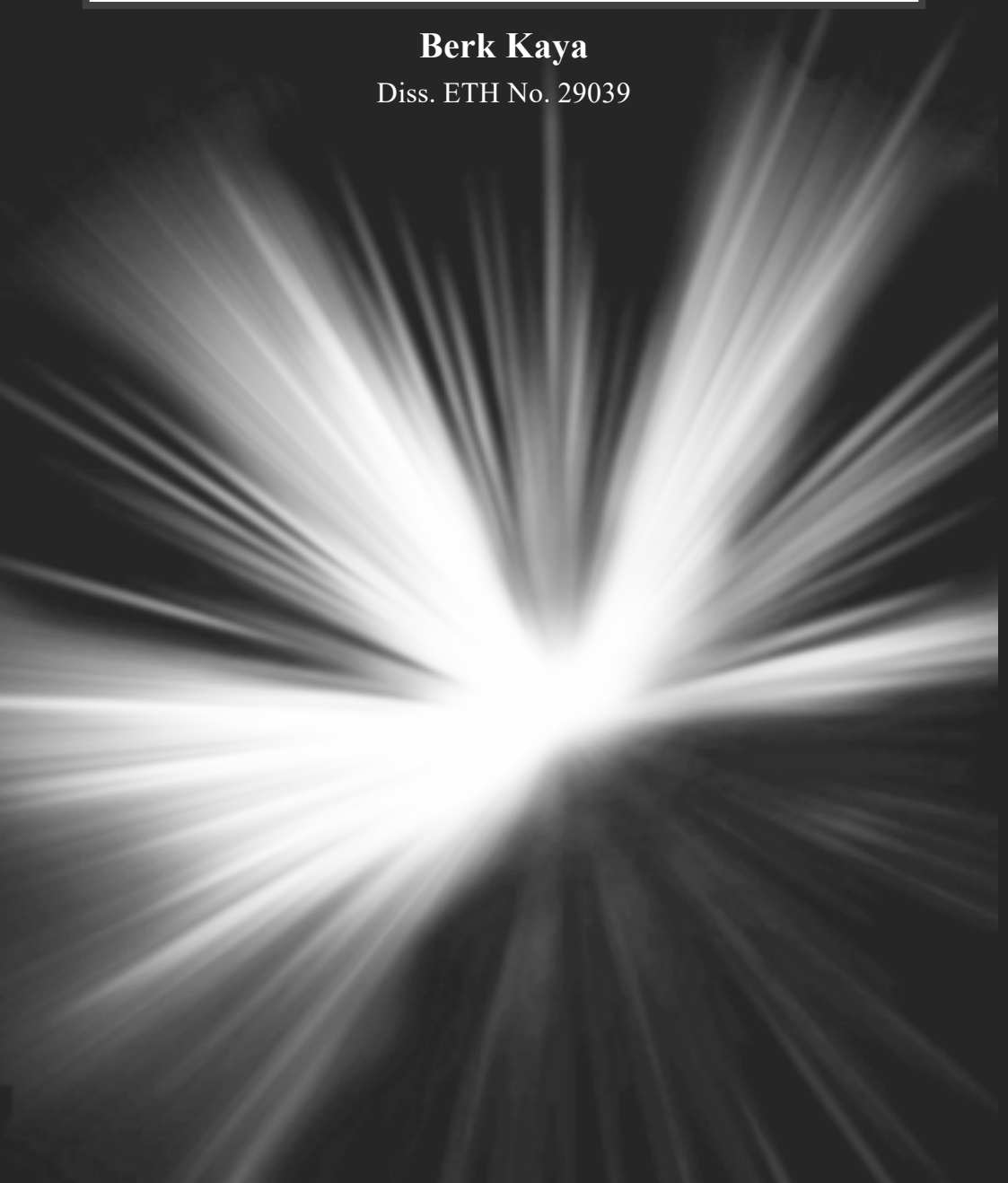


A MODERN TAKE ON PHOTOMETRIC
3D-RECONSTRUCTION

Berk Kaya

Diss. ETH No. 29039



DISS. ETH NO. 29039

A MODERN TAKE ON PHOTOMETRIC
3D-RECONSTRUCTION

A dissertation submitted to attain the degree of
DOCTOR OF SCIENCES OF ETH ZURICH

presented by

BERK KAYA

Master of Science in Information Technology and Electrical
Engineering (ETH Zurich)

born on 1 Jan 1996
citizen of Turkey

accepted on the recommendation of

Prof. Dr. Luc Van Gool, examiner
Prof. Dr. Daniel Cremers, co-examiner
Prof. Dr. Satoshi Ikehata, co-examiner

2023

To my family

ABSTRACT

In this thesis, modern approaches for precise and accurate 3D acquisition are proposed. It is well-known that photometric techniques are exceptional at capturing fine details of the object’s surface by exploiting the variations in the shading. Therefore, it is helpful for many computer vision tasks and other cutting-edge scientific disciplines such as metrology, geometry processing, forensics, etc. However, existing methods depend on strong assumptions about the surface and require calibrated acquisition setups, restricting the practical application in real-world settings. This thesis takes a modern approach to improve the applicability and performance of photometric 3D reconstruction techniques.

Photometric stereo uses multiple images of the object under varying illumination conditions to recover its fine surface details. Existing neural network-based methods for this task either require exact light directions or training data with ground-truth surface normals. However, in practice, it is challenging to procure such information precisely. To bypass this difficulty, this thesis proposes an uncalibrated neural inverse rendering pipeline. By utilizing light physics and modeling global illumination effects, the proposed approach handles challenging objects, composed of convex and concave regions.

Next, this thesis addresses the multi-view photometric stereo problem for the dense reconstruction of objects. Existing classical methods for this task require several complicated steps and can handle only certain material types. Our work introduces the first modern approach to the problem. To this end, it initially presents a view synthesis method to combine photometric stereo predictions with multi-view information. Subsequently, it proposes uncertainty modeling for a reliable fusion of surface estimates. Finally, it presents an uncertainty-aware volume rendering approach to handle surfaces with non-typical reflectance properties.

Lastly, this thesis presents an automated machine-learning approach for uncalibrated photometric stereo. For that, it leverages the differentiable neural architecture search methodology. The proposed solution considers the task-specific constraints of the problem explicitly to search for an optimal architecture. The proposed approach provides lightweight network architectures that perform better than existing networks, which are hand-crafted and carefully tuned.

ZUSAMMENFASSUNG

In dieser Arbeit werden moderne Ansätze vorgestellt, um präzise und genaue 3D-Erfassung zu ermöglichen. Es ist allgemein bekannt, dass photometrische Techniken hervorragend geeignet sind, um feine Details der Oberfläche von Objekten durch Ausnutzung der Variationen in der Schattierung zu erfassen. Daher sind sie für viele Computer-Vision-Aufgaben und andere hochmoderne wissenschaftliche Disziplinen wie Metrologie, Geometrieverarbeitung, Forensik usw. hilfreich. Die vorhandenen Methoden setzen jedoch starke Annahmen über die Oberflächenbeschaffenheit voraus und erfordern kalibrierte Einstellungen für die Erfassung, was die praktische Anwendung in realen Umgebungen einschränkt. Diese Arbeit verfolgt einen modernen Ansatz, um die Anwendbarkeit und Leistungsfähigkeit photometrischer 3D-Rekonstruktionsmethoden zu verbessern.

Photometrisches Stereo nutzt mehrere Bilder des Objekts unter unterschiedlichen Beleuchtungsbedingungen, um dessen feine Oberflächendetails wiederherzustellen. Vorhandene auf neuronalen Netzen basierende Methoden für diese Aufgabe erfordern entweder genaue Lichtrichtungen oder Trainingsdaten mit Boden-Truth-Oberflächennormalen. In der Praxis ist es jedoch schwierig, solche Informationen präzise zu beschaffen. Um diese Schwierigkeit zu umgehen, schlägt diese Arbeit eine unkalibrierte neuronale Inverse Rendering-Pipeline vor. Durch Nutzung der Lichtphysik und Modellierung globaler Beleuchtungseffekte bewältigt der vorgeschlagene Ansatz herausfordernde Objekte, die aus konvexen und konkaven Regionen bestehen.

Als Nächstes wird in dieser Arbeit das Problem des Multi-View Photometric Stereo zur dichten Rekonstruktion von Objekten angegangen. Vorhandene klassische Methoden für diese Aufgabe erfordern mehrere komplizierte Schritte und können nur bestimmte Materialtypen handhaben. Unsere Arbeit stellt den ersten modernen Ansatz für das Problem vor. Zu diesem Zweck präsentiert sie zunächst eine View-Synthesis-Methode, um Photometrische Stereo-Vorhersagen mit Multi-View-Informationen zu kombinieren. Anschließend schlägt sie eine Unsicherheitsmodellierung für eine zuverlässige Fusion von Oberflächenschätzungen vor. Schließlich präsentiert sie einen Unsicherheits-bewussten Volumen-Rendering-Ansatz, um Oberflächen mit nicht-typischen Reflexionseigenschaften zu handhaben.

