirectional Reflectance Distribution Function, is a function of 4 variables that defines how

light from a source is reflected off an opaque surface. The function takes an incoming light direction ω_i and outgoing direction ω_o, and returns the ratio of reflected radiance L_o exiting along ω_o to the irradiance incident E_i on the surface from direction ω_i:

$$f_o(\omega_\mathrm{i}, \omega_\mathrm{o}) = \frac{dL_o(\omega_\mathrm{o})}{dE_i(\omega_\mathrm{i})} = \frac{dL_o(\omega_\mathrm{o})}{L_i(\omega_\mathrm{i}) \cos\theta_i d\omega_\mathrm{i}}. \tag{4.11}$$

In the physical world or physically plausible simulation, BRDF possesses the following properties [21]:

- positivity: $f_o(\omega_\mathrm{i}, \omega_\mathrm{o}) \geq 0$,
- obeying Helmholtz reciprocity: $f_o(\omega_\mathrm{i}, \omega_\mathrm{o}) = f_o(\omega_\mathrm{o}, \omega_\mathrm{i})$,
- conserving energy: $\forall\omega_\mathrm{i}$, $\int_\Omega f_o(\omega_\mathrm{i}, \omega_\mathrm{o}) \cos\theta_o d\omega_\mathrm{o} \leq 1$,

where Ω is the unit hemisphere containing all posible values for ω_o.

BRDF could be measured directly from real objects using calibrated cameras and light sources (see Figure 4.6) or gonioreflectometer. The measurement involves moving light source many times to establish the relationship between incoming and outgoing lights at various angles - thus being a very time-consuming process. Alternatively, we could assume material property and follow a well-studied model such as Lambertian (perfectly diffuse), Phong (plastic-like specular), and Torrance–Sparrow (specular microfacet) models.

To find the outgoing radiance L_o, we integrate both sides of the Equation 4.11, obtaining the rendering equation:

$$L_o(\omega_\mathrm{o}) = \int_\Omega f(\omega_\mathrm{i}, \omega_\mathrm{o}) L_i \cos\theta_i d\omega_\mathrm{i}. \tag{4.12}$$

As we are dealing with diffuse and specular reflections, we can further deconstruct the rendering equation into 2 separated components, and extend the BRDF to be the function of position, also known as Spatially Varying Bidirectional Reflectance Distribution Function (SVBRDF):

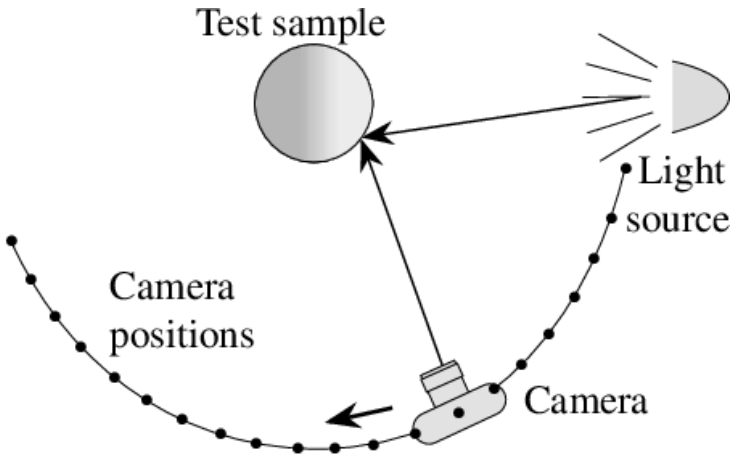


Figure 4.6. **Gonioreflectometer for measuring BRDF.**

$$L_{\mathrm{o}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{o}}) = \underbrace{\frac{b_{\mathrm{d}}}{\pi} \int_{\Omega} L_{\mathrm{i}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{i}})(\boldsymbol{\omega}_{\mathrm{i}} \cdot \mathbf{n}) d\boldsymbol{\omega}_{\mathrm{i}}}_{\text{diffuse}} + \underbrace{\int_{\Omega} f_{\mathrm{s}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}; b_{\mathrm{s}})(\boldsymbol{\omega}_{\mathrm{i}} \cdot \mathbf{n}) L_{\mathrm{i}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{i}}) d\boldsymbol{\omega}_{\mathrm{i}}}_{\text{specular}}, \tag{4.13}$$

where b_{d} is diffuse roughness, b_{s} is specular roughness, and $\mathbf{n}$ is surface normal. Since ω_{i} and $\mathbf{n}$ are a unit vector, $\cos\theta_i$ and $(\omega_{\mathrm{i}} \cdot \mathbf{n})$ are interchangeable *i.e.* representing the same quantity.

Under Lambertian assumption (as mentioned in section 4.2.3), our diffuse reflection would be a function of position only. Striving for real-world objects, we do not set a strong assumption about specular reflection but analyse the appearance contributed from geometry, illumination, and material properties.

1st observation: for a rotationally-symetric BRDF that satisfies $f(\omega_{\mathrm{i}}, \omega_{\mathrm{o}}) = p(\omega_{\mathrm{r}} \cdot \omega_{\mathrm{i}})$ for a lobe function p, the outgoing radiance is a function of the reflection direction ω_{r}:

$$L_o(\omega_{\mathrm{o}}) \propto \int L_i(\omega_i) p(\omega_{\mathrm{r}} \cdot \omega_i) d\omega_i = F(\omega_{\mathrm{r}}). \tag{4.14}$$

2nd observation: the fresnel effect affects how materials reflect light at different viewing angles. This is proportionate to $\cos\theta$.

3rd observation: the latent parameters z from SDF network, which encodes

the geometric information of the object, would help the MLP assign color to the right 3D location.

From all the observations and the Lambertian assumption, we could set 2 separated MLPs to represent diffuse and specular reflections by using following rationale. Since diffuse radiance does not depend on viewer direction we use simply $F_{I_\mathrm{d}} : (\mathbf{x}, z) \mapsto I_d$ to map a 3D position to the RGB unpolarised diffuse radiance at that point. Specular reflectance does depend on viewing direction and geometry, parameterised as discussed above: $F_{I_\mathrm{s}} : (\mathbf{x}, \max(\mathbf{n} \cdot \mathbf{v}, 0), \mathbf{ref}(\mathbf{n}, \mathbf{v}), z) \mapsto I_\mathrm{s}$. Again, the output is RGB unpolarised specular radiance at that point, but this time in the direction, $\mathbf{v}$, of the viewer. Since viewing rays may not be restricted to the upper hemisphere about the normal, we clamp the cosine of the view angle to be non-negative. The $\mathbf{ref}(\mathbf{a}, \mathbf{b})$ function represents a vector reflection.

Overall, there are 3 separated networks: neural SDF, diffuse and specular networks. Neural SDF and diffuse networks are conditioned on position and its variant only *i.e.* no view dependence. It is noted that we output the latent parameters z from neural SDF network and use them as the input for diffuse network to ease the MLP's task. Being view dependent, the specular network requires both viewing and directional information. The reflected view and cosine angle are derived from normal direction which could be obtained by calculating a derivative of SDF value, while we share the same position-related input as done in diffuse network.

Assumption: Figure 4.7 provides a brief overview of how each element relates to each other, from network input to loss calculation. As we moved along each component, we gradually introduced assumptions as follows: A) both diffuse and specular polarisation models (Equations 4.3 and 4.5) are derived from Fresnel Equations and only valid for dielectric material (*e.g.* human skin); B) the incident light must be unpolarised, which is generally true in nature; C) the light undergoes a single bounce *i.e.* no interreflections; D) diffuse reflectance from subsurface scattering is independent of viewing direction *i.e.* Lambertian model; E) the refractive index is assumed to be 1.5, which well represents our objects of interest.

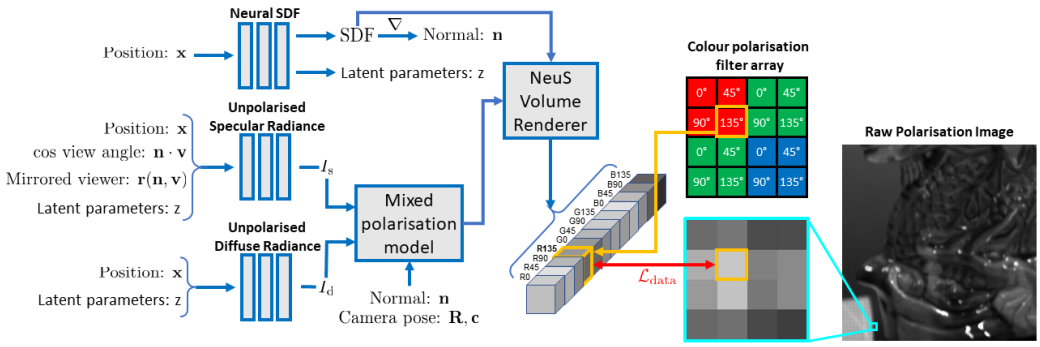


Figure 4.7. **Neural shape-from-polarisation.** We use a neural signed distance function (Neural SDF in the diagram) to represent the surface and derive the surface normal via differentiation. Two other MLPs (Unpolarised Specular Radinace and Unpolarised Diffuse Radiance in the diagram) learn unpolarised diffuse and specular radiances as black boxes, with diffuse radiance being conditioned on position and geometric features from Neural SDF MLP i.e. Lambertian assumption, and specular one additionally on the cosine of zenith angle and reflection direction. Via a mixed polarisation model, we capture the dependence between surface normal, camera pose and unpolarised radiances to predict polarised radiance. This is volume rendered according to the NeuS [84] model for any combination of colour channel and polariser angle. We select the appropriate channel for each measured pixel and compute a data loss. While it may not be obvious in the diagram, we want to emphasise the following links: A) the latent parameters obtained from Neural SDF Network are used as an input of both diffuse and specular networks, and B) the normal obtained from Neural SDF Network is used to calculate cosine of zenith angle and reflection direction. Those links are omitted from the diagram for a compact visualisation.

4.4.2 Rotationally precised ray

As we incorporate multi-view polarisation information into our pipeline, we need to put an extra care on how we define camera rays. In essence, our rays have to be distinguishable when rotating, in addition to telling how far the point is from the camera centre in traditional graphics. Thus we define the ray that we render in terms of a camera pose $(\mathbf{R}, \mathbf{c})$ and a pixel with normalised coordinates $\mathbf{u} = [u, v, 1]^T$; the ray is defined as $\{\mathbf{p}(\mathbf{R}, \mathbf{c}, \mathbf{u}, t) = \mathbf{c} + t\mathbf{R}^\top \mathbf{u}, t \geq 0\}$.

We then volume-render the ray to acquire the object color, using NeRF [55] time-discrete volume rendering:

$$I(\mathbf{R}, \mathbf{c}, \mathbf{u}, \vartheta) = \sum_{i=1}^{S} w(t_i) I_\vartheta^{\text{mixed}}(\mathbf{x}_i, \mathbf{R}, \mathbf{c}, \mathbf{n}_i, F_{I_d}(\mathbf{x}_i, \mathbf{z}_i), F_{I_s}[\mathbf{x}_i, \max(\mathbf{n}_i \cdot \mathbf{v}_i, 0), \mathbf{r}(\mathbf{n}_i, \mathbf{v}_i), \mathbf{z}_i]), \quad (4.15)$$

where $\mathbf{x}_i = \mathbf{p}(\mathbf{R}, \mathbf{c}, \mathbf{u}, t_i)$, $\mathbf{n}_i = \frac{\nabla \text{SDF}(\mathbf{x}_i)}{|\nabla \text{SDF}(\mathbf{x}_i)|}$, $\mathbf{v}_i = \mathbf{v}(\mathbf{c}, \mathbf{x}_i)$, $\mathbf{z}_i$ are the geometric features from the SDF Network F_{SDF} at $\mathbf{x}_i$, the t_i are the S sample points along the ray and $w(t_i)$ the volume rendering weight given by the density derived from the SDF value, as in NeuS [84].