Schließlich stellt diese Arbeit einen automatisierten Machine-Learning-Ansatz für unkalibriertes Photometrisches Stereo vor. Dazu wird die Methode der differenzierbaren neuronalen Architektursuche genutzt. Die vorgeschlagene Lösung berücksichtigt die aufgabenspezifischen Einschränkungen des Problems explizit, um nach einer optimalen Architektur zu suchen. Der vorgeschlagene Ansatz liefert leichte Netzwerkarchitekturen, die besser abschneiden als vorhandene Netzwerke, die von Hand erstellt und sorgfältig abgestimmt wurden.

PUBLICATIONS

The following publications are included as a whole or in parts in this thesis:

1. Kaya, B., Kumar, S., Oliveira, C., Ferrari, V. & Van Gool, L. *Uncalibrated Neural Inverse Rendering for Photometric Stereo of General Surfaces* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), 3804.
2. Kaya, B., Kumar, S., Sarno, F., Ferrari, V. & Van Gool, L. *Neural Radiance Fields Approach to Deep Multi-View Photometric Stereo* in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2022), 1965.
3. Sarno, F., Kumar, S., Kaya, B., Huang, Z., Ferrari, V. & Van Gool, L. *Neural Architecture Search for Efficient Uncalibrated Deep Photometric Stereo* in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2022), 361.
4. Kaya, B., Kumar, S., Oliveira, C., Ferrari, V. & Van Gool, L. *Uncertainty-Aware Deep Multi-View Photometric Stereo* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 12601.
5. Kaya, B., Kumar, S., Oliveira, C., Ferrari, V. & Van Gool, L. *Multi-View Photometric Stereo Revisited* in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2023).

ACKNOWLEDGEMENTS

I would like to express my gratitude to Prof. Dr. Luc Van Gool for giving me the opportunity to pursue my PhD in this lab. His support and guidance have been invaluable to me throughout my PhD. I am also thankful to Prof. Vittorio Ferrari for his guidance. Without his mentorship, I could not have completed this thesis.

I am deeply grateful to Dr. Suryansh Kumar for supporting and guiding me every step of the way. His expertise, patience, and willingness to teach me how to conduct research have been essential in my development.

I would like to extend a special thanks to Yigit Baran Can for his unwavering friendship. I will miss our kicker games together. I am also grateful to my labmates Erik Sandström and Ozan Ünal for their support, encouragement, and helpful feedback throughout my studies. Their guidance has helped me to improve my work. I would like to express my appreciation to Mertalp Öcal and Recep Fırat Çekinel for their belief in me and emotional support during difficult times. Their kindness and support have given me strength when I needed it.

Finally, I would like to thank my friends, Ömer Mert Aksoy, Ali Batuhan Yardım, and Sarp Uzun, for their support, encouragement, and guidance throughout my studies. I am grateful for their friendship and for the ways they have supported me. Last but not least, I would like to express my thanks to my family for their love, encouragement, and support throughout my studies. Their love has given me the strength to achieve my goals, and I could not have done it without them.

CONTENTS

1	Introduction	1
1.1	Background	2
1.1.1	Image Formation	2
1.1.2	Photometric Stereo	4
1.1.3	Limitations	9
1.2	Thesis Outline	10
2	Uncalibrated Neural Inverse Rendering	13
2.1	Motivation	13
2.2	Contributions	14
2.3	Interreflection Model	15
2.4	Method	17
2.4.1	Light Estimation Network	18
2.4.2	Inverse Rendering Network	20
2.4.3	Robust Initialization	25
2.5	Experiments and Results	26
2.5.1	Implementation Details	27
2.5.2	Evaluation and Ablation Study	31
2.5.3	Case Study	35
2.6	Limitations and Future Extension	37
2.7	Conclusion	38
3	Multi-View Photometric Stereo	41
3.1	Motivation	41
3.2	Related Work	43
3.2.1	Multi-View Stereo (MVS) Methods	43
3.2.2	View Synthesis for 3D Reconstruction	44
3.2.3	Multi-View Photometric Stereo (MVPS) Methods	45
3.3	Preliminaries	46
3.3.1	MVPS Setup	46
3.3.2	Notation and Definitions	47
3.3.3	Benchmark	48
3.4	Neural Radiance Fields Approach to MVPS	49
3.4.1	Contributions	50
3.4.2	Method	50
3.4.3	Experiments and Results	55
3.4.4	Conclusion	61

3.5	Uncertainty-Aware Deep MVPS	62
3.5.1	Contributions	62
3.5.2	Method	63
3.5.3	Experiments and Results	70
3.5.4	Conclusion	78
3.6	Uncertainty-Aware Neural Volume Rendering	78
3.6.1	Contributions	79
3.6.2	Method	79
3.6.3	Experiments and Results	85
3.6.4	Conclusion	90
4	Neural Architecture Search for Uncalibrated Photometric Stereo	91
4.1	Motivation	91
4.2	Contributions	92
4.3	Method	92
4.3.1	Architecture Search for Uncalibrated PS	93
4.3.2	Light Calibration Network	96
4.3.3	Normal Estimation Network	98
4.4	Experiments and Results	100
4.4.1	Dataset Preparation	100
4.4.2	Implementation Details	101
4.4.3	Qualitative and Quantitative Evaluation	104
4.5	Limitations and Further Study	105
4.6	Conclusion	107
5	Discussion	109
A	Appendix	111
A.1	Uncalibrated Neural Inverse Rendering	111
A.2	Uncertainty-Aware Deep MVPS	117
A.3	Uncertainty-Aware Volume Rendering for MVPS	120
	Bibliography	123

INTRODUCTION

The acquisition of the 3D object shape has been a fundamental research problem in computer vision and industrial machine vision for several decades. Many applications in computer graphics, medical imaging, and virtual reality demand precise 3D models of real-world objects. Moreover, it is helpful for other cutting-edge scientific disciplines such as metrology [1], geometry processing [2], forensics [3], etc. To that end, several active and passive methods are proposed in the past to recover the object geometry [1, 3–5]. Among popular methods, structure-from-motion (SfM) [6, 7] and multi-view stereo (MVS) [1] are used to recover the object geometry from its images. However, it is widely accepted that they are not sufficient on their own to provide detailed and precise 3D reconstruction for all kinds of surfaces. Although other techniques such as shape from focus [8], and shape from texture [9] exist, solving the problem of estimating the geometry of an object from its images is still an open area for researchers to explore.

Another line of research in 3D data acquisition focuses on photometric techniques which is based on the interaction between surface geometry, illumination, and material. As the image acquired by the camera sensor is formed by such interactions, shading variations provide important cues about the object's shape. To recover the geometry using these cues, *Shape-from-Shading* (SfS) techniques have been developed by Horn in the early 1970s [10]. It has been shown that SfS algorithms can provide local orientation information when the problem is constrained with strong assumptions on the geometry and surface reflectance. However, its effectiveness is questionable for real-world objects, as a single image does not provide sufficient information for reliable surface recovery. Therefore, the concept of SfS is extended to multiple images with the introduction of the photometric stereo by Woodham's seminal work [11].

Photometric stereo (PS) estimates the surface normals of a static object from its light-varying images captured from a fixed viewpoint. Unlike multi-view methods, photometric stereo is excellent at recovering fine details such as indentations and scratches. Moreover, it allows the fast acquisition of 3D information using low-cost hardware. All these advantages make photometric stereo a useful approach for 3D acquisition. For this reason,

it is widely used in shape analysis, shape manipulation, image rendering, and digitization of cultural heritage.

This thesis focuses on the 3D acquisition of objects focusing on photometric stereo techniques and addresses the limitations of the existing approaches. For that, we draw inspiration from the classical algorithms and recent learning-based approaches, combining the research output of the last forty years. Specifically, we contribute in three main aspects: (i) *Applicability*: We extend the application of photometric reconstruction methods to a broader range of object types. (ii) *Performance*: We propose methods that can provide dense and detailed surface reconstructions. (iii) *Practicality*: We provide solutions which are easy to implement or execute. We begin our discussion with a background on image formation and photometric stereo. Next, we present an overview of the thesis.

1.1 BACKGROUND

This section briefly introduces the theoretical and practical aspects of photometric stereo techniques. As photometric stereo exploits the intensity variations caused by illumination changes for surface recovery, let's have a closer look at the image formation process in its most general form.