4.5 Results

Implementation: There are 3 neural networks in total, including SDF, diffuse and specular networks. The SDF network encodes the object geometry, via signed distance function, as a function of postion $\mathbf{x}$. This network outputs A) the signed distance function which tells us the shortest distance between a queried point and the object surface, and B) latent parameters $\mathbf{z}$ which is a compact representation of the object surfaces. In theory, either position $\mathbf{x}$ or latent parameters $\mathbf{z}$ on its own should adequately provide positional contexts to radiance networks; however, in practice, we observe that providing both enhances the training performance. For the SDF network, we employ 6 layers of MLP with intermediate layers having 256 nodes *i.e.* the network width

Scene	Method	Diffuse		Specular		Mixed		Normals
		PSNR ↑ (dB)	SSIM ↑	PSNR ↑ (dB)	SSIM ↑	PSNR ↑ (dB)	SSIM ↑	MAE ↓ (°)
bust	NeuralPIL*	23.90	0.87	18.04	0.87	26.71	0.87	15.36
	PhySG*	22.64	0.94	23.00	0.94	19.94	0.72	9.81
	PANDORA†	25.82	0.81	22.96	0.75	22.79	0.79	3.91
	NeuS*	N/A	N/A	N/A	N/A	28.09	0.85	8.53
	P-NeRF	37.79	0.999	32.21	0.984	34.63	0.963	0.49
globe	NeuralPIL*	13.09	0.55	12.92	0.55	20.04	0.66	38.73
	PhySG*	21.76	0.76	18.90	0.76	17.93	0.70	8.42
	PANDORA†	24.33	0.77	22.70	0.89	21.76	0.81	1.41
	NeuS*	N/A	N/A	N/A	N/A	23.57	0.81	3.72
	P-NeRF	34.71	0.972	29.69	0.957	29.63	0.933	0.14

Table 4.1. **Quantitative evaluation** on PANDORA [19] synthetic image benchmark. * = method is given access to ground truth demosaiced RGB images. † = method is given access ground truth demosaiced 12 channel RGB/polarisation images.

of 256 in size, and softplus as an activation function. As explained earlier, taking position $\mathbf{x}$ and latent parameters $\mathbf{z}$, the diffuse network learns to map positional information to diffuse radiance. Depending on the dataset types, the diffuse network requires a different size of MLP: 6 layers for real datasets and 2 layers for synthetic datasets with 256 nodes for each layer. This reflects the dimension of the datasets *i.e.* large images requiring 'powerful' MLPs to represent the scene. Because of being view-dependent, the specular network additionally takes cosine of zenith angle $\cos(\theta)$ and reflected view direction $\mathbf{ref}(\mathbf{n}, \mathbf{v})$. For a similar reason, specular network's depth is 6 for real datasets and 2 for synthetic datasets with 256 nodes belonged to each layer. For both diffuse and specular networks, we use ReLU as an activation function for all layers except the final one where exponential is applied to reflect a wide range of HDR data.

To get a performance boost, we apply proposal network and hash encoding as done in Mip-NeRF 360 and instant NGP respectively. As suggested by NeuS, we apply $L1$ loss to our final rendering as in Equation 3.13. Since our method polarises the radiances predicted by neural network, we call our method Polarised Neural Radiance Field or P-NeRF.

Figures 4.8 and 4.9 illustrate the 3D reconstruction as well as radiance

decomposition of synthetic datasets including bust and globe, while Table 4.1 shows the corresponding quantitative evaluation. After properly taking physics phenomena into account, our model becomes better at learning the scene. Our model can now predict the specularity on the bust's forehead, which were missed in the blackbox model *i.e.* 12-channel NeRF. Furthermore, the hole around neck in bust geometry, which without physical insights was apparent, is now filled and becomes comparable to geometric prediction from PANDORA. We see a significant improvement on globe geometry without rough artifacts. Nonetheless, the grid line (in diffuse reflection) on the ocean of the globe is missed.

Even though, by injecting physical knowledge into our pipeline, P-NeRF is capable of replicating a synthetic scene, our method struggles with real datasets where noise appears. Our model could not produce the whole scene as shown in Figure 4.10. The Figure 4.11 shows the objects for which our method manages to learn the whole scene. Nevertheless, the geometric reconstructions are full of artifacts.

The contrast between result from synthetic and real datasets, suggests that there could potentially be an issue with the real-world training. Thus, in the next chapter, we will explore the causes of the issue whether it comes from the captured images or training practicality. For this chapter, we have seen how we incorporate SfP technique into machine learning pipeline. In particular, we develop a mixed polarisation model and make a modification so that the model is the function of camera pose, allowing multi-view training, before demonstrating the plausible results which are trained on synthetic datasets.

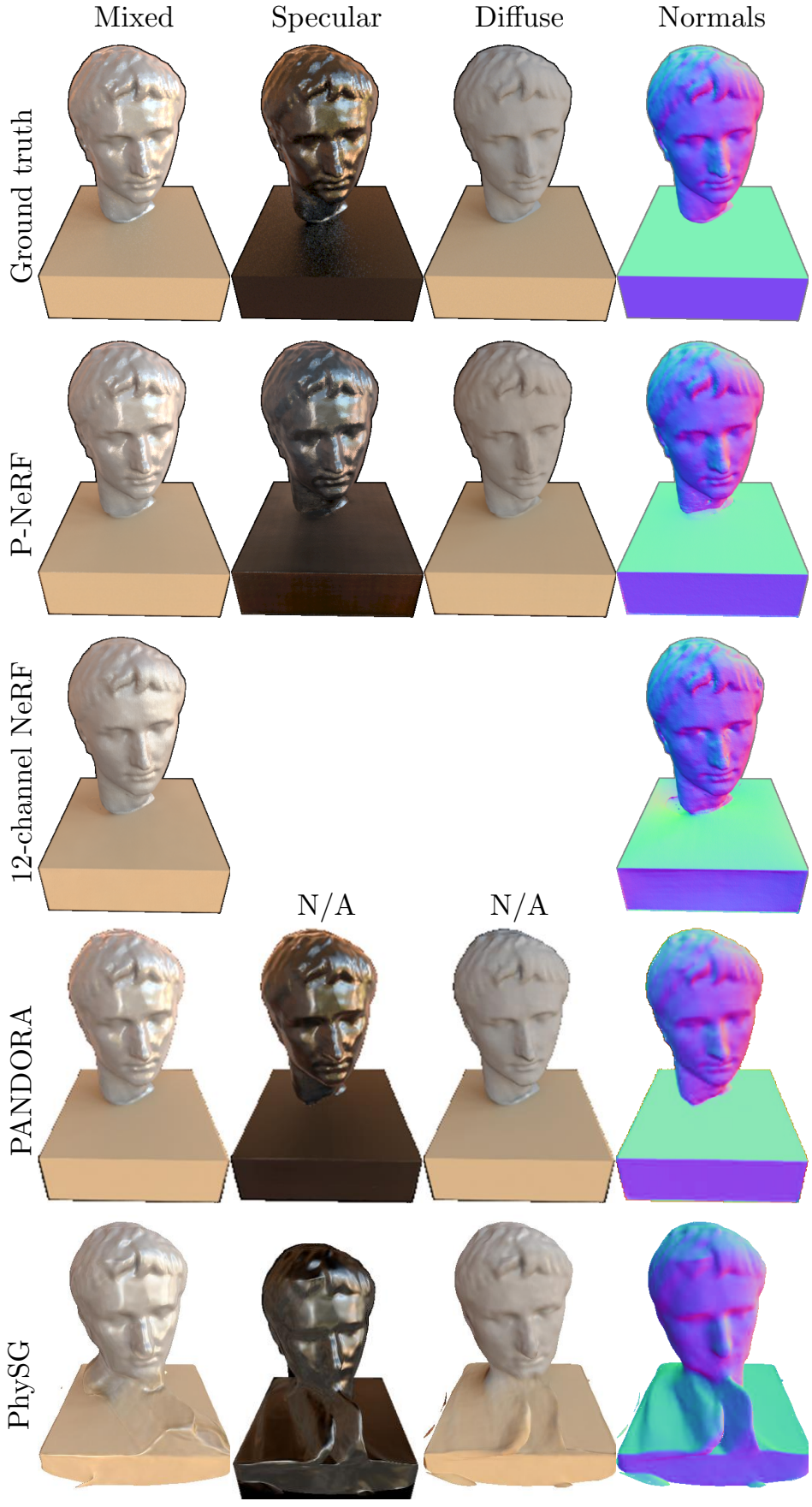


Figure 4.8. **Reflectance decomposition and geometry estimation against ground truth (synthetic) bust data.**

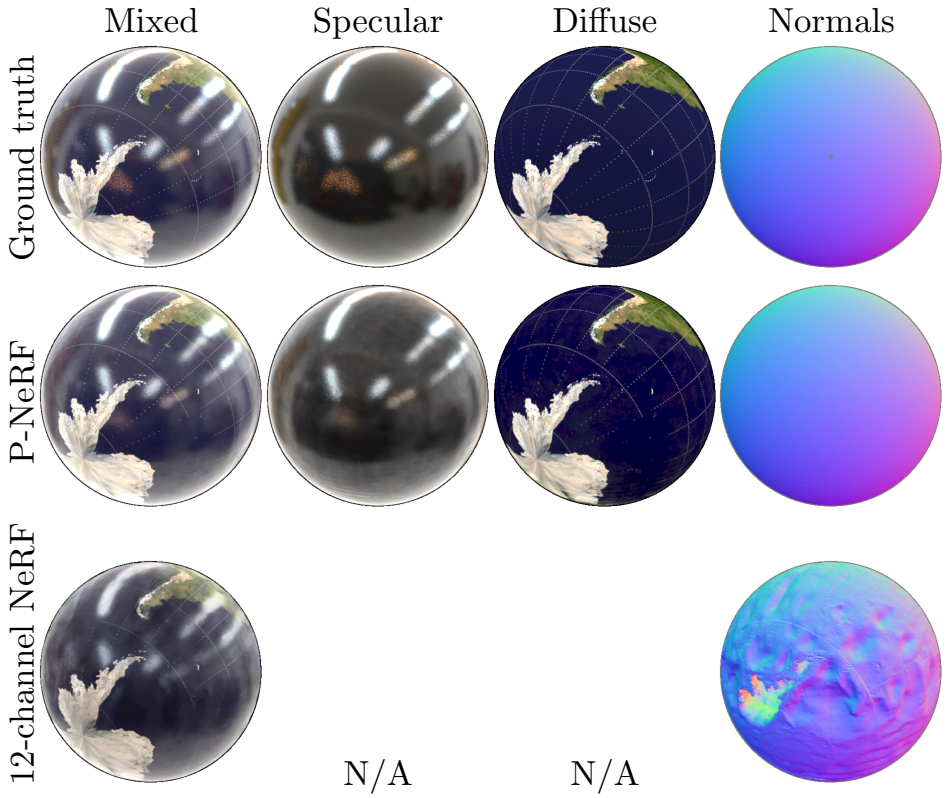


Figure 4.9. **Reflectance decomposition and geometry estimation against ground truth (synthetic) globe data.**

Figure 4.10. **Reflectance decomposition and geometry estimation of (real-world) gnome and vase data.**

Figure 4.11. **More P-NeRF results of (real-world) Shakespeare and owl data.**

5 The missing elements

> The good thing about science is that it's true whether or not you believe in it.
>
> *Neil deGrasse Tyson*

While chapter 4 illustrates an elegant blend between traditional shape from polarisation and emerging neural rendering, there are many details that were overlooked, which we will investigate in this chapter.

5.1 Data distribution

In Polarised Neural Radiance Fields, we put a set of polarised images of a scene into a pipeline which predicts diffuse and specular radiances as well as the scene geometry. Apart from raw image (section 3.4.1) obtained from camera sensor, we never mention image characteristic beyond filter array which is physically applied on top of camera sensor, and as a camera user, we do not have much control over how the filter array could be arranged differently. The same applies for noise sensitivity, focus capability, image resolution and the list goes on, where we have limited level of control.

As computer vision researchers, we tend to take it for granted that our images are doing their job as an input to the pipeline we are studying. As long as the captured images look reasonably good with overall details being in an appropriate range, we generally assume that our collected data is good for the next processing step. If this is not the case, we can adjust involving

factors such as focusing distance for a blurred image or scene lighting for a bright/dim scene, and start the whole capturing again. Nonetheless, this might not be possible when we use public dataset which is collected for other purposes.

In this section, we will dissect a 'good' image from data-distribution perspective, propose a solution for image saturation when we have no control over input data, and lastly suggest a loss function that works well on raw images that we use in our experiment.