1.1.1 Image Formation

When light hits a solid surface, it is either absorbed or reflected. Assuming that surface material does not transmit light and ignoring its wave-like properties, the reflectance behavior on a particular surface type can be described by a *Bidirectional Reflectance Distribution Function* (BRDF). In general, BRDF $f_r(\theta_i, \phi_i, \theta_r, \phi_r)$ can be written in terms the incident and reflected light direction angles which are relative to the surface reference frame. For simplicity, we omit BRDF's dependency on the light wavelength. A realistic BRDF has the following properties for all incident and reflected light angles:

- *Non-negativity*: $f_r(\theta_i, \phi_i, \theta_r, \phi_r) \geq 0$
- *Helmholtz reciprocity*: $f_r(\theta_i, \phi_i, \theta_r, \phi_r) = f_r(\theta_r, \phi_r, \theta_i, \phi_i)$
- *Energy conservation*: To total quantity of the scattering cannot exceed the quantity of the incoming radiance.

Most real-world surfaces are *isotropic*, that is BRDF has no dependency on the orientation of the material. However, certain material types such as wood or brushed metal are the exceptions since the reflectance behavior depends on the orientation of the surface. We analyze such anisotropic materials in more detail in Chapter 3 of the thesis and continue our analysis assuming that the isotropy constraint is satisfied. Let $\mathbf{x} \in \mathbb{R}^3$ be a surface point and $\mathbf{n} \in \mathbb{R}^3$ be its normal vector. We express the BRDF simply as:

$$f_r(\mathbf{n}, \omega_i, \omega_j) \quad (1.1)$$

Accordingly, the total radiance $L_r(\omega_r)$ leaving a surface point in the direction of ω_r can be described by the following *rendering equation*:

$$L_r(\omega_r) = L_e(\omega_r) + \int_{\Omega} f_r(\mathbf{n}, \omega_i, \omega_r) L_i(\omega_i) (\mathbf{n} \cdot \omega_i) d\omega_i \quad (1.2)$$

Here, $L_e(\omega_r)$ is the emitted radiance by the object to the direction ω_r , and $L_i(\omega_i)$ is the incoming radiance due to the light source at direction ω_i . The term $(\mathbf{n} \cdot \omega_i)$ stands for the foreshortening factor, which attenuates the light by the dot product of normal vector \mathbf{n} and reflection direction ω_r . The integral term is computed over a hemisphere Ω centered at the normal vector \mathbf{n} such that the foreshortening factor $(\mathbf{n} \cdot \omega_i)$ is always non-negative. If there exists only a set of discrete light sources, and the emitted radiance by the object is zero, i. e. $L_e(\omega_r) = 0$ for all ω_r , the rendering equation in Eq:(1.2) can be written as a summation:

$$L_r(\omega_r) = \sum_i f_r(\mathbf{n}, \omega_i, \omega_r) L_i(\omega_i) \max(\mathbf{n} \cdot \omega_i, 0) \quad (1.3)$$

Here, the $\max(\mathbf{n} \cdot \omega_i, 0)$ term ensures that the surface point is not shadowed. Note that the image formation model in Eq:(1.3) does not take global illumination effects into account, and attempts to describe the reflectance by ignoring the interactions among multiple surface elements. Furthermore, it is important to note that BRDF is a complex function and expressing it with an analytical representation is difficult, if not impossible. Therefore, typical BRDFs are generally split into *diffuse* and *specular* components to construct simplified image formation models.

Diffuse Reflection: It is a reflection where the light is scattered uniformly in all directions and creates the shading effect observed in matte objects. Ideally, a diffuse surface exhibits a *Lambertian reflectance* property, i. e. $f_r(\mathbf{n}, \omega_i, \omega_r) = \rho / \pi$.

Specular Reflection: It is the mirror-like reflection where the incident light is mostly reflected in a direction $\mathbf{s}_i = (2\mathbf{n}\mathbf{n}^T - I)\omega_i$. For this reason, it is also referred as gloss or highlight reflection.

1.1.2 *Photometric Stereo*1.1.2.1 *Classical Photometric Stereo*

The classical setup for photometric stereo is introduced by Woodham in 1980 to recover the surface orientation of an object from its images under a known combination of light sources [11]. For that, it considers an orthographic camera model with a fixed viewpoint and a set of directional light sources. The images are captured by firing one unique directional light source per image. To solve for per-pixel surface normals of an object using such an acquisition setup, the classical PS assumes a Lambertian surface model resulting in a constant BRDF value across the whole surface. For this reason, it is also referred to as *Lambertian photometric stereo*. Additionally, the surface is considered to be illuminated only due to the light source, that is ambient illumination doesn't exist and secondary illumination effects are ignored. Under such assumptions, photometric stereo becomes a linearly tractable problem and it is possible to recover the surface normals by solving a simple system of linear equations.

Let all the n light source directions be denoted as $\mathbf{L} = [\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_n] \in \mathbb{R}^{3 \times n}$ and m unknown surface point normal be $\mathbf{N} = [\mathbf{n}(\mathbf{x}_1), \mathbf{n}(\mathbf{x}_2), \dots, \mathbf{n}(\mathbf{x}_m)] \in \mathbb{R}^{3 \times m}$. Using this notation, we can describe the image formation due to all the light sources and surface points compactly as follows:

$$\mathbf{X} = \rho \mathbf{N}^T \mathbf{L} \quad (1.4)$$

where, $\mathbf{X} \in \mathbb{R}^{m \times n}$ is the matrix consisting of n images with m object pixels stacked as column vectors, and ρ is the constant albedo. The above system can be solved for the surface normals using the matrix pseudo-inverse approach under calibrated setting if $n \geq 3$ (i. e., at least three light sources are given in non-degenerate configuration). Although the model in Eq.(1.4) builds a foundation for photometric stereo techniques and widely used for its simplicity, it depends on unrealistic assumptions. In a real setup, photometric measurements do not obey a such a simple linear model. Most surfaces exhibit specular reflections which severely affect the imaging. Furthermore, pixel intensity values are affected by shadows, non-linear camera sensor responses, interreflections and imaging noise. Therefore, the classical photometric stereo generally results in inaccurate surface normal predictions in practice.

1.1.2.2 Non-Lambertian Photometric Stereo

As many objects in the real-world exhibit non-Lambertian surface reflectance properties and exposed to the imaging effects mentioned in the previous section, we start by introducing a more extensive image formation model for general photometric stereo. When a surface point \mathbf{x} is illuminated by a distant point light source from direction $\mathbf{l}_s \in \mathbb{R}^{3 \times 1}$, the image intensity $X_s(\mathbf{x})$ measured by the camera due to s^{th} source in the view direction \mathbf{v} is given by:

$$X_s(\mathbf{x}) = e_s \cdot \rho(\mathbf{n}(\mathbf{x}), \mathbf{l}_s, \mathbf{v}) \cdot \zeta_a(\mathbf{n}(\mathbf{x}), \mathbf{l}_s) \cdot \zeta_c(\mathbf{x}) + \epsilon_s \quad (1.5)$$

Here, the camera projection model is assumed to be orthographic. It also assumes that a unique image per light source is captured by a camera from a constant view direction \mathbf{v} which is at $(0, 0, 1)^T$. The function $\rho(\mathbf{n}(\mathbf{x}), \mathbf{l}_s, \mathbf{v})$ gives the BRDF value, $\zeta_a(\mathbf{n}(\mathbf{x}), \mathbf{l}_s) = \max(\mathbf{n}(\mathbf{x})^T \mathbf{l}_s, 0)$ accounts for the attached shadow, and $\zeta_c(\mathbf{x}) \in \{0, 1\}$ assign 0 or 1 value to \mathbf{x} depending on whether it lies in the cast shadow region or not. $e_s \in \mathbb{R}_+$ is a scalar for light intensity value, $\mathbf{n}(\mathbf{x}) \in \mathbb{R}^{3 \times 1}$ is the surface normal vector at point \mathbf{x} , and ϵ_s is an additive error. Eq:(1.5) is most-widely used image formation model and it lies at the core of many non-Lambertian photometric stereo approaches [12–17].

Robust Methods. These methods extend the classical photometric stereo assumptions to non-Lambertian setting. For that, they treat all the non-Lambertian effects such as specularities and shadows as outliers. Accordingly, the classical PS equation in Eq:(1.4) can be modified as follows:

$$\mathbf{X} = \rho \mathbf{N}^T \mathbf{L} + \mathbf{E} \quad (1.6)$$

Here, $\mathbf{E} \in \mathbb{R}^{m \times n}$ stands for the corruption matrix of n images with m object pixels. Under the assumption that the corruption matrix is sparse, i. e. most entries of \mathbf{E} is zero, statistical techniques can be applied to recover surface normals. Accordingly, Wu *et al.* [18] uses a low-rank matrix factorization technique to eliminate outliers. Similarly, Ikehata *et al.* [19] uses Bayesian regression and Oh *et al.* [20] uses Robust Principal Component Analysis (RPCA) based method for the same task. Other popular outlier rejection methods were based on RANSAC [21], and expectation-maximization [22]. The common drawback of these approaches is that they cannot handle non-Lambertian objects if the outliers are dense and they generally require more images to perform statistical analysis. We provide more details on the RPCA-based robust photometric stereo in Chapter 2 of this thesis.

Analytical BRDF Models. Instead of eliminating the specular components, these methods attempt to model image formation using an analytical reflectance function. For that, the object’s reflectance is generally modeled as a mixture of diffuse and specular components [23–25]. Existing methods generally use Torrance-Sparrow [23] and Ward [24, 25] analytical models. Unlike outlier-rejection approaches, these methods have the advantage of exploiting information from pixels which exhibit specular reflection. The common drawback is that they are applicable to only limited material types, as the real reflectance behavior may divert significantly from the proposed analytical model.

1.1.2.3 *Uncalibrated Photometric Stereo*

Most existing photometric stereo methods assume a *calibrated* setting where all light source directions and intensities are given. However, the exact calibration of light sources is a difficult task and requires a tedious procedure. Moreover, it limits the application of photometric stereo in the wild. To overcome such limitations, *uncalibrated* photometric stereo techniques estimate surface normals and light source directions simultaneously. When all non-Lambertian effects are ignored as in Eq:(1.4), the surface normals of an object can be recovered up to a 3×3 linear ambiguity under the uncalibrated setting. When surface integrability constraint is also enforced, we can reduce the ambiguity to a 3-parameter *Generalized Bas-Relief* (GBR) transformation, such that $\mathbf{X} = (\mathbf{G}^{-T} \tilde{\mathbf{N}})^T (\mathbf{GL})$. Here, $\tilde{\mathbf{N}} \in \mathbb{R}^{3 \times m}$ denotes the albedo scaled normals and $\mathbf{G} \in \mathbb{R}^{3 \times 3}$ is the transformation matrix with 3 unknown parameters [12, 26].

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \mu & \nu & \lambda \end{bmatrix} \quad (1.7)$$

The GBR ambiguity indicates that there are infinitely many solutions leading up to the same appearance of the Lambertian surface. Existing methods eliminate this ambiguity by making some additional assumptions in their proposed solution. Alldrin *et al.* [27] assumes bounded values on the GBR variables and resolves the ambiguity by minimizing the entropy of albedo distribution. Shi *et al.* [28] assumes at least four pixels with different normals but the same albedo. Papadhimitri *et al.* [29] presents a closed-form solution by detecting local diffuse reflectance maxima (LDR). Other meth-

ods assume perspective projection [30], specularities [23, 31], low-rank [32], interreflections [12] or symmetry properties of BRDFs [33–35].

The methods mentioned above assume a directional light source per image to solve for the surface normals. However, there exists several works which consider the problem under the natural illumination setting. The seminal work of Basri *et al.* [36] uses spherical harmonics representation to solve photometric stereo for arbitrary lighting conditions. It also inspired several follow-ups [37–41] which attempt to handle natural lighting. Nevertheless, this thesis focuses on the directional lighting setting as in Woodham’s classical setup.

1.1.2.4 Learning-Based Methods

With the recent success of deep learning in many computer vision areas, several learning-based approaches have also emerged for the photometric stereo problem. Unlike early traditional approaches, learning based methods generally don’t attempt to explicitly model the reflectance. Instead, they learn a mapping between image intensity profiles and surface normals by utilizing the powerful learning capability of deep neural networks. This brings the advantage of handling not only complex reflectance behavior of the subjects, but also the global illumination effects. We review learning-based approaches in two categories depending on the availability of light source information:

Calibrated Setting. Santo *et al.* [42] introduced the first deep photometric stereo network (DPSN) that learns the mapping between the surface normals and the reflectance map. However, it required a pre-defined lighting configuration for both training and testing, which limits its application to different acquisition setups. To overcome this limitation, Chen *et al.* [43] proposed a flexible deep learning pipeline that can aggregate information coming from arbitrary configuration of photometric stereo images. Similarly, Ikehata [14] merged all pixel-wise information to an observation map and trained a network to perform per-pixel estimation of normals.

Apart from the fully supervised approaches, Tani ai *et al.* [15] used an inverse rendering framework to recover surface normals from input images. Similarly, Li *et al.* [44] exploited information of shadowed regions in a self-supervised pipeline.

Recent approaches to the problem benefit from attention mechanisms [45–47], graph convolutional networks [48] and data augmentation strategies [49]. Learning-based methods also consider non-conventional settings such as near-field [50, 51], and sparse photometric stereo [52, 53].

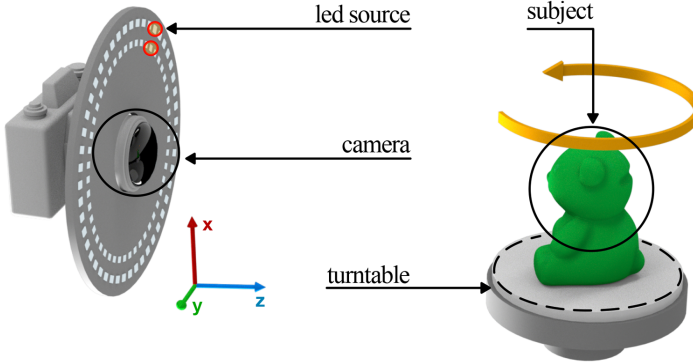


FIGURE 1.1: The classical MVPS setup as outlined in Hernández *et al.* [56] work.

Uncalibrated Setting. Chen *et al.* [43] proposed a learning-based framework (UPS-FCN) for the uncalibrated setting. This method bypasses the light estimation process and learns a direct mapping between the image and the surface normal. But, the knowledge of the light source would provide useful evidence about the surface normals, and therefore completely ignoring the light source data seems implausible. The self-calibrating deep photometric stereo network [54] introduced an initial lighting estimation stage (LCNet) to overcome the problem with UPS-FCN. Later, Chen *et al.* [16] extended the LCNet pipeline by feeding the surface normal predictions to a guided calibration network (GCNet). Recently, Ikehata [55] proposed an approach that does not assume a specific lighting model. The proposed universal photometric stereo network (UniPS) predicts surface normals from images captured under arbitrary illumination conditions.

1.1.2.5 Multi-View Photometric Stereo

Multi-view photometric stereo (MVPS) aims at recovering accurate and complete 3D reconstruction of an object using multi-view stereo (MVS) and photometric stereo (PS) images [56]. While PS is exemplary in recovering an object’s high-frequency surface details, MVS helps in retaining the global consistency of the object’s 3D shape and assists in correcting overall low-frequency distortion due to PS [1, 57, 58]. Hence, MVPS inherits the complementary output response of PS and MVS methods. Contrary to the active range scanning methods [57, 59, 60], it provides an efficient, low-cost, and effective alternative for trustworthy 3D data acquisition. And

therefore, it is widely preferred in architectural restoration [57], machine vision industry [56, 58, 61], etc.

Hernández *et al.* [56] proposed the introductory MVPS acquisition setup. It is composed of a turntable arrangement, where light-varying images (PS images) of the object placed on the table are captured from a given viewpoint. Note that the camera and light sources' position remains fixed, and only the table rotates, providing a new view of the object per rotation. For every table rotation, PS images for each light source are captured and stored (see Fig.1.1). More details on the existing MVPS methods are presented in Chapter 3 of the thesis.

1.1.3 Limitations

The classical photometric stereo approach assumes a simple image formation model with Lambertian surface reflectance. Despite the popularity and simplicity of the method, its assumptions don't apply for the real-world objects. Most objects don't reflect the light diffusely due to the presence of specularities, shadows or interreflections. Moreover, the response of the camera sensor and imaging noise have great impact on the results. Therefore, traditional photometric stereo has progressed towards outlier-rejection based methods and analytical BRDF models, considering non-Lambertian reflectance functions. However, image formation is a complex natural process, and cannot be expressed with explicit hand-crafted reflectance models.

Despite the success achieved by learning-based approaches, current state-of-the-art photometric stereo has several practical limitations. First of all, the assumptions on the surface reflectance properties greatly limits the application of photometric techniques to real-world objects. Existing methods generally fail on surfaces with non-typical reflectance characteristics such as anisotropic and glossy materials. Second, getting accurate surface normals requires processing of per-pixel and global information provided by large number of images. Hand-crafted architectures for this task are computationally inefficient. Furthermore, deep photometric stereo methods are limited by the availability of diverse real-world datasets for training. Therefore, methods which do not require training with ground-truth surface normals are becoming more important. Moreover, most photometric stereo methods require exact calibration of the directional light sources. As the calibration of light sources is a difficult task, the application to uncontrolled settings are restricted.