5.1.1 What is a good image?

In the context of data collection, a perfect image is the image that captures scene information at the highest quality allowed by a particular camera. The perfect image is the ideal situation that rarely, if ever, happens in a physical lab. Hence, instead of aiming for a perfect image, we will try capturing a good image. So how does a good image look like?

Before moving any further, we want to introduce histogram, a tool that could visually illustrate the frequency of a pixel intensity appearing on an image. For those who are unfamiliar with histogram, it is a representation of the distribution of quantitative data. Typically, the x-axis would show the range of values in the form of "bin" and y-axis would be the number of times we found a value within the bin interval. We could distinguish a good image from bad ones by looking at the histogram of pixel intensity.

To better identify the characteristic of a good image, we will begin with an obviously bad image. That image could be a very dim image whose pixel intensity ranging from 0 to 400 or around a tenth of 12-bit camera capacity *i.e.* HDR or high dynamic range image. When we count the number of pixels that fit into each histogram bin, we will get the diagram shown in the top of the Figure 5.1. There is nothing looking suspicious about the diagram until we take into account the sensor capacity which could store up to 4096 different pixel values. Realizing this, we plot another histogram showing the entire range of sensor capacity (Figure 5.1, middle). We can now tell immediately that this image has a bad data distribution since there is no captured pixel value on the right side of the histogram. How could we

improve the distribution so that the poor image become a better image?

Without further assuming anything about the captured scene, *i.e.* universally applied to any dim images, we can digitally scale the image by multiplying each pixel intensity by a factor which is larger than 1. The factor is given by dividing the maximum possible pixel value by the current maximum pixel value and then rounding down the result. In our case, we multiply every pixel by the factor of 10 and the bottom of the Figure 5.1 shows the histogram of the newly distributed data. As expected, the diagram keeps the same shape with x-axis being scaled. This is what we call a good image whose pixel values fill in most of the histogram from low-value bin to high-value bin, and cover almost an entire range of sensor capacity.

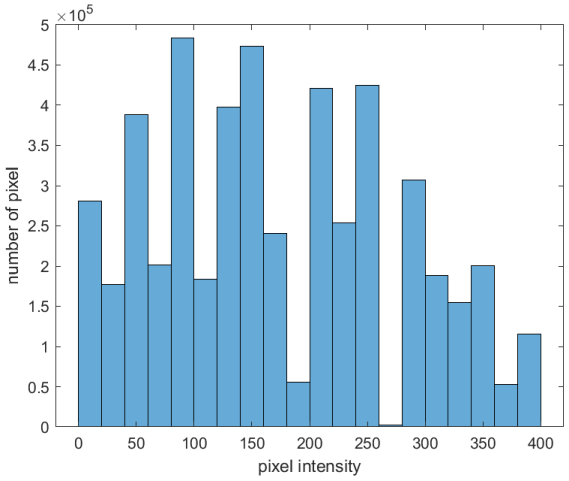
Digitally scaling the whole image with a certain factor as done in the above paragraph could alleviate the poorly distributed data, but does not recover the lost information due to inappropriate capture. For instance, a dim pixel value of 215 would become the value of 2150 after scaling by a factor of 10, whereas a proper captured pixel could range between 2145 and 2154. Therefore, wherever possible, we should aim to capture a good image in the first place rather than applying a digital hack after the fact.

As dim images are simple to spot, we do not observe them in the datasets. So we never have to scale up pixel intensity, in practice.

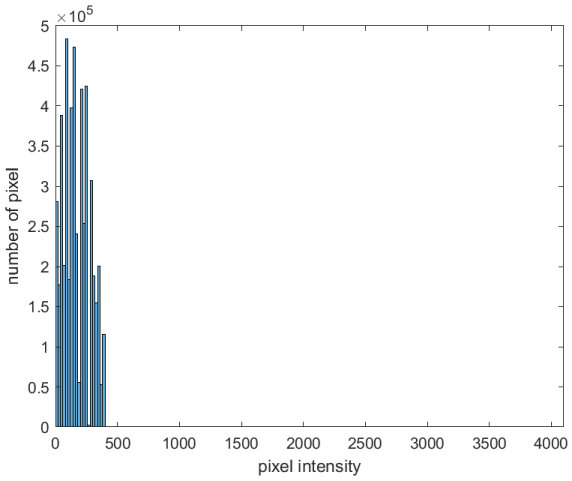
5.1.2 Saturation handling - learning to ignore

We have just seen one end of the spectrum where an image is too dim and we lose image details as we try to recover the data distribution. This section will show another end of the spectrum where we have a brightly lit scene and the captured image is saturated. Would the scaling strategy in section 5.1.1 work? Let's find out.

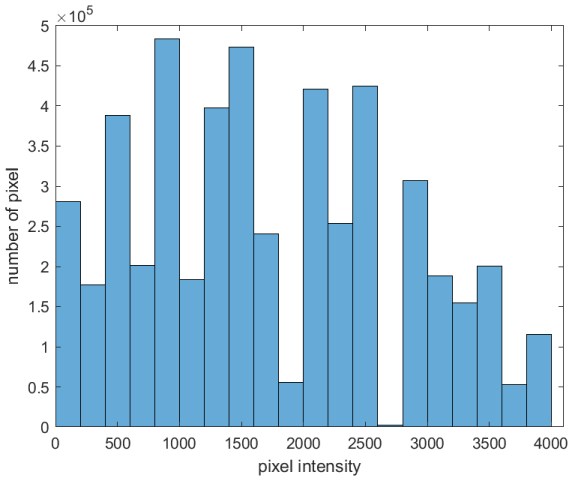
We start with plotting a histogram of a saturated image as shown at the top of Figure 5.2. We divide each pixel intensity by a factor of 10 and then re-plot the histogram, getting the bottom of Figure 5.2. Not only do we get a clutter of low-value pixels, but we also lose scene details when converting from decimal to integer (also known as quantisation). So naively repeating the method for dim images on a saturated image would worsen the issue.



Dim image

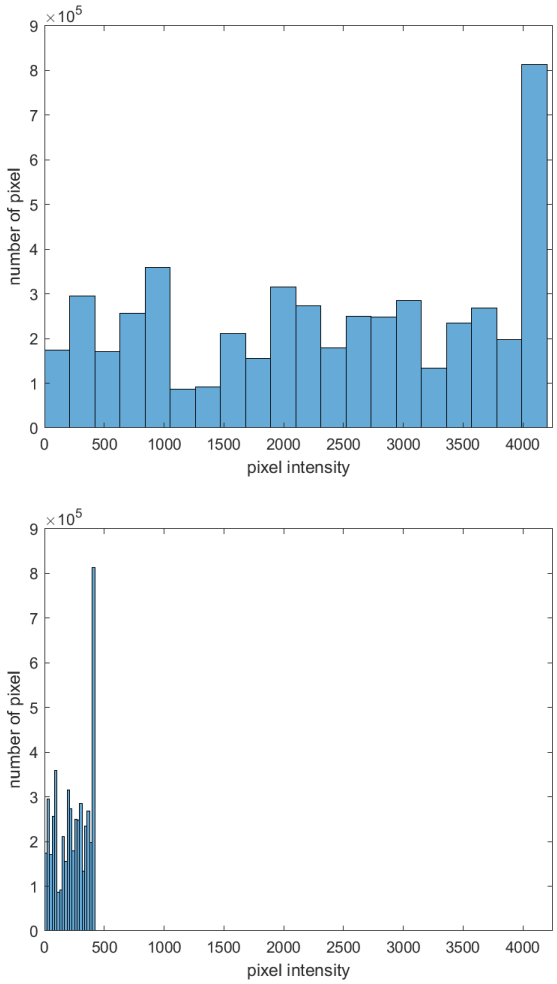


Dim image on a 12-bit scale



Digitally scaled dim image

Figure 5.1. **Data distribution of a dim image.** (top) The histogram that plots the data distribution, (middle) the same histogram that shows the whole possible data range, and (bottom) the histogram of the dim image that is digitally scaled by a factor of 10.



Saturated image

Saturated image being scaled down

Figure 5.2. **Data distribution of a saturated image.** (top) The histogram that plots the data distribution, and (bottom) the histogram of the saturated image that is digitally scaled down by a factor of 10.

What could we do to improve the image? Since the intensity beyond saturation point is completely ignored by camera sensor, unless relying on heuristic approach, there is no way to retrieve the ignored information. What we could do as a person who train neural networks is to be aware that our input images are saturated and inform machine that there is a saturation. Doing so would simplify the task of interpolation and hence getting a better fit from MLPs.

Spotting a saturated image. While we cannot say for sure which image is saturated from seeing the data distribution alone, there is a common characteristic which is shared among saturated images. Because the pixel that should be brighter than the saturation point would be recorded as bright as the saturation, there would be a lot of pixels having the value of saturation. What these mean is the last bin of the histogram would be significantly taller than the others (Figure 5.2, top). It is noted that this is not a guarantee but we can say that an image is not saturated when none of the pixels belong to the last bin of the histogram.

Since we don't know the exact information in the saturation region, an intuitive approach is to ignore the saturated pixels and let the model only learn from unsaturated pixels. However, such approach would not fully utilise the information provided by the image. A better way to deal with saturation is to encourage model to predict as good as the provided information and not penalise when both training data is saturated and the model predicts a value larger than saturation point. To simplify our discussion, we call model prediction being unsaturated/saturated when the model predicts a value under/above the saturation point. We could break down into 4 scenarios: 1) when training data and model prediction are unsaturated, the model should learn to meet the training data; 2) when training data is saturated whereas the model prediction is not, we expect the model prediction to get closer to the training data; 3) when model prediction is saturated but the training data is not, the model should become unsaturated and follow what data suggests; and 4) when both training data and model prediction are saturated, we have no information to tell how far away the model is from ground truth data - thus excluding gradient descent from the training. In summary, we can draw the below table noting whether to include gradient descent in the

training or not:

	saturated prediction	unsaturated prediction
saturated data	exclude	include
unsaturated data	include	include

Calling a saturation point $C_{\max}$, we could put this notion into practice by creating a conditional mask:

$$M_s = (\hat{C} < C_{\max}) \vee (C < C_{\max}), \tag{5.1}$$

and applying to color loss (Equation 3.13) to obtain:

$$(\mathcal{L}_{\text{color}})_{\text{masked}} = M_s|\hat{C} - C|. \tag{5.2}$$

By this minimal implementation, we utilise all the information provided from an image - both unsaturated and saturated pixels.

5.1.3 From LDR to HDR

As we have replaced a traditional low dynamic range (LDR) with high dynamic range (HDR) image, the data is technically stretched while preserving good image quality as we capture from the source. For 12-bit images, the range between lowest- and highest-possible pixel values become 4095 rather than 255. Since the loss function is basically the difference between model prediction and training data as defined in Equation 5.2, the model would be biased towards the bright region in the image. Calculating loss in the linear space becomes unsuitable [61] for HDR images. Instead, the loss defined in logarithmic space is found to be more robust to a larger intensity range.

Thus, we define a new color loss as:

$$(\mathcal{L}_{\text{color}})_{\text{HDR}} = M_s|\log(\alpha\hat{C} + \beta) - \log(\alpha C + \beta)|, \tag{5.3}$$

where α and β are hyperparameters which we found setting $\alpha = 60$ and $\beta = 3$ leads to a stable training and a good result. Even though the Equation 5.3 seems arbitrarily set, there is a logic behind both hyperparameters. In the image space, it is highly possible that there is at least 1 pixel whose intensity

value is 0. The main point of having β is to avoid log(0) which is undefined in Mathematics, while α is added to allow the model to work on an appropriate range in the logarithmic curve. Figure 5.3 compares the difference between linear and logarithmic curves. The logarithmic graph of scaled data yields a good balance of steep gradient in low-value region and shallow gradient in high-value region - hence being suitable for HDR data.

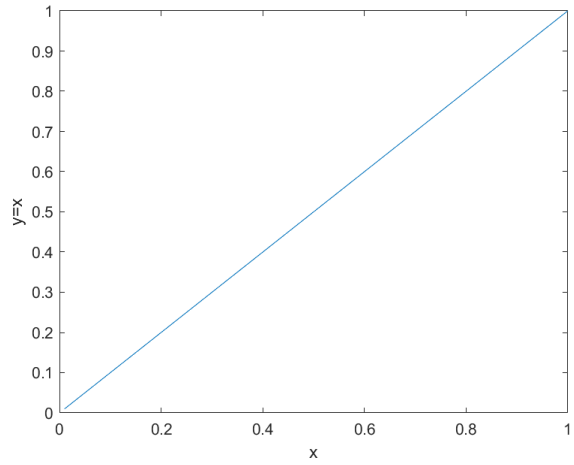
5.2 Invisible surface

In the chapter 4, the model works only on some datasets. This shows that there is a room for the model to optimise in the wrong direction *i.e.* towards the undesired solution. To ensure generalisation across datasets, we want to put further constraint in the training process but what would it be?

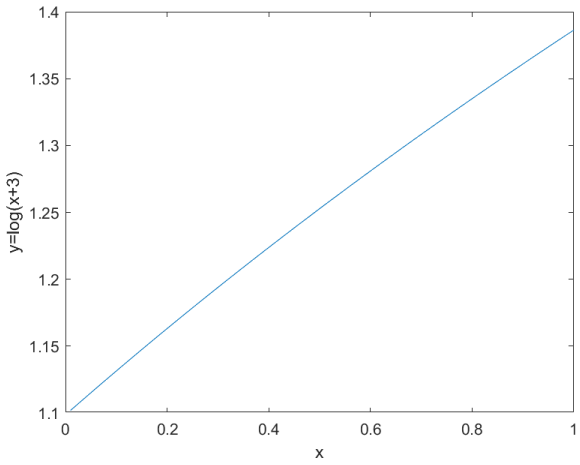
We consider a surface *i.e.* the intersection between 2 mediums. This surface has 2 faces, (A) the one we observe and (B) the other one that is hidden behind. In the scope of this work, assuming opaque material, we don't consider the light that travels inside the material and penetrates the surface from face B. Hence, the observed reflection could only come from face A, or mathematically speaking, the angle between surface normal $\mathbf{n}$ and viewing vector $\mathbf{v}$ must be less than 90 degrees; the surface whose the angle is larger than 90 degrees is, by definition, self-occluded and hence non-visible. The Figure 5.4 illustrates this concept.

In NeuS [84] volume rendering, the appearance is the weighted sum of the radiance values along the casted ray. As explained in the last paragraph, to be physically plausible and thus meaningful for mixed polarisation model, we want to restrict the contribution from the back-faced surface (face B) and only allow the forward-faced surface (face A) to contribute to final appearance. This means that contributing points must have 2 properties: 1) their weight is greater than zero and 2) their surface normal makes an angle that is less than 90 degrees to viewing direction. For instance, in Figure 5.5, only point D can contribute to final appearance while point C cannot, due to how surfaces are facing the viewer.

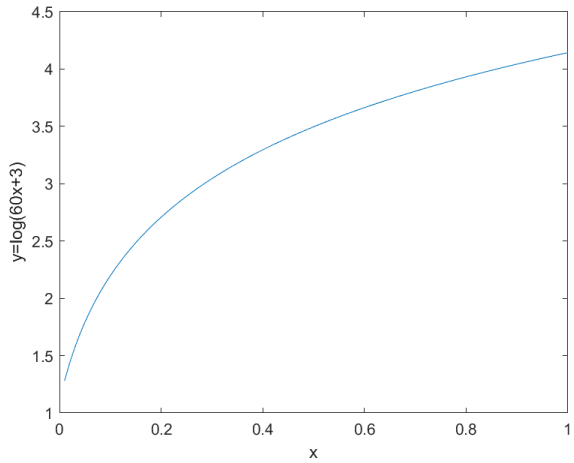
Nonetheless, we experimentally found the points that are physically implausible in the model trained with P-NeRF. While this could be simply



linear data ($y = x$)



logarithmic data ($y = log(x + 3)$)



logarithmic scaled data ($y = log(60x + 3)$)

Figure 5.3. **The relationship between x and y for different functions.** (top) linear curve gives a one-to-one ratio between x and y, (middle) logarithmic curve gives almost a linear relationship whose y-axis is shifted, and (bottom) logarithmic curve from scaled data gives a mixture of large and small gradients which well balance the signal of back propagation in the training.

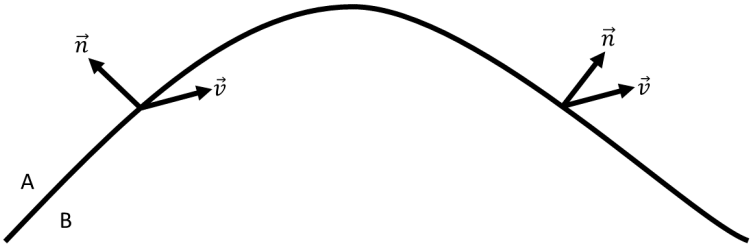


Figure 5.4. **Surface visibility.** The surface is visible when normal vector points in the same hemisphere as viewing vector (right) whereas the surface is invisible when normal vector points in the different hemisphere as viewing vector (left).

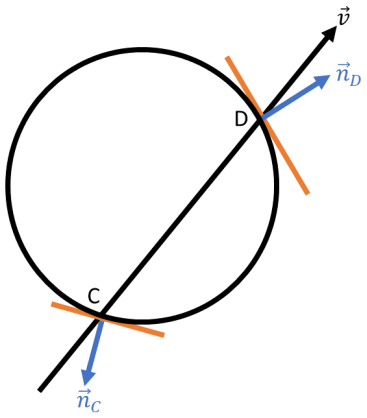


Figure 5.5. **Contributing points in volume rendering.** The camera is put above point D on a circle. During volume rendering, only points (*e.g.* point D) whose angle between surface normal and viewing vector is less than 90 degrees can contribute to final appearance, whereas points (*e.g.* point C) whose angle is larger than 90 degrees cannot.

solved by masking out the points which violate the physical constraint, doing so would stop gradient flow from these points, and as a result, the optimiser could not improve the solution. Hence, instead of solidly filtering out those points, we introduce a soft penalty - leading to a stable training. In particular, we add theta loss that encourages SDF network to produce zenith angle in a reasonable range (assuming ray sampling size N):

$$\mathcal{L}_{\text{theta}} = -\frac{1}{N}\sum_{i=1}^{N} \min(\max(\mathbf{v_i} \cdot \mathbf{n_i}, -1), 0). \tag{5.4}$$

Intuitively, when encountering a non-zero theta loss, the optimiser has to reduce it by either 1) changing the SDF so that the point has viewing angle being less than 90 degrees, or 2) morphing the object shape. The effect of latter choice is to reduce visibility of that point by lowering the weight of that point and giving a higher weight to the object part laying in front of that point.

5.3 Smooth surface across noisy measurements

When using raw images captured from camera sensor, there is a trade-off between spatial resolution and polarimetric information. Compared to fully demosaiced images, our raw images contain only one twelfth of the total information provided by the scene. This is very sparse and the 3D reconstruction could be heavily affected at the texture edge where a large intensity shift occurs.

To make the matter worse, the raw measurements naturally come with noises due to capturing process. When the camera shutter is opened, there is a number of photons hitting camera sensor and the electrons at that pixel are stimulated (proportionally to the number of photons). In practice, though, the sensor is generally over-sensitive and could measure a random value even without a single photon hitting that pixel. So, depending on the camera, the manufacturer has to introduce an offset which treats a certain measurement as zero level, and what's below that threshold is sometimes called a "negative measurement". A good camera would set that threshold high enough so

that the measurement is zero in the absence of light. The second source of noise comes from the photon behavior which follows Poisson distribution. The statistics describes the number of events occurring in a fixed interval of time. A good example of this distribution in daily life is the number of buses passing a particular stop over a period of time. The last source of error is introduced when converting the analog electrical signal to a digital value *i.e.* discretisation. Fortunately, combined noise can be modeled as Gaussian whose mean is zero, and neural networks would smooth out the noisy measurement as done in RawNeRF [53].

Since Polarised Neural Radiance Fields links surface normal and radiances together through mixed polarisation model, the noise shown in the image is passed on to the geometry and we do not want the reconstructed geometry to get affected by those noises. Fortunately, real-world objects usually possess piecewise coherence regarding geometry *e.g.* a smooth surface tends to continue its smoothness. So if we encourage the networks to produce a surface normal which points to a similar direction to the normal at the 3D adjacent point $(\mathbf{x} + \delta\mathbf{x})$, we will implicitly prohibit the noisy measurements to get baked into geometry. We define smoothness loss which does that job as:

$$\mathcal{L}_{\text{smooth}} = \arccos \mathbf{n}_i(\mathbf{x}) \cdot \mathbf{n}_i(\mathbf{x} + \delta\mathbf{x}). \tag{5.5}$$

In this chapter, we have made the following improvements: we introduce an extra pre-processing step that ensures well distributed data either when the images are under-exposed or saturated, we propose a logarithmic loss which better suits HDR data, we add a theta loss encouraging networks to predict a reasonable zenith angle hence leading to physically valid model, and lastly we reduce the effect of noisy measurement from the reconstructed geometry.

5.4 Lightstage application

Up until this point, we have made a minimal assumption for an incident light that the light has to be unpolarised. For most settings in nature, this is a

reasonable assumption.

In this section, we will explore a special lighting condition that leads to intrinsic properties of the object in the scene. In particular, we apply our method in spherically-uniform, unpolarised illumination. In contrast to previous methods [49, 25, 40, 41], we do not modulate individual light sources to create varying illumination patterns; we only require a single image from each camera under a fixed, uniform illumination. The method is therefore single-shot and suitable for dynamic objects such as faces, potentially running at full frame rate. Because the illumination is unpolarised, we make no assumption about its plane of polarisation relative to the camera. Rather than using polarisation to approximate diffuse/specular separation, we exploit it purely as a shape cue.

Under uniform illumination, $L_{\mathrm{i}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{i}}) = k, \forall \boldsymbol{\omega}_{\mathrm{i}} \in \Omega$, for an arbitrary constant k. The diffuse term in Eq. (4.13) then reduces to:

$$L_{\mathrm{diffuse}}(\mathbf{x}, \boldsymbol{\omega}_{\mathrm{o}}) = \frac{\mathbf{b}_{\mathrm{d}}}{\pi} k \int_{\Omega} (\boldsymbol{\omega}_{\mathrm{i}} \cdot \mathbf{n}) d\boldsymbol{\omega}_{\mathrm{i}} = \mathbf{b}_{\mathrm{d}} k \tag{5.6}$$

In other words, the diffuse radiance estimated by fitting our model directly provides the diffuse albedo map (up to a scaling). Under the same lighting, the specular radiance becomes a view-dependent reflection coefficient that, after Fresnel compensation, yields the specular albedo (R_0) map [25]. In practice, we capture multi-view face images under uniform, unpolarised light, run our PP-NeRF pipeline, take F_{I_d} as diffuse albedo, and evaluate F_{I_s} at $\mathbf{v} = \mathbf{n}$ to obtain the specular albedo.

5.5 Results

Assumption: In this chapter, we address practical limitations founded in the last chapter and therefore set no further assumptions *i.e.* we share the same set of assumptions as we previously stated in Chapter4.

5.5.1 Under spatially-varying illumination (general case)

Implementation: As we have discussed so far, to make P-NeRF model practical, we introduce 3 losses: logarithmic color loss, theta loss and smoothness loss. Technically, we also scale dim images to have a suitable range however encountering zero such images. Otherwise stated, we inherit the implementation from Chapter 4.

Since we address the practical aspects of P-NeRF by adding losses to the training, we name the method in this chapter "practical P-NeRF" or PP-NeRF. The Table 5.2 quantitatively illustrates the effects of losses introduced in this chapter. Except PSNR of specular and mixed radiances, the combination of losses provides a better training supervision hence obtaining the best result. Without theta loss, the color loss becomes dominant so the radiance renderings outperform the training with all losses by a small margin. PP-NeRF is also tested against prior methods. Shown in the Table 5.1, our method gives the best reflectance estimation as well as geometric reconstruction.