In addition to the surface normal estimation, there are several limitations to the task of getting dense reconstructions using photometric techniques. Recovery of the actual depth from predicted surface normals is a challenge due to the integration, non-directional lighting effects and lack of global constraints on the 3D shape. To mitigate this issue, photometric stereo is often combined with multi-view stereo (MVS) in Multi-view photometric stereo (MVPS) setting. However, existing MVPS methods rely on sparse keypoints acquired from multi-view images and explicit image formation models to recover the object shape [62, 63]. Furthermore, they are comprised of intricate manual steps, restricting its application in real-world scenarios.

1.2 THESIS OUTLINE

After a brief introduction, we provide an overview of the thesis. In Chapter 2, we introduce an uncalibrated neural inverse rendering pipeline for the uncalibrated photometric stereo problem. The proposed learning-based approach initially estimates the light directions from the input images. Then it optimizes the network parameters using an image reconstruction loss to calculate the surface normals, bidirectional reflectance distribution function value, and depth. The rendering equation used in image reconstruction loss explicitly models the concave and convex parts of a complex surface to consider the effects of interreflections to handle a broad range of surface geometries.

Chapter 3 of the thesis is devoted to the multi-view photometric stereo problem. We start the chapter by reviewing the multi-view stereo and recent neural view-synthesis approaches. Then, we present three different approaches to the problem. First, we present a neural radiance fields approach in Section §3.4. Our method procures the surface orientation using a deep photometric stereo model and blends it with a multi-view neural radiance field representation to recover the object’s surface geometry. Second, in Section §3.5, we put forward an approach that can effectively utilize complementary strengths of PS and MVS. Our key idea is to combine them suitably while considering the per-pixel uncertainty of their estimates. To this end, we estimate per-pixel surface normals and depth using an uncertainty-aware deep-PS network and deep- MVS network, respectively. We present an extensive analysis to show that uncertainty modeling helps select reliable surface normal and depth estimates at each pixel which then act as a true representative of the dense surface geometry. Finally, we present an extension of the uncertainty-aware approach in Section §3.6 with

the aim of generalizing the application of multi-view photometric stereo to glossy and anisotropic material objects. To this end, we introduce neural volume rendering methodology for a trustworthy fusion of MVS and PS measurements. The advantage of introducing neural volume rendering is that it helps in the reliable modeling of objects with diverse material types, where existing MVS methods, PS methods, or both may fail. Our suggested approach aims to fit the zero level set of the implicit neural function using the most certain MVS and PS network predictions coupled with weighted neural volume rendering cost to recover the object surface accurately for challenging material types.

Chapter 4 presents our work on automated machine learning approach for the uncalibrated photometric stereo problem. Unlike previous uncalibrated deep PS networks, which are handcrafted and carefully tuned, we leverage differentiable neural architecture search (NAS) strategy to find the architecture automatically. As directly applying the NAS methodology to uncalibrated PS is not straightforward, we impose certain task specific constraints explicitly. Our experimental study shows that automatically searched neural architectures performance compares favorably with the state-of-the-art uncalibrated PS methods while having a lower memory footprint.

UNCALIBRATED NEURAL INVERSE RENDERING

2.1 MOTIVATION

Since Woodham’s seminal work [11], the photometric stereo problem has become a popular choice to estimate an object’s surface normals from its light varying images. The formulation proposed in that paper assumes the Lambertian reflectance model of the object, and therefore, it does not apply to general objects with unknown reflectance property. Of course, the solution proposed in Woodham’s paper has some unrealistic assumptions. Still, it is central to the development of several robust algorithms [13, 18, 25, 64–66] and also lies at the core of the current state-of-the-art deep photometric stereo methods [14, 15, 17, 43, 51, 54, 67].

Generally, deep learning-based photometric stereo methods assume a calibrated setting, where all the light source information is given both at the train and test time [14, 15, 42, 43]. Such methods attempt to learn an explicit relation between the reflectance map and the ground-truth surface normals. But, the exact estimation of light directions is a tedious process and requires expert skill for calibration. Motivated by that, Chen *et al.* [17, 54] proposed an uncalibrated photometric stereo method. Though it estimates light directions using image data, the proposed method requires ground-truth surface normals for training the neural network. Certainly, procuring ground-truth 3D surface geometry is difficult, if not impossible, which makes the acquisition task of correct surface normals strenuous. For 3D data acquisition, active sensors are mostly used, which are expensive and often needs post-processing of the data to remove noise and outliers. Hence, the necessity of ground-truth surface normals limits the usage of such an approach.

Further, most photometric stereo methods, including current deep-learning methods, assume that each surface point is illuminated only by the light source, which generally holds for a convex surface [68]. However, objects, mainly from ancient architectures, have complex geometric structures, where the shape may compose of convex, concave, and other fine geometric primitives (see Fig.2.1). When illuminated under a varying light source, certain concave parts of the surface might reflect light onto other parts of the object, depending on its position. Surprisingly, this phenomenon

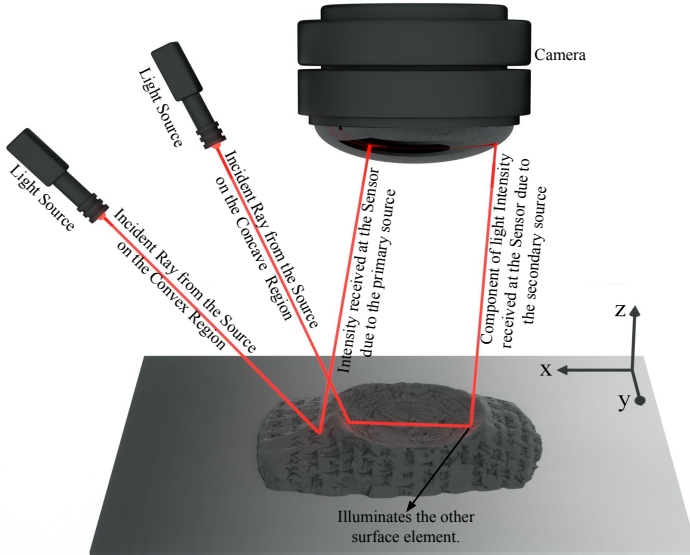


FIGURE 2.1: Example showing the interreflection effect due to concave geometric structure. The light from the primary source hits the concave region of the surface that illuminates the other surface points which then act as a secondary light source.

of *interreflections* is often ignored in the modeling and formulation of a photometric stereo problem, despite its vital role in the object’s imaging.

2.2 CONTRIBUTIONS

In this chapter, we introduce an uncalibrated deep photometric stereo approach to overcome the limitations mentioned above. The proposed method does not require ground-truth surface normals at train time and leverages neural inverse rendering principles to infer the surface normals, depth, and spatially varying bidirectional reflectance distribution function (BRDF) values only from input images. Our approach considers the contribution of both the source light and interreflections in the image formation process. Consequently, it is more general and applicable to a wide range of objects. We present an extensive analysis to show that ignoring interreflections can dramatically affect the accuracy of the surface normals estimate (see Fig 2.2).

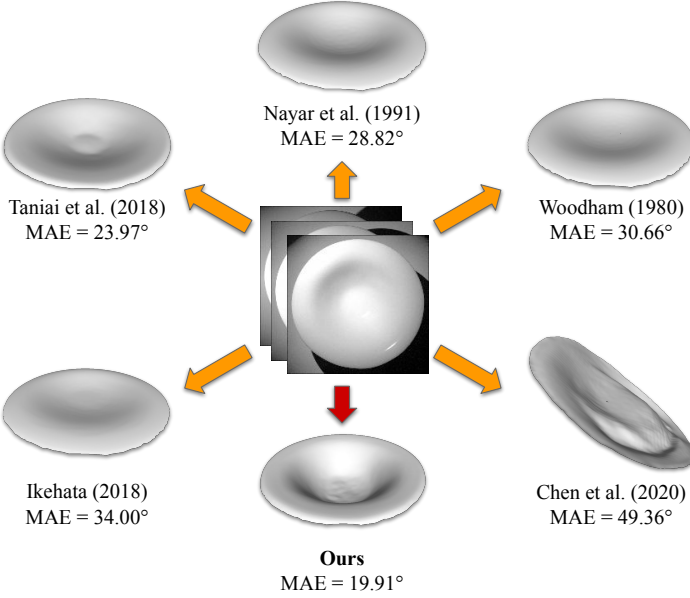


FIGURE 2.2: Comparison of our approach against the classical and deep-learning methods on the Vase dataset which shows that it performs better than others. We used Mean Angular Error (MAE) metric to report the results.