In the Figures 5.6, 5.7, 5.8, 5.9, 5.10, we show how we gradually improve our first model (12-channel NeRF) by adding physical insights (P-NeRF) and extra practical constraints (PP-NeRF); as well as results from methods with similar performance such as PANDORA [19] and PMVIR [94]. Because results from synthetic datasets (bust and globe) are fairly good since the introduction of physical insights in chapter 4, we observe PP-NeRF giving only a marginal improvement over P-NeRF. Nonetheless, it is worth pointing out that PP-NeRF is able to recreate grid line that was missed in globe's diffuse reflectance in P-NeRF result. For car dataset, the clearer details are observed in geometry estimation and specular highlights are apparent in the results provided by PP-NeRF. This leads to realistic looking of the car when considering mixed reflectance. When coming to real datasets (gnome and vase), PP-NeRF clearly outperforms our prior models (P-NeRF and 12-channel NeRF). With PP-NeRF, for the first time, we are able to recreate the whole object (recalling the vase reconstruction where we missed the mug handle). Thanks to our technique which fits the rendering directly to raw measurement, *i.e.* without first demosaicing, PP-NeRF preserve the fine

Scene	Method	Diffuse PSNR ↑ (dB)	Diffuse SSIM ↑	Specular PSNR ↑ (dB)	Specular SSIM ↑	Mixed PSNR ↑ (dB)	Mixed SSIM ↑	Normals MAE ↓ (°)
bust	NeuralPIL*	23.90	0.87	18.04	0.87	26.71	0.87	15.36
	PhySG*	22.64	0.94	23.00	0.94	19.94	0.72	9.81
	PANDORA†	25.82	0.81	22.96	0.75	22.79	0.79	3.91
	NeuS*	N/A	N/A	N/A	N/A	28.09	0.85	8.53
	PP-NeRF	**37.59**	**0.999**	**32.01**	**0.983**	**32.72**	**0.962**	**0.4290**
globe	NeuralPIL*	13.09	0.55	12.92	0.55	20.04	0.66	38.73
	PhySG*	21.76	0.76	18.90	0.76	17.93	0.70	8.42
	PANDORA†	24.33	0.77	22.70	0.89	21.76	0.81	1.41
	NeuS*	N/A	N/A	N/A	N/A	23.57	0.81	3.72
	PP-NeRF	**36.58**	**0.975**	**29.98**	**0.958**	**30.25**	**0.939**	**0.1144**

Table 5.1. **Quantitative evaluation** on PANDORA [19] synthetic image benchmark. * = method is given access to ground truth demosaiced RGB images. † = method is given access ground truth demosaiced 12 channel RGB/polarisation images. No smoothness loss is applied to show performance without demosaicing benefits i.e. 12-time more input.

Scene	Loss Element	Diffuse PSNR ↑ (dB)	Diffuse SSIM ↑	Specular PSNR ↑ (dB)	Specular SSIM ↑	Mixed PSNR ↑ (dB)	Mixed SSIM ↑	Normals MAE ↓ (°)
bust	all	**37.95**	**0.999**	32.29	**0.984**	35.04	**0.964**	**0.3453**
	no smoothness loss	37.64	**0.999**	32.05	0.983	34.74	0.963	0.4322
	no theta loss	37.88	**0.999**	**32.39**	**0.984**	**35.14**	**0.964**	0.3460
globe	all	**36.83**	**0.976**	**30.04**	**0.959**	**30.32**	**0.940**	**0.1035**
	no smoothness loss	36.64	0.975	30.03	0.958	30.27	0.939	0.1159
	no theta loss	36.61	0.975	30.00	0.958	30.23	0.938	0.1072

Table 5.2. **Quantitative ablation study** on PANDORA [19] synthetic image benchmark.

details of gnome's beard that is missed in reconstruction from PANDORA.

Figures 5.11 and 5.12 show the ablation study, visualising the effects of losses introduced in this chapter. As expected, theta loss plays a minor role in geometric reconstruction (see Table 5.2 for comparison) so the results without theta loss are very similar to the results with all losses. The marginal changes in reflectance renderings, as measured by PSNR and SSIM, are too small and thus hard to observe with bare human eyes. Smoothness loss, on the other hand, help reducing texture baking. For instance, the undesired texture at the middle of the mug (in vase dataset) becomes less visible when

the model is trained with smoothness loss.

Benefiting from hash-grid [62] and proposal network [6], our model converges rapidly within 30 minutes on NVIDIA A40 (compared to 15 hours of NeuS trained on NVIDIA RTX2080Ti). We show the convergence behavior in Fig. 5.15. A rough structure of the subject is formed within the first 350 iterations. The rendered normal map at 5k iteration is similar to that at 20k, implying the convergence point. To demonstrate the benefit of training on raw measurements, we conduct the ablation shown in Fig. 5.16. Demosaicing unintentionally mixes diffuse and specular radiances by spreading specularities into diffuse regions and vice versa. This leads to a specular artefact around the chin. Moreover, the model trained on raw measurement preserves high-frequency details which are blurred when compared to a counterpart trained on demosaiced images.

5.5.2 Under uniform illumination (lightstage)

For lightstage capture, we use CPFA cameras from 15 viewpoints simultaneously providing a sparser input than for the static objects above. Uniform spherical illumination is provided by a geodesic dome comprising 160 nodes each supporting 9 LEDs set at full brightness. We extract meshes from the reconstructed SDF as follows. First, we construct an oriented point cloud by projecting all pixels by their expected termination depth [15] (filtering pixels with low accumulation values) and refining to ensure they lie on the zero level set. We evaluate the SDF gradient to determine normals. Second, we reconstruct a mesh using Poisson surface reconstruction [34]. Finally, we transfer the diffuse and specular albedos to texture maps using xatlas.

As shown in Fig. 5.13, we demonstrate results for three faces. Column 1 shows one raw CPFA view for each face. Columns 2 – 4 present the learned decomposition: surface normals, diffuse radiance, and specular radiance, respectively. In Column 5, we render the recovered meshes with their material maps under HDR light-probe illumination[81] using Blender, demonstrating accurate geometry, albedo, and specular response. Figure 5.14 compares our reconstruction with that of a desktop-based facial-capture system [41]. Pose and grooming of the participant may vary due to different capture days. To

make a fair comparison, we omit the photometric-normal refinement [41], and only render both meshes with their diffuse maps and geometric normals. Across the three images (raw mesh, textured mesh, and HDR-lit render), our method matches the competing setup in geometric details while requiring only a single-shot polarisation capture.

In conclusion, in this chapter, we address the practicality of P-NeRF training (chapter 4). This is particularly important when training data is imperfect *e.g.* being full of noises and having saturation. Even for synthetic data which gives a reasonably good result, we also observe a numerical improvement. So the losses discussed in this chapter could be universally applied - no matter how we collect the data - making the training effective and thus giving the desired results. Lastly, we demonstrate an application of PP-NeRF in a lightstage setting.

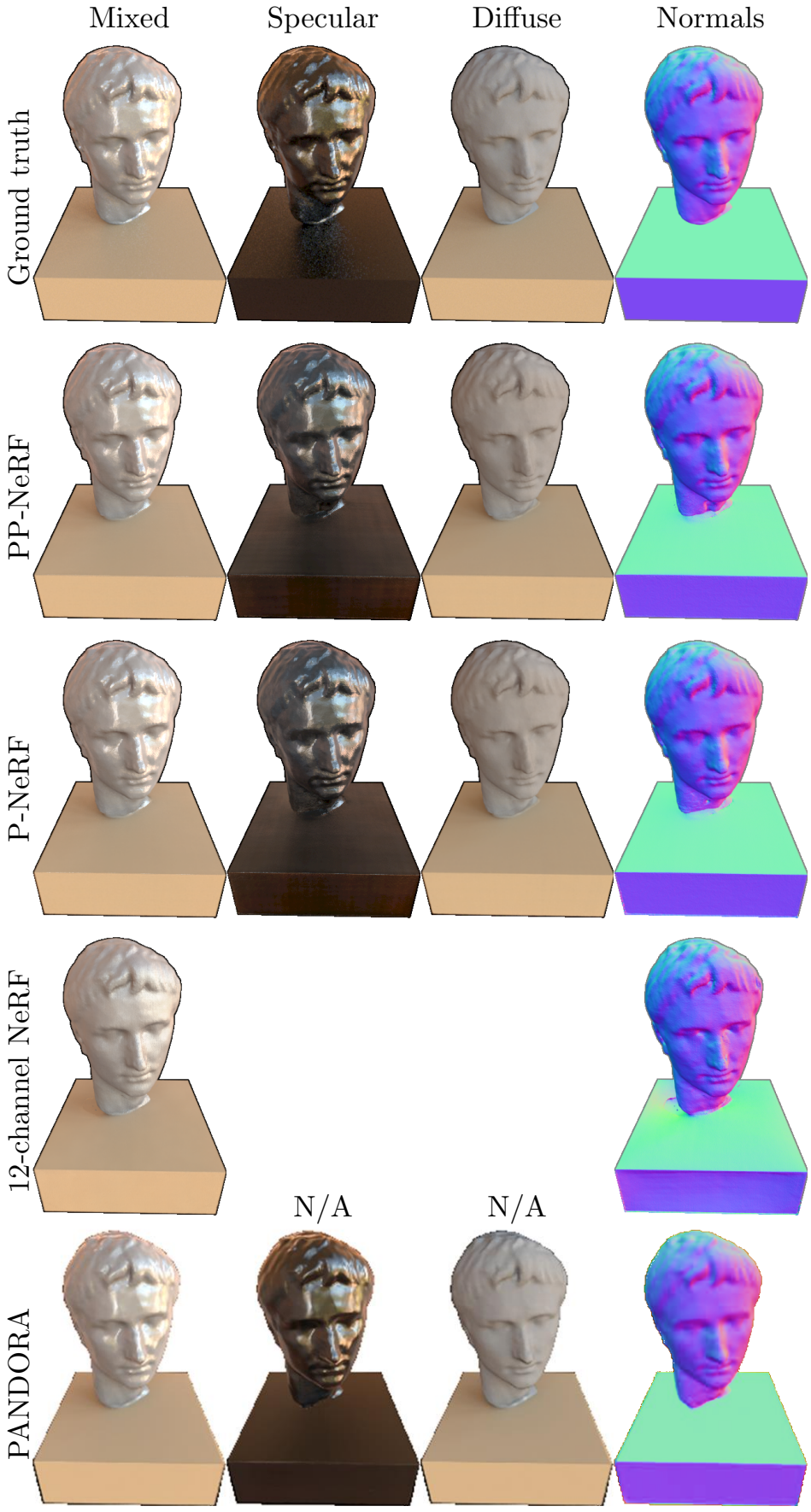


Figure 5.6. **Reflectance decomposition and geometry estimation against ground truth of (synthetic) bust data.**

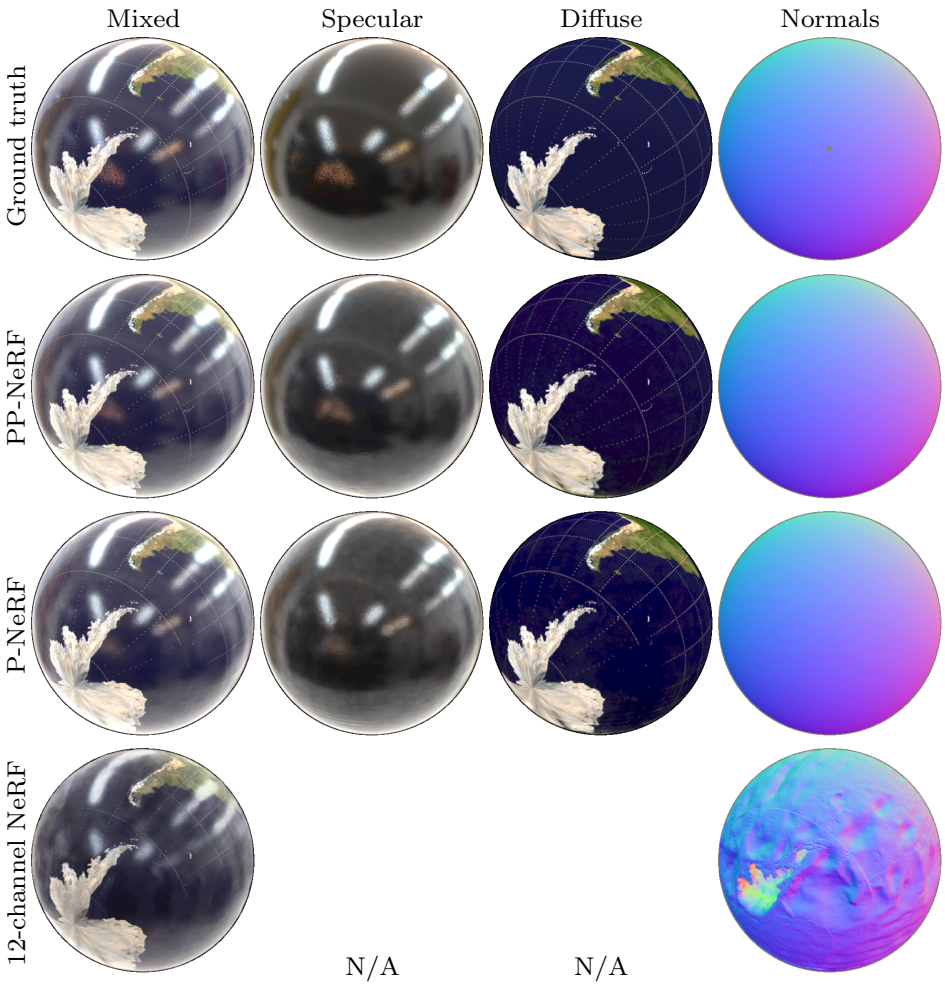


Figure 5.7. **Reflectance decomposition and geometry estimation against ground truth of (synthetic) globe data.**

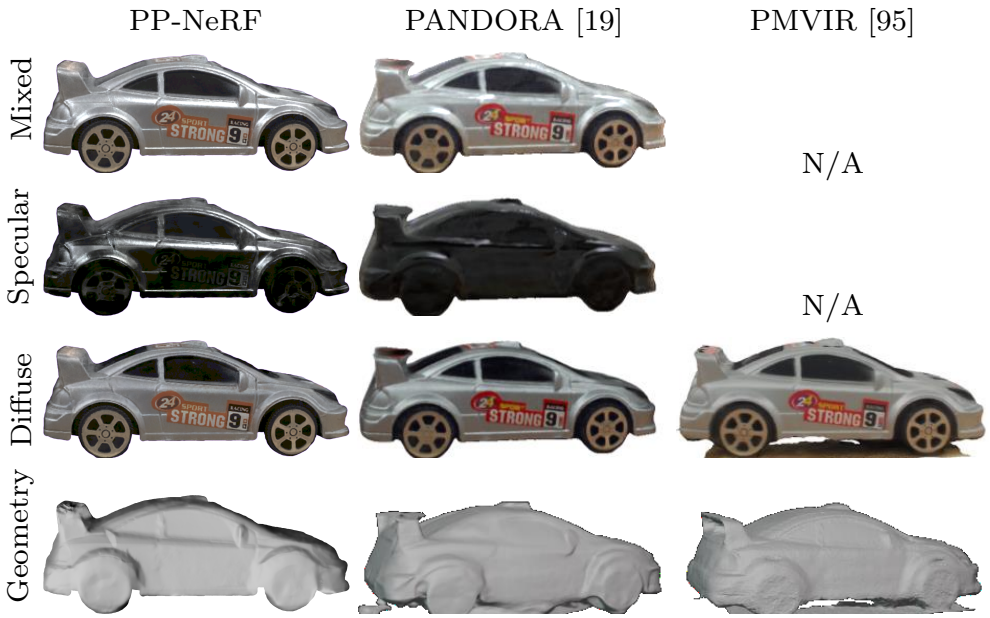


Figure 5.8. **Reflectance decomposition and geometry estimation on (real-world) car data.**

Figure 5.9. **Reflectance decomposition and geometry estimation on (real-world) gnome data.**

Figure 5.10. **Reflectance decomposition and geometry estimation on (real-world) vase data.**

Figure 5.11. **Ablation study for (synthetic) bust data.**

Figure 5.12. **Ablation study for (real-world) vase data.**

Figure 5.13. Three facial **subjects with different skin-tones**. Columns: (1) raw CPFA view, (2) predicted surface normals, (3) diffuse radiance, (4) specular radiance, and (5) mesh re-rendered with the recovered material maps under a distinct HDR light probe for each subject [81]. All images are tonemapped for display.

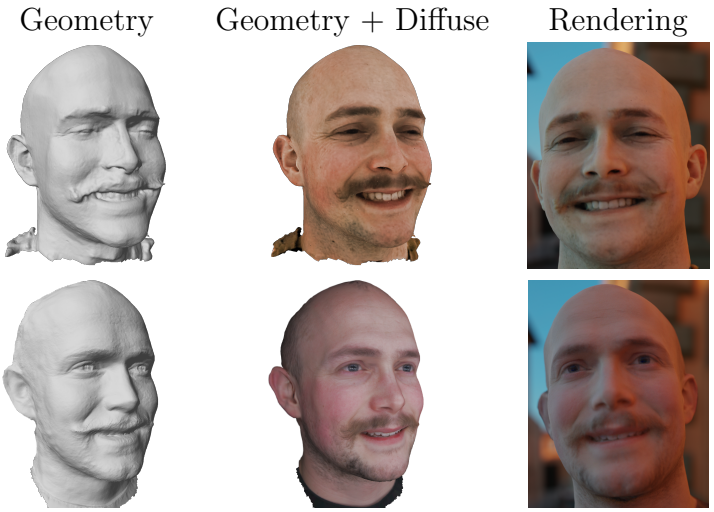


Figure 5.14. **Qualitative comparison** of the same subject captured on two different days. Top row: our pipeline; bottom row: Lattas et al.[41] Column 1 shows the mesh geometry, Column 2 the mesh shaded with the estimated diffuse albedo, and Column 3 a Blender render under the *Pisa Courtyard* HDR probe[81]. Both renders use geometric normals only; the photometric normals of [41] are disabled. Despite slight pose and appearance changes, the two reconstructions exhibit comparable geometric fidelity.

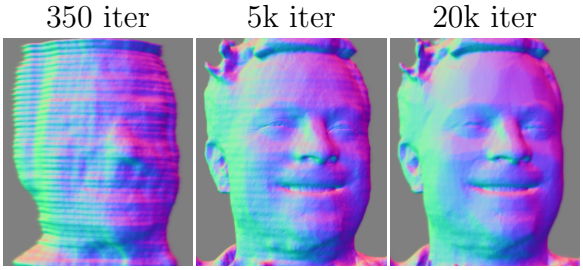


Figure 5.15. **Convergence rate**: geometry after 350 iterations (∼2 minutes), 5k iterations (30 minutes) and 20k iterations (2 hours).

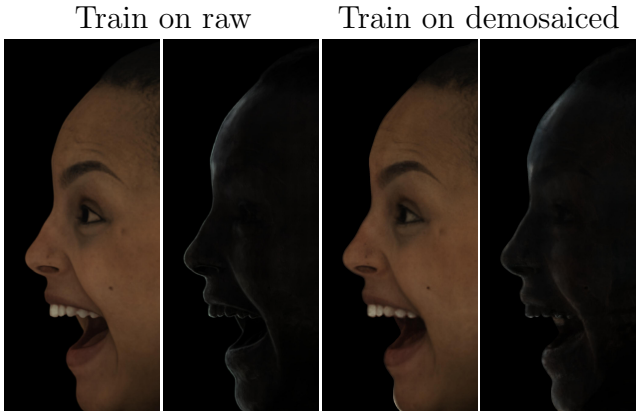


Figure 5.16. **Demosaicing ablation**: On the left we show diffuse and specular radiance for our method. On the right we show an ablation where we first demosaic and then train our method on all channels. Zoom to see blurring artefacts.

6 Conclusions

> Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning.
>
> *Winston Churchill*

6.1 Summary

Through this work, we have seen the gradual development of how polarisation information could improve 3D reconstruction, from using extra information in black box model to the constraints needed to achieve state-of-the-art result.

We begin this thesis with a literature review in which we compile interesting work in related areas. The discussion is structured into 3 topics including shape from polarisation, diffuse-specular separation, and neural inverse rendering. We hope these help the readers understanding the importance of each topic as well as seeing what have been done so far to tackle challenges in the field.

We start off the Neural Polarised Radiance Fields chapter by laying the foundation of Neural Radiance Fields (NeRF). NeRF represents a scene, using 2 networks: spatial and directional MLPs. The NeRF result shows many promises including reasoning occlusions, knowing the lighting direction and understanding material properties - hence inspiring us to work with coordinate-based architecture. Inside NeRF, we discovered a geometric bias which gives a surface prediction before the casted ray hitting the physical

surface. The bias was resolved by replacing traditional weight function with logistic density distribution [84]. Then we accelerate the training pipeline with proposal network and hash encoding. Finally we investigate the data format obtained from polarisation camera sensor, and apply NeRF on raw polarisation images.

Fresnel Equations describe the relationship between surface normal and polarisation state of light reflected off the surface. This relation has played a crucial role in Shape-from-Polarisation work for decades. In Polarised Neural Radiance Fields chapter, borrowing physical insights from the equations, we combine diffuse and specular polarisation models into a mixed polarisation model. We show that, to resolve the normal ambiguity, we need polarimetric information from at least 2 viewpoints. Then we make observations on the factors contributing to object appearance before parameterising diffuse and specular MLPs. In essence, via the mixed polarisation model, we capture the dependence between surface normal, camera pose and unpolarised radiances to predict polarised radiance.

Even though Polarised Neural Radiance Fields give reasonably good results, we realise that there are practical issues needed to be tackled. We address those details in Practical Polarised Neural Radiance Fields (PP-NeRF) or chapter 5. In particular, we propose a new color loss which suits saturated HDR data, a theta loss which encourages SDF network to produce zenith angle in the valid range, and lastly a smoothness loss which reduces the effect of noisy capture getting baked into geometry. After implementing all the losses, our model generalises to wider datasets as well as achieves the state-of-the-art result on PANDORA benchmark.

6.2 Future Work

6.2.1 Material diversification

The way we develop our mixed polarisation model narrows down the type of materials that could be used in the pipeline. In particular, our model works with dielectric materials such as porcelain and human skin. This imposes limitations which hinder the wider adaptation of this work. We would like

to expand the scope of possible materials, considering different underlying physical phenomena and including it into our framework.

In addition to material behavior, we also set a firm assumption of universally known refractive index. We would like to output spatially-varying refractive index. This could be done with either extra output of existing neural networks or constraints imposed by physical laws.

6.2.2 Increase/decrease in constraint on incident light

Our P-NeRF and PP-NeRF only work when the incident light is unpolarised. While this is true in general, there could be chances where the assumption is violated. For example, an object captured nearby a lake would get specular reflection from the lake surface and the incident lighting becomes partically polarised. Realising this, we want to relax the assumption set on environmental light. Considering multi-bounced light, NeISF [43] relaxes the assumption of unpolarised incident light, showing less baking texture compared to PANDORA [19] which assumes single-bounced illumination.

On the other hand, setting stronger assumption could lead to meaningful result. Smith *et al.* [75] exploit constant illumination provided by lightstage to estimate albedo maps. When our object is lit under such environment, our estimated diffuse reflectance becomes diffuse albedo - an intrinsic property of material, being independent of the scene.

Last but not least, setting reasonable assumptions and making sure that the assumptions are satisfied, are crucial to ensure the accuracy of results. Li *et al.* [45] show that even randomly polarised light projected on the scene could alter the prediction of polarisation-based methods.

6.2.3 Scene representations

While SDF is applied in our work, there are various techniques to represent a scene namely depth maps, surface normals, voxel grids, point clouds and meshes. These possess their own properties and require different amount of memory and processing power. In general, points and meshes are the most common representation due to its explicitness and short rasterising time. However, neither of them is differentiable and thus does not work with

neural optimasation (our framework). So, without knowing what task we are tackling, it is almost impossible to say which representation is the best.

We started our experiments right after the boom of NeRF. Since NeRF demonstrates the ability to understand the scene geometry and material inside the scene, we use NeRF as a baseline and incorporate polarimetric insights on top. With the popularity of NeRF, we have enjoyed different aspects of following-up studies *e.g.* Ref-NeRF [82] using reflected view to ease the MLP interpolation, instant-NGP [62] exloiting hash encoding to boost memory efficiency as well as fasten the training time, and proposal network offered by Mip-NeRF 360 [6].