2.3 INTERREFLECTION MODEL

Before we describe the details of the proposed method, we provide an image formation model which considers the interreflections on the object. To construct such an interreflection model, we start by Woodham’s classical photometric stereo formulation. Let $\mathbf{L} = [\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_n] \in \mathbb{R}^{3 \times n}$ denote all the n light source directions and $\mathbf{N} = [\mathbf{n}(\mathbf{x}_1), \mathbf{n}(\mathbf{x}_2), \dots, \mathbf{n}(\mathbf{x}_m)] \in \mathbb{R}^{3 \times m}$ denote m unknown surface point normal vectors and ρ is the constant albedo. Then, the matrix consisting of all image intensity values due to the primary light source $\mathbf{X}_s \in \mathbb{R}^{m \times n}$ can be written as follows:

$$\mathbf{X}_s = \rho \mathbf{N}^T \mathbf{L} \quad (2.1)$$

Of course, this classical equation assumes a convex Lambertian surface model resulting in a constant BRDF value across the whole surface. Additionally, the surface is considered to be illuminated only due to the light

source. In contrast to the classical photometric stereo, here, the total radiance at a point \mathbf{x} on the surface is the sum of radiance due to light source s and the radiance due to interreflection from other surface points.

$$X(\mathbf{x}) = \underbrace{X_s(\mathbf{x})}_{\text{due to light source}} + \overbrace{\frac{\rho(\mathbf{x})}{\pi} \int_S K(\mathbf{x}, \mathbf{x}') X(\mathbf{x}') d\mathbf{x}'}_{\text{due to interreflections}} \quad (2.2)$$

where, S represents the surface, \mathbf{x}' is another surface point, and $d\mathbf{x}'$ the differential surface element at \mathbf{x}' . The value of the interreflection kernel ' K ' at \mathbf{x} due to \mathbf{x}' is defined as:

$$K(\mathbf{x}, \mathbf{x}') = \left(\frac{(\mathbf{n}(\mathbf{x})^T(-\mathbf{r})) \cdot (\mathbf{n}(\mathbf{x}')^T \mathbf{r}) \cdot V(\mathbf{x}, \mathbf{x}')}{(\mathbf{r}^T \mathbf{r})^2} \right) \quad (2.3)$$

The values of K , when measured for each surface element form a symmetric and positive semi-definite matrix. In Eq:(2.3), $V(\mathbf{x}, \mathbf{x}')$ captures the visibility. When \mathbf{x} occludes \mathbf{x}' or vice-versa, then V is 0. Otherwise, V is 1. It is computed using the following expression:

$$V(\mathbf{x}, \mathbf{x}') = \left(\frac{\mathbf{n}(\mathbf{x})^T(-\mathbf{r}) + |\mathbf{n}(\mathbf{x})^T(-\mathbf{r})|}{2|\mathbf{n}(\mathbf{x})^T(-\mathbf{r})|} \right) \cdot \left(\frac{\mathbf{n}(\mathbf{x}')^T \mathbf{r} + |\mathbf{n}(\mathbf{x}')^T \mathbf{r}|}{2|\mathbf{n}(\mathbf{x}')^T \mathbf{r}|} \right) \quad (2.4)$$

where, $\mathbf{n}(\mathbf{x})$ and $\mathbf{n}(\mathbf{x}')$ are the surface normal at \mathbf{x} and \mathbf{x}' , and $\mathbf{r} = \mathbf{x} - \mathbf{x}'$ is the vector from \mathbf{x}' to \mathbf{x} . Substituting V and K in Eq:(2.2) gives an infinite sum over every infinitesimally small surface element (point) and therefore, it is not computationally easy to find a solution to $X(\mathbf{x})$ in its continuous form. Nevertheless, the solution to Eq:(2.2) is guaranteed to converge as $\rho(\mathbf{x}) < 1$ for a real surface. To practically implement the interreflection model, the object surface is discretized into m facets [68]. Assuming the radiance and albedo values to be constant within each facet, then Eq:(2.2) for the i^{th} facet becomes

$$X_i = X_{si} + \frac{\rho_i}{\pi} \sum_{j=1, j \neq i}^m X_j K_{ij} \quad (2.5)$$

where $X_i \in \mathbb{R}^{n \times 1}$ and ρ_i are the radiance and albedo of facet i . Considering the contribution of all the light sources for each facet, it can be compactly re-written as:

$$\mathbf{X} = \mathbf{X}_s + \mathbf{PKX}, \Rightarrow \mathbf{X} = (\mathbf{I} - \mathbf{PK})^{-1} \mathbf{X}_s \quad (2.6)$$

where, $\mathbf{X} = [X_1, X_2, \dots, X_m]^T$ is the total radiance for all the facets, and $\mathbf{X}_s = [X_{s1}, X_{s2}, \dots, X_{sm}]^T$ is the light source contribution to the radiance of m facets. Furthermore, \mathbf{P} is a diagonal matrix composed of albedo values and \mathbf{K} is a $m \times m$ interreflection kernel matrix with $\text{diag}(\mathbf{K}) = 0$. Nayar *et al.* [68] proposed Eq:(2.6) to recover the surface normals for concave objects. The algorithm proposed to estimate surface normals using Eq:(2.6) first computes the pseudo surface normals by treating the object as directly illuminated by light sources. These pseudo surface normals are then used to iteratively update for the interreflection kernel and surface normals via depth map estimation step, until convergence. In the later part of the chapter, we denote the normals estimated using Eq:(2.6) as \mathbf{N}_{ny} . The Nayar’s interreflection model assumes Lambertian surfaces and overlooks surfaces with unknown non-Lambertian properties.

2.4 METHOD

In this section, we introduce our uncalibrated neural inverse rendering network. Our approach first estimates all the light source directions and intensities using image data. Computed light source information is then fed into the proposed neural inverse rendering network to estimate the surface normals. The idea is, those correct surface normals, when provided to the rendering equation, should reconstruct the input image as close as possible. Consequently, we can bypass the requirement of the ground-truth surface normals at train time. Unlike recent methods, our approach models the effects of both the light source and the interreflections for rendering the image. Although one can handle interreflection using classical methods [12, 68], the reflectance characteristics of different types of material are quite diverse. Hence, we want to leverage neural network’s powerful capability to learn complex reflectance behavior from the input image data.

Let $\mathbf{X} = [X_1, X_2, \dots, X_n]$ be a set of n input images captured by photometric stereo setup and \mathbf{O} be the object mask. Here, each image X_i is reshaped as a column vector and not a facet symbol as used in interreflection modeling. Even though the problem with unknown light directions gives rise to the bas-relief ambiguity [26], we leverage the potential of the deep neural networks to learn those source directions from the input image data using a light estimation network as described in Section §2.4.1. The estimated light directions are used by the inverse rendering network in Section §2.4.2 to infer the unknown BRDFs and surface normals using our proposed

rendering equation. Our rendering approach explicitly utilizes the role of the light source and interreflections in the image reconstruction process.

2.4.1 Light Estimation Network

Given \mathbf{X} and \mathbf{O} , the light estimation network predicts the light source intensities (\mathbf{e}_i 's) and direction vectors (\mathbf{l}_i 's). We can train such a network either by regressing the intensity values and the corresponding unit vector in the source's direction or classifying intensity values into pre-defined angle-range bins. The latter choice seems reasonable as it is easier than regressing the exact direction and intensity values as discussed in [54]. Further, quantizing the continuous space of directions and intensities for classification makes the network robust to small changes due to outliers or noise. Following that, we express the light source directions in the range $\phi \in [0, \pi]$ for azimuth angles and $\theta \in [-\pi/2, \pi/2]$ for elevation angles (Fig.2.4(a)). We divide the azimuth and elevation spaces into $K_d = 36$ classes. We classify azimuth and elevation separately, which reduces the problem's dimensionality and leads to efficient computation. Similarly, we divide the light intensity range $[0.2, 2.0]$ into $K_e = 20$ classes following [54].

We used seven feature extraction layers to extract image features for each input image separately, where each layer applies 3×3 convolution and LReLU activation [69]. The weights of the feature extraction layers are shared among all the input images. However, single image features cannot completely disambiguate the object geometry with the light source information. Therefore, we utilize multiple images to have a global implicit knowledge about the surface's geometry and its reflectance property. We use image specific local features and combine them using a fusion layer to get a global representation of the image set via a max-pooling operation (Fig.2.3). The global feature representation with the image-specific features is then fed to a classifier. The classifier applies four layers of 3×3 convolution and LReLU activation [69] as well as two fully-connected layers to provide output softmax probability vectors for azimuth (K_d), elevation (K_d), and intensity (K_e). Similar to the feature extraction, the classifier weights are shared among each other. The output value with maximum probability is converted into a light direction vector \mathbf{l}_i and scalar intensity \mathbf{e}_i .