In addition to NeRF variations, which modify minor elements inside NeRF, recent works have explored a wider picture of scene representation - providing alternative baselines for future researchers to work on 3D reconstruction. One of the highlights is Gaussian splatting [36] which represents the scene with 3D Gaussian, preserving desirable properties of continuous volumetric radiance fields for optimization while avoiding unnecessary computation in the empty space. Due to this concept, the authors achieve state-of-the-art for real-time scene rendering (measured in PSNR) in only 51 minutes of training. Neverthelss, popping artifacts usually occur when the order of primitives changes. Addressing this issue, EVER [50] positions between Gaussian splatting and NeRF - parameterising the scene with a collection of anisotropic primitives while allowing 3D consistent volume rendering. In particular, the scene is represented using ellipsoid which is fully characterised by a rotation and scale *i.e.* similar to Gassian representation. While being slower than Gassian-based representation, EVER is reasonably fast and can acheive framerate of 30 FPS at 720p on a consumer-grade gpu.

One of the main issues with original NeRF [55] is fuzzy surface obtained from density fields. While we decided to adopt NeuS for surface reconstruction *i.e.* SDF with unbiased weight function, Binary Opacity Grids or BOG [67] is also an interesting alternative. The idea is to employ a discrete opacity grid representation and apply a binary entropy loss to opacity values, encouraging them to be either zero or one. To avoid floating artifacts when converting occupancy grid to triangle mesh, volumetric fusion [16, 17] is used. BOG is considered to be state-of-the-art for surface-based rendering, bridging

the gap between volume-based and surface-based methods.

Even though mesh is a discrete geometry which, by nature, is not differentiable (unlike implicit representations such as SDF), Sivaram *et al.* [22] present Neural Geometry Fields, representing surface using quadrangular patches and surface details using coordinate neural network by displacing the patches. The traditional triangular mesh is extracted from neural geometry field by sampling the displacement. This method reduces the memory footprint of meshes without compromising on surface details.

6.2.4 Scene acquisition protocol

Today, cameras are surrounded us and many people start learning to capture images as a recreational activity. As we enter a scientific world, there are certain specifications we need in an image more than just aesthetic purposes. However, we did not find a scientific description of how a scene should be captured.

The lack of such a description creates problems in various forms: the saturation addressed in chapter 5, the scene coverage discussed in section 6.2.8, the suggestion to have an ArUco board in the scene [84] or the capturing distance which helps NeRF achieve a good result [5]. Having a descriptive procedure which does not rely on the person behind the camera, would not only improve the quality but also create a wider variety of datasets. Ideally, we wish to see large datasets that could be described with a few parameters in the suggested capturing procedure.

6.2.5 Different normals for different wavelengths/techniques

Currently we output 1 normal map for 1 viewpoint. This is true under our assumptions. However, taking into account the properties of light, technically we should differentiate the normals obtained from different wavelengths [49]. Particularly, the blue light with short wavelength would scatter close to surface while the red light with longer wavelength would get scattered deeper into the surface. This nature of light gives relatively sharper normal maps when acquired from blue light (compared to red light).

Moreover, due to different phenomenas of diffuse and specular reflections, we should also add specular normals on top of 3 RGB normals. Although 1 normal could well serve our purpose of 3D reconstruction, specific normals for each phenomena/color would improve plausibility when rendering with physics-based engine.

6.2.6 Wider range of wavelength

In all of our experiments, we do not care whether the properties of material would change or not under our lighting environment. Nonetheless, in certain circumstances, the material properties could be important and we want to preserve the quality of the captured object. A basic example is food where vitamin C decays at different rate under different lighting [98]. In the abundance of alternatives, food might not sound appealing, considering the needs of special equipment. However, when dealing with valuable assets (*e.g.* painting by renowned artist) or museum collections (*e.g.* stuffed animal); fading, discoloration and embrittlement should be avoided at all costs, and different range of light could be an option to capture these objects of high value.

In addition to preserving material properties, the extended range of wavelength could be applied to transparent objects (to human eyes). For instance, a marine biologist could estimate the shape of Barreleyes (transparent fish in the deep ocean) through our technique with the light in different range. Our methods also fail when encountering an emissive object *e.g.* glow stick in concerts and dance clubs. Again, avoiding the visible light and instead capturing the light in different range could work well in such scenario.

Last but not least, Thermal-NeRF [90] shows a better performance (compared to NeRF) when dealing with infrared rather than visible light. As a result, we are interested to see the performance of our models when expanding the captured range of light. Especially, as our methods employ 1 channel per pixel, we are keen to see the tradeoff between sparse input and rich color information. This insight might lead to different design of future camera sensor in specific domains.

6.2.7 Polarimetric filter array

Since the invention of photography, over the centuries, we have seen the continuous development of cameras from sophisticated experiments of chemical substances in the early days, to the digital devices in the present. One aspect of this advancement is the color filter array which gives color to a black-and-white image (after demosicing). There are a number of filters being developed:

- **Bayer filter** was invented by Bryce Bayer in 1974. The entire array is spreaded over 2x2 pixels accounted for 25% red, 50% green and 25% blue, or often known as RGGB.

- **RGBE** replaces the green in Bayer filter with emerald denoted by "E". Sony, the developer, claimed that the fourth color was used to reduce color reproduction errors. Nevertheless, this filter only made one appearance in Sony Cyber-Shot DSC-F828, indicating a hidden flaw and becoming obsolete.

- **Fujifilm X-Trans** was developed by Fujifilm and have been used in Fujifilm X series. Unlike traditional 2x2 pattern, Fujifilm X-Trans features a unique 6x6 pattern of photosites. The array is claimed to minimise moire artefacts.

- **Quad Bayer** was introduced by Sony. The Bayer 4x4 pattern, where the 2x2 adjacent pixels are the same color, improves cameras' performance in the dark. The 2x2 group could be processed together, essentially behaving as a larger pixel, hence having more light on the sensor.

Today, we only see the first generation of polarisation cameras. Similarly to color filter array, polarisation filter array would encounter differentiations to suit particular tasks. In our case, we would like to run similar experiments with different polariser arrays to see its effect on 3D reconstruction.

Regardless of technology advancement, the design of camera sensors generally involves a trade-off between richer information on the image and lower

image resolution. Demosaicing is the algorithm to restore the lost information that was not captured by camera sensor. As machine learning becomes mature, recent demosaicing methods start learning from image collections and create prior to predict the missing information in the arranged filter. Kurita *et al.* [39], using a sparse polarisation sensor, propose Stokes Network Architecture which consists of a refinement for RGB images and compensation network for polarisation information. By allowing the non-polarised pixels to gain more exposure and augmenting Stokes Vectors with compensation network, the sensor output achieves higher peak signal-to-noise ratio as well as lower angular error in AoLP, when compared to the output from conventional polarisation sensors.

6.2.8 Appropriate metrics

Since one of our primary goals is 3D reconstruction, it is crucial to report the accuracy of reconstructed mesh. However, we show none of them in this thesis. Why? The metrics designed to compare 2 meshes would mislead the readers when the object is only observed from some viewpoints *i.e.* some parts of the object are not captured in the dataset. Unless coupling with 3D inpainting task, we decided to report only the error in normal maps *e.g.* Mean Angular Error (MAE), and not mesh metrics *e.g.* Hausdorff Distance (HD).

Nevertheless, metrics are informative when correctly applied. For example, both PSNR and SSIM are metric for image but each of them serves different purpose: PSNR checking absolute error, whereas SSIM reporting visual similarity. Therefore, in the future, we wish to see a mesh metric that conditions on scene coverage rather than the whole mesh. Seitz *et al.* [71] provides a discussion on mesh evaluation, which is categorised into the accuracy and completeness. Instead of detecting and removing the error due to incomplete mesh, the authors' suggestion is to augment the mesh to the completion *i.e.* a hole-filled mesh, and then exclude the augmented parts when calculating the accuracy metric.

Bibliography

[1] G. Atkinson and E. Hancock, "Recovery of surface orientation from diffuse polarization," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1653–1664, 2006.

[2] ——, "Recovery of surface orientation from diffuse polarization," *Image Processing, IEEE Transactions on*, vol. 15, pp. 1653 – 1664, 07 2006.

[3] Y. Ba, A. Gilbert, F. Wang, J. Yang, R. Chen, Y. Wang, L. Yan, B. Shi, and A. Kadambi, "Deep shape from polarization," in *European Conference on Computer Vision*. Springer, 2020, pp. 554–571.

[4] S.-H. Baek, T. Zeltner, H. J. Ku, I. Hwang, X. Tong, W. Jakob, and M. H. Kim, "Image-based acquisition and modeling of polarimetric reflectance," *ACM Transactions on Graphics (Proc. SIGGRAPH 2020)*, 2020.

[5] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," *ICCV*, 2021.

[6] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," *CVPR*, 2022.

[7] S. Bi, Z. Xu, P. P. Srinivasan, B. Mildenhall, K. Sunkavalli, M. Havsan, Y. Hold-Geoffroy, D. J. Kriegman, and R. Ramamoorthi, "Neural reflectance fields for appearance acquisition," *ArXiv*, vol. abs/2008.03824, 2020.

[8] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. P. Lensch, "Nerd: Neural reflectance decomposition from image collections," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[9] M. Boss, V. Jampani, R. Braun, C. Liu, J. T. Barron, and H. P. Lensch, "Neural-pil: Neural pre-integrated lighting for reflectance decomposition," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[10] J. Cao, Z. Yuan, T. Mao, Z. Wang, and Z. Li, "Nerf-based polarimetric multi-view stereo," *Pattern Recognition*, vol. 158, p. 111036, 2025.

[11] X. Cao, H. Santo, F. Okura, and Y. Matsushita, "Multi-view azimuth stereo via tangent space consistency," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[12] J. Y. Chang, K. M. Lee, and S. U. Lee, "Multiview normal field integration using level set methods," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[13] J. Chibane, T. Alldieck, and G. Pons-Moll, "Implicit functions in feature space for 3d shape reconstruction and completion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020.

[14] Z. Cui, J. Gu, B. Shi, P. Tan, and J. Kautz, "Polarimetric multi-view stereo," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 369–378.

[15] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 303–312.

[16] ——, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 303–312. [Online]. Available: https://doi.org/10.1145/237170.237269

[17] ——, *A Volumetric Method for Building Complex Models from Range Images*, 1st ed. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3596711.3596726

[18] A. Dave, Y. Hold-Geoffroy, M. Hašan, K. Sunkavalli, and A. Veeraraghavan, "Snapshot polarimetric diffuse-specular separation," *Opt. Express*, vol. 30, no. 19, pp. 34 239–34 255, Sep 2022. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-30-19-34239

[19] A. Dave, Y. Zhao, and A. Veeraraghavan, "Pandora: Polarization-aided neural decomposition of radiance," *arXiv preprint arXiv:2203.13458*, 2022.

[20] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar, "Acquiring the reflectance field of a human face," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00. USA: ACM Press/Addison-Wesley Publishing Co., 2000, p. 145–156. [Online]. Available: https://doi.org/10.1145/344779.344855

[21] B. Duvenhage, K. Bouatouch, and D. Kourie, "Numerical verification of bidirectional reflectance distribution functions for physical plausibility," 10 2013, pp. 200–208.

[22] V. Edavamadathil Sivaram, T.-M. Li, and R. Ramamoorthi, "Neural geometry fields for meshes," in *ACM SIGGRAPH 2024 Conference Papers*, ser. SIGGRAPH '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3641519.3657399

[23] H. Fan, H. Su, and L. Guibas, "A point set generation network for 3d object reconstruction from a single image," 07 2017, pp. 2463–2471.

[24] A. Ghosh, T. Chen, P. Peers, C. A. Wilson, and P. Debevec, "Estimating Specular Roughness and Anisotropy from Second Order Spherical Gradient Illumination," *Computer Graphics Forum*, 2009.

[25] A. Ghosh, G. Fyffe, B. Tunwattanapong, J. Busch, X. Yu, and P. Debevec, "Multiview face capture using polarized spherical gradient illumination," in *Proceedings of the 2011 SIGGRAPH Asia Conference*, 2011, pp. 1–10.

[26] G. C. Guarnera, A. Ghosh, I. Hall, M. Glencross, and D. Guarnera, "Material capture and representation with applications in virtual reality," in *ACM SIGGRAPH 2017 Courses*, ser. SIGGRAPH '17. New York, NY, USA: Association for Computing Machinery, 2017.

[27] G. C. Guarnera, P. Peers, P. Debevec, and A. Ghosh, "Estimating surface normals from spherical stokes reflectance fields," in *Computer Vision – ECCV 2012. Workshops and Demonstrations*. Springer Berlin Heidelberg, 2012, pp. 340–349.

[28] S. Hadadan, S. Chen, and M. Zwicker, "Neural radiosity," *ACM Trans. Graph.*, vol. 40, no. 6, dec 2021. [Online]. Available: https://doi.org/10.1145/3478513.3480569

[29] Y. Han, H. Guo, K. Fukai, H. Santo, B. Shi, F. Okura, Z. Ma, and Y. Jia, "NeRSP: Neural 3d reconstruction for reflective objects with sparse polarized images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 11 821–11 830.

[30] A. Kadambi, V. Taamazyan, B. Shi, and R. Raskar, "Polarized 3d: High-quality depth sensing with polarization cues," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3370–3378.

[31] J. T. Kajiya and B. P. Von Herzen, "Ray tracing volume densities," in *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '84. New York, NY, USA: Association for Computing Machinery, 1984, p. 165–174. [Online]. Available: https://doi.org/10.1145/800031.808594

[32] A. Kar, C. Häne, and J. Malik, "Learning a multi-view stereo machine," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 364–375.

[33] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[34] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, no. 4, 2006.

[35] P. Kellnhofer, L. Jebe, A. Jones, R. Spicer, K. Pulli, and G. Wetzstein, "Neural lumigraph rendering," in *CVPR*, 2021.

[36] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, July 2023. [Online]. Available: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[37] Y. Kim, W. Jin, S. Cho, and S.-H. Baek, "Neural spectro-polarimetric fields," in *SIGGRAPH Asia 2023 Conference Papers*, 2023, pp. 1–11.

[38] S. Kobayashi, E. Matsumoto, and V. Sitzmann, "Decomposing nerf for editing via feature field distillation," in *Advances in Neural Information Processing Systems*, vol. 35, 2022. [Online]. Available: https://arxiv.org/pdf/2205.15585.pdf

[39] T. Kurita, Y. Kondo, L. Sun, and Y. Moriuchi, " Simultaneous Acquisition of High Quality RGB Image and Polarization Information using a Sparse Polarization Sensor ," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Los Alamitos, CA, USA: IEEE Computer Society, Jan. 2023, pp. 178–188. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/WACV56688.2023.00026

[40] A. Lattas, M. Wang, S. Zafeiriou, and A. Ghosh, "Multi-view facial capture using binary spherical gradient illumination," in *ACM SIGGRAPH 2019 Posters*, 2019, pp. 1–2.

[41] A. Lattas, Y. Lin, J. Kannan, E. Ozturk, L. Filipi, G. C. Guarnera, G. Chawla, and A. Ghosh, "Practical and scalable desktop-based high-quality facial capture," in *European Conference on Computer Vision*. Springer, 2022, pp. 522–537.

[42] C. Lei, C. Qi, J. Xie, N. Fan, V. Koltun, and Q. Chen, "Shape from polarization for complex scenes in the wild," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2022, pp. 12 622–12 631. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01230

[43] C. Li, T. Ono, T. Uemori, H. Mihara, A. Gatto, H. Nagahara, and Y. Moriuchi, "Neisf: Neural incident stokes field for geometry and material estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 21 434–21 445.

[44] Y. Li, R. Wu, J. Li, and Y.-C. Chen, "Gnerp: Gaussian-guided neural reconstruction of reflective objects with noisy polarization priors," *ICLR*, 2024.

[45] Z. Li, Z. Zhong, S. Nobuhara, K. Nishino, and Y. Zheng, " Fooling Polarization-Based Vision Using Locally Controllable Polarizing Projection ," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE

Computer Society, Jun. 2024, pp. 24 706–24 715. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.02333

[46] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[47] S. Liu, Y. Zhang, S. Peng, B. Shi, M. Pollefeys, and Z. Cui, "Dist: Rendering deep implicit signed distance function with differentiable sphere tracing," 11 2019.

[48] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, "Neural volumes: Learning dynamic renderable volumes from images," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 65:1–65:14, Jul. 2019.

[49] W.-C. Ma, T. Hawkins, P. Peers, C.-F. Chabert, M. Weiss, and P. Debevec, "Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination." 01 2007, pp. 183–194.

[50] A. Mai, P. Hedman, G. Kopanas, D. Verbin, D. Futschik, Q. Xu, F. Kuester, J. Barron, and Y. Zhang, "Ever: Exact volumetric ellipsoid rendering for real-time view synthesis," 2024. [Online]. Available: https://arxiv.org/abs/2410.01804

[51] S. Mallick, T. Zickler, P. Belhumeur, and D. Kriegman, "Dichromatic separation: specularity removal and editing," 01 2006.

[52] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, and J. T. Barron, "NeRF in the dark: High dynamic range view synthesis from noisy raw images," *arXiv*, 2021.

[53] ——, "NeRF in the dark: High dynamic range view synthesis from noisy raw images," *CVPR*, 2022.

[54] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics (TOG)*, 2019.

[55] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[56] Miyazaki, Tan, Hara, and Ikeuchi, "Polarization-based inverse rendering from a single view," in *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003, pp. 982–987 vol.2.

[57] D. Miyazaki, M. Kagesawa, and K. Ikeuchi, "Transparent surface modeling from a pair of polarization images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, pp. 73–82, 02 2004.

[58] D. Miyazaki, T. Shigetomi, M. Baba, R. Furukawa, S. Hiura, and N. Asada, "Polarization-based surface normal estimation of black specular objects from multiple viewpoints," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*, 2012, pp. 104–111.

[59] ——, "Surface normal estimation of black specular objects from multiview polarization images," *Optical Engineering*, vol. 56, 04 2017.

[60] O. Morel, F. Meriaudeau, C. Stolz, and P. Gorria, "Polarization imaging applied to 3d reconstruction of specular metallic surfaces," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5679, 02 2005.

[61] K. Moriwaki, R. Yoshihashi, R. Kawakami, S. You, and T. Naemura, "Hybrid loss for learning single-image-based hdr reconstruction," *ArXiv*, vol. abs/1812.07134, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:56171426

[62] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[63] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob, "Mitsuba 2: a retargetable forward and inverse renderer," *ACM Trans. Graph.*, vol. 38, no. 6, Nov. 2019. [Online]. Available: https://doi.org/10.1145/3355089.3356498

[64] E. Nogue, Y. Lin, and A. Ghosh, "Polarization-imaging Surface Reflectometry using Near-field Display," in *Eurographics Symposium on Rendering*, A. Ghosh and L.-Y. Wei, Eds. The Eurographics Association, 2022.

[65] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural

networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 5301–5310. [Online]. Available: https://proceedings.mlr.press/v97/rahaman19a.html

[66] S. Rahmann and N. Canterakis, "Reconstruction of specular surfaces using polarization imaging," in *Proc. CVPR*, 2001.

[67] C. Reiser, S. Garbin, P. P. Srinivasan, D. Verbin, R. Szeliski, B. Mildenhall, J. T. Barron, P. Hedman, and A. Geiger, "Binary opacity grids: Capturing fine geometric detail for mesh-based view synthesis," *SIGGRAPH*, 2024.

[68] J. Riviere, I. Reshetouski, L. Filipi, and A. Ghosh, "Polarization imaging reflectometry in the wild," *ACM Transactions on Graphics (TOG)*, vol. 36, pp. 1 – 14, 2017.

[69] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[70] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[71] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, 2006, pp. 519–528.

[72] S. A. Shafer, *Using Color to Separate Reflection Components.* USA: Jones and Bartlett Publishers, Inc., 1992, p. 43–51.

[73] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3d-structure-aware neural scene representations," in *Advances in Neural Information Processing Systems*, 2019.

[74] W. Smith, R. Ramamoorthi, and S. Tozza, "Linear depth estimation from an uncalibrated, monocular polarisation image," vol. 9912, 10 2016, pp. 109–125.

[75] W. A. P. Smith, A. Seck, H. Dee, B. Tiddeman, J. Tenenbaum, and B. Egger, "A morphable face albedo model," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5011–5020.

[76] A. C. Souza, M. C. Macedo, V. P. Nascimento, and B. S. Oliveira, "Real-time high-quality specular highlight removal using efficient pixel clustering," in *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2018, pp. 56–63.

[77] V. Taamazyan, A. Kadambi, and R. Raskar, "Shape from mixed polarization," *ArXiv*, vol. abs/1605.02066, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:14550418

[78] R. Tan, K. Ikeuchi, and D. Miyazaki, *Separating Reflection Components of Textured Surfaces using a Single Image*, 01 2008, pp. 353–384.

[79] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, "Nerfstudio: A modular framework for neural radiance field development," in *ACM SIGGRAPH 2023 Conference Proceedings*, ser. SIGGRAPH '23, 2023.

[80] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi, T. Simon, C. Theobalt, M. Nießner, J. Barron, G. Wetzstein, M. Zollhöfer, and V. Golyanik, "Advances in neural rendering," *Computer Graphics Forum*, vol. 41, pp. 703–735, 05 2022.

[81] USC Institute for Creative Technologies, "High-resolution light probe image gallery," https://vgl.ict.usc.edu/Data/HighResProbes/, accessed 5 Jun 2025.

[82] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-NeRF: Structured view-dependent appearance for neural radiance fields," *CVPR*, 2022.

[83] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3d mesh models from single rgb images," in *ECCV*, 2018.

[84] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *NeurIPS*, 2021.

[85] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang, "Pix2vox: Context-aware 3d reconstruction from single and multi-view images," in *ICCV*, 2019.

[86] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar, "Neural fields in visual computing and beyond," *Computer Graphics Forum*, 2022.

[87] B. Yang, Y. Zhang, Y. Xu, Y. Li, H. Zhou, H. Bao, G. Zhang, and Z. Cui, "Learning object-compositional neural radiance field for editable scene rendering," in *International Conference on Computer Vision (ICCV)*, October 2021.

[88] W. Yang, G. Chen, C. Chen, Z. Chen, and K.-Y. K. Wong, "Ps-nerf: Neural inverse rendering for multi-view photometric stereo," in *European Conference on Computer Vision (ECCV)*, 2022.

[89] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman, "Multiview neural surface reconstruction by disentangling geometry and appearance," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[90] T. Ye, Q. Wu, J. Deng, G. Liu, L. Liu, S. Xia, L. Pang, W. Yu, and L. Pei, "Thermal-nerf: Neural radiance fields from an infrared camera," 2024. [Online]. Available: https://arxiv.org/abs/2403.10340

[91] L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, C. Liu, M. Liu, Z. Liu, Y. Lu, Y. Shi, L. Wang, J. Wang, B. Xiao, Z. Xiao, and P. Zhang, "Florence: A new foundation model for computer vision," 11 2021.

[92] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. Snavely, "PhySG: Inverse rendering with spherical gaussians for physics-based material editing and relighting," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[93] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron, "Nerfactor: Neural factorization of shape and reflectance under an unknown illumination," *ACM Trans. Graph.*, vol. 40, no. 6, dec 2021. [Online]. Available: https://doi.org/10.1145/3478513.3480496

[94] J. Zhao, Y. Monno, and M. Okutomi, "Polarimetric multi-view inverse rendering," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part*

XXIV. Berlin, Heidelberg: Springer-Verlag, 2020, p. 85–102. [Online]. Available: https://doi.org/10.1007/978-3-030-58586-0_6

[95] ——, "Polarimetric multi-view inverse rendering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 7, pp. 8798–8812, 2023.

[96] J. Zhao, J. Oishi, Y. Monno, and M. Okutomi, "Polarimetric patch-match multi-view stereo," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 3464–3472.

[97] D. Zhu and W. Smith, "Depth from a polarisation + rgb stereo pair," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2019, pp. 7578–7587.

[98] S. Çakmakçı and T. Turgut, "Influence of different light sources, illumination intensities and storage times on the vitamin c content in pasteurized milk," *Turkish Journal of Veterinary and Animal Sciences*, vol. 29, pp. 1097–1100, 10 2005.