Loss function for Light Estimation Network. The light estimation network is trained using a multi-class cross-entropy loss [54]. The total calibration loss $\mathcal{L}_{\text{calib}}$ is:

$$\mathcal{L}_{\text{calib}} = \mathcal{L}_{\text{az}} + \mathcal{L}_{\text{el}} + \mathcal{L}_{\text{in}} \quad (2.7)$$

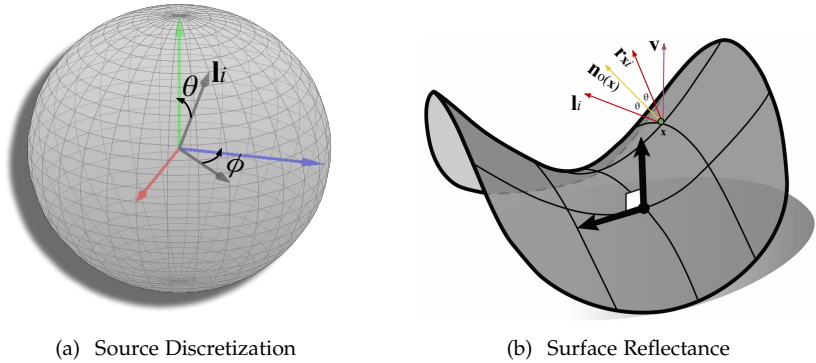


FIGURE 2.4: (a) The estimated source directions are given by two parameters: $\phi \in [0, \pi]$ and $\theta \in [-\pi/2, \pi/2]$. (b) Illustration of surface reflectance. When light ray \mathbf{l}_i hits a surface element, the specular component along the view-direction of the point x due to i^{th} source is given by \mathbf{r}_{xi} . Figure 2(b) geometry presentation is inspired by Keenan work [70].

Here \mathcal{L}_{az} , \mathcal{L}_{el} , and \mathcal{L}_{in} are the loss terms for azimuth, elevation, and intensity respectively. We used synthetic Blobby and Sculpture datasets [43] to train the network. The light source labels from these datasets are used for supervision at the train time. The network is trained using the above loss for once and the same network is used at the test time for all other datasets for our experimental study in Section §2.5.

2.4.2 Inverse Rendering Network

To estimate an object surface normals from \mathbf{X} , we leverage neural networks' powerful capability to learn from data. The prime reason for that is, it is difficult to mathematically model the broad classes of BRDFs without any prior assumptions about the reflectance model [23–25]. Although there are methods to estimate BRDF values using its isotropic and low-frequency property [71, 72], it prohibits the modeling of unrestricted reflectance behavior of the material. Instead of such explicit modeling, we build on the idea of neural inverse rendering [15], where the BRDFs and surface normals are predicted during the image reconstruction process by the neural network. We go beyond Taniai *et al.* [15] work by proposing an inverse rendering

network that synthesizes the input images using a rendering equation that explicitly uses interreflections to infer surface normals.

(a) Surface Normal Modeling. We first convert \mathbf{X} into a tensor $\mathcal{X} \in \mathbb{R}^{h \times w \times nc}$, where $h \times w$ denote the spatial dimensions, n is the number of images, and c is the number of color channels ($c = 1$ for grayscale and $c = 3$ for color images). \mathcal{X} is then mapped to a global feature map Φ as follows:

$$\Phi = \zeta_f(\mathcal{X}, \mathbf{O}, \Theta_f) \quad (2.8)$$

\mathbf{O} is used to separate the object information from the background. ζ_f is a three layer feed-forward convolutional network with learnable parameter Θ_f . Each layer applies 3×3 convolution, batch-normalization [73] and ReLU activation [69] to extract global feature map Φ . In the next step, we use Φ to compute the surface normals. Let ζ_{n1} be the function that converts Φ into output normal map \mathbf{N}_o via 3×3 convolution and L2-normalization operation.

$$\mathbf{N}_o = \zeta_{n1}(\Phi, \Theta_{n1}) \quad (2.9)$$

Here, Θ_{n1} is the learnable parameter. We used the estimated \mathbf{N}_o to compute \mathbf{N}_{ny} using function ζ_{n2} .

$$\mathbf{N}_{ny} = \zeta_{n2}(\mathbf{N}_o, \mathbf{P}, \mathbf{K}) \quad (2.10)$$

ζ_{n2} requires the interreflection kernel \mathbf{K} and albedo matrix \mathbf{P} as input. To calculate \mathbf{K} , we integrate the \mathbf{N}_o over masked object pixel coordinates (x, y) to obtain the depth map [4, 74]. Afterward, the depth map is used to infer the kernel matrix \mathbf{K} (see Eq:(2.3)). Once we have \mathbf{K} , we employ Eq:(2.6) to compute \mathbf{N}_{ny} . Later, \mathbf{N}_{ny} is used in the rendering equation (Eq:(2.16)) for image reconstruction.

(b) Reflectance Modeling. For effective learning of BRDFs, it is important to model the specular component. To incorporate that, we feed a specular map along with the input image as a channel. To compute it, we first compute \mathbf{r}_{xi} for each point \mathbf{x} that is the direction vector with the highest specular component using the following well-known relation; assuming \mathbf{l}_i , and \mathbf{n}_o as unit length vectors:

$$\begin{aligned} \mathbf{r}_{xi} + \mathbf{l}_i &= 2\cos(\theta) \cdot \mathbf{n}_o(\mathbf{x}); \quad \mathbf{n}_o(\mathbf{x})^T \mathbf{l}_i = \cos(\theta) \\ \mathbf{r}_{xi} &= 2(\mathbf{n}_o(\mathbf{x})^T \mathbf{l}_i) \cdot \mathbf{n}_o(\mathbf{x}) - \mathbf{l}_i \end{aligned} \quad (2.11)$$

Here, \mathbf{r}_{xi} is also a unit length vector (see Fig.2.4(b)). The component of specular reflection in the view-direction $\mathbf{v} = (0, 0, 1)^T$ of the point \mathbf{x} due to i^{th} light is computed as:

$$R_i(\mathbf{x}) = \mathbf{v}^T \left(2(\mathbf{n}_o(\mathbf{x})^T \mathbf{l}_i) \cdot \mathbf{n}_o(\mathbf{x}) - \mathbf{l}_i \right) \quad (2.12)$$

The above relation shows that the specular highlights are strongest if \mathbf{r}_{xi} is close to \mathbf{v} . Computing $R_i(\mathbf{x})$ for all surface points provides the specular-reflection map $R_i \in \mathbb{R}^{h \times w \times 1}$. Concatenating $X_i \in \mathbb{R}^{h \times w \times c}$ with R_i across channel guides the network to learn complex BRDFs. Thus, we compute feature map S_i as:

$$S_i = f_{sp}(X_i \oplus R_i, \Theta_{sp}) \quad (2.13)$$

We used \oplus to denote the concatenation operation. f_{sp} is a three-layer network where each layer applies 3×3 convolution, batch-normalization [73] and ReLU operations [69]. Although the feature map S_i models the actual specular component of a BRDF, it is computed using a single image observation X_i which has limited information. To enrich the feature, we concatenate it with the global features Φ (see Eq:(2.8)) and compute enhanced feature block Z_i .

$$Z_i = f_{lg}(S_i \oplus \Phi, \Theta_{lg}) \quad (2.14)$$

f_{lg} function applies 1×1 convolution, batch normalization [73] and ReLU operations [69] to estimate Z_i . Finally, we define the reflectance function f_r that blends the image specific features with Φ along with the specular component of the image to compute the reflectance map Ψ_i .

$$\Psi_i = f_r(Z_i, \Theta_{ri}) \quad (2.15)$$

The function f_r applies 3×3 convolution, batch normalization [73], ReLU operation [69] with an additional 3×3 convolution layer to compute Ψ_i . The predicted Ψ_i by the network contains the BRDFs and cast shadow information. The specular (Θ_{sp}), local-global (Θ_{lg}), and reflectance image (Θ_{ri}) parameters are learned over SGD iteration by the network. Details about the implementation of above functions, learning and testing strategy are described in §2.5.

(c) Rendering equation. Assuming photometric stereo setup, once we have the surface normals, reflectance map, and light source information, we render the input image using the following equation:

$$\check{X}_i = \Psi_i \odot (e_i \cdot \zeta_a(\mathbf{N}_{ny}, \mathbf{l}_i)) \quad (2.16)$$

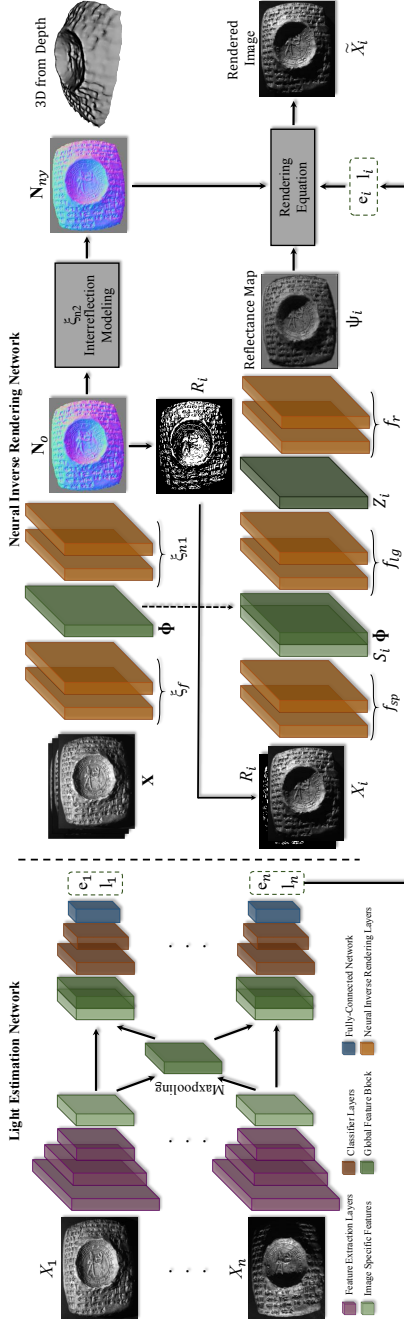


FIGURE 2.5: The proposed method consists of two networks. Light estimation network initially predicts the light source directions and intensities from input images. Then, neural inverse rendering network uses the images and the light source estimations to recover surface normals, depth and BRDF values.

Here, we explicitly model the effects of interreflections in the image formation. For a given source, Ψ_i encapsulates the BRDF values with the cast shadow information. Further, ζ_a is defined for the attached shadow. With a slight abuse of notation used in Eq:(1.5), ζ_a computes the inner product between a light source and the surface normal matrix for each pixel, and the maximum operation is done element-wise i. e., $\max(\mathbf{N}_{ny}^T \mathbf{l}_i, 0)$. $e_i \in \mathbb{R}_+$ is a scalar intensity value of the light source, and \odot denotes the Hadamard product. Fig.2.5 shows the entire rendering network pipeline.

Loss Function for Inverse Rendering Network. To train the proposed inverse rendering network, we use l_1 loss between the rendered images $\tilde{\mathbf{X}}$ and input images \mathbf{X} on the masked pixels (\mathbf{O}). The network parameters are learned by minimizing the following loss using the SGD algorithm:

$$\mathcal{L}_{rec}(\mathbf{X}, \tilde{\mathbf{X}}) = \frac{1}{mnc} \sum_{i,c,\mathbf{x}} |X_{i,c}(\mathbf{x}) - \tilde{X}_{i,c}(\mathbf{x})| \quad (2.17)$$

Here, m is the number of pixels within \mathbf{O} and n, c are the number of input images and color channels, respectively. The optimization of the above image reconstruction loss function seems reasonable; but, it may provide unstable behavior leading to inferior results. Therefore, we apply weak supervision to the network at the early stages of the optimization by adding a surface normal regularizer in the loss function using an initial normal estimate \mathbf{N}_{init} . Such a strategy guides the network for stable convergence behavior and a better solution to the surface normals. Accordingly, the total loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{rec}(\mathbf{X}, \tilde{\mathbf{X}}) + \lambda_w \mathcal{L}_{weak}(\mathbf{N}_{ny}, \mathbf{N}_{init}) \quad (2.18)$$

where, function \mathcal{L}_{weak} is defined as:

$$\mathcal{L}_{weak}(\mathbf{N}_{ny}, \mathbf{N}_{init}) = \frac{1}{m} \sum_{\mathbf{x}} \|\mathbf{n}_{ny}(\mathbf{x}) - \mathbf{n}_{init}(\mathbf{x})\|_2^2 \quad (2.19)$$

Least-square solution of \mathbf{N} in Eq:(2.1) can provide weak supervision to the network in the early stage of the optimization. However, such initialization may provide undesirable behavior at times. Therefore, we adhere to the robust optimization algorithm on photometric stereo to initialize the surface normal in Eq:(2.18).

2.4.3 Robust Initialization

Our method uses an initial surface normals prior \mathbf{N}_{init} (Eq:(2.18)) to warm up the rendering network and to initialize the interreflection kernel \mathbf{K} values. We propose to use partial sum of singular values optimization [20]. Let $\mathbf{X} \in \mathbb{R}^{m \times n}$, $\mathbf{L} \in \mathbb{R}^{3 \times n}$, $\mathbf{N} \in \mathbb{R}^{3 \times m}$, then photometric stereo equation under Lambertian assumption with $\rho = 1$ can be written as $\mathbf{X} = \mathbf{N}^T \mathbf{L} + \mathbf{E}$. Here, $\mathbf{E} \in \mathbb{R}^{m \times n}$ is a matrix of outliers and assumed to be sparse [18]. Substituting $\mathbf{Z} = \mathbf{N}^T \mathbf{L}$, the normal estimation under low rank assumption can be formulated as a *Robust Principal Component Analysis* (RPCA) problem [18]. We know that RPCA performs the nuclear norm minimization of \mathbf{Z} matrix which not only minimizes the rank but also the variance of \mathbf{Z} within the target rank. Now, for the photometric stereo model, it is easy to infer that \mathbf{N} lies in a rank-3 space. As the true rank for \mathbf{Z} is known from its construction, we do not minimize the subspace variance within the target rank (K). We preserve the variance of information within the target rank while minimizing other singular values outside it via the following optimization:

$$\min_{\mathbf{Z}, \mathbf{E}} \|\mathbf{Z}\|_{r=K} + \lambda \|\mathbf{E}\|_1, \quad \text{subject to: } \mathbf{X} = \mathbf{Z} + \mathbf{E} \quad (2.20)$$

The Augmented Lagrangian function of Eq:(2.20) can be written as follows:

$$\mathcal{L}(\mathbf{Z}, \mathbf{E}, \mathbf{Y}) = \|\mathbf{Z}\|_{r=K} + \lambda \|\mathbf{E}\|_1 + \frac{\mu}{2} \|\mathbf{X} - \mathbf{Z} - \mathbf{E}\|_F^2 + \langle \mathbf{Y}, \mathbf{X} - \mathbf{Z} - \mathbf{E} \rangle \quad (2.21)$$

Here, μ is a positive scalar and $\mathbf{Y} \in \mathbb{R}^{m \times n}$ is the estimate of the Lagrange multiplier. As minimizing this function is challenging, we solve it by utilizing the alternating direction method of multipliers (ADMM) [20, 75, 76]. Accordingly, the optimization problem in Eq:(2.21) can be divided into sub-problems, where \mathbf{Z} , \mathbf{E} and \mathbf{Y} are updated alternatively while keeping the other variables fixed.

1. Solution to \mathbf{Z} :

$$\mathbf{Z}^* = \underset{\mathbf{Z}}{\operatorname{argmin}} \|\mathbf{Z}\|_{r=K} + \frac{\mu_k}{2} \|\mathbf{Z} - (\mathbf{X} - \mathbf{E}_k + \mu_k^{-1} \mathbf{Y}_k)\|_F^2 \quad (2.22)$$

The solution to Eq:(2.22) sub-problem at k^{th} iteration is given by $\mathbf{Z}_k = \mathcal{P}_{K, \mu_k^{-1}}[\mathbf{X} - \mathbf{E}_k + \mu_k^{-1} \mathbf{Y}_k]$ where, $\mathcal{P}_{K, \tau}[\mathbf{M}] = \mathbf{U}_M(\Sigma_{M_1} + \mathcal{S}_\tau[\Sigma_{M_2}])\mathbf{V}_M^T$ is the partial singular value thresholding operator [20] and $\mathcal{S}_\tau[x] = \operatorname{sign}(x) \max(|x| - \tau, 0)$ is the soft-thresholding operator [77]. Here, $\mathbf{U}_M, \mathbf{V}_M$ are the singular

vector of matrix \mathbf{M} and $\Sigma_{\mathbf{M}_1} = \mathbf{diag}(\sigma_1, \sigma_2, \dots, \sigma_K, 0, 0)$, $\Sigma_{\mathbf{M}_2} = \mathbf{diag}(0, 0, \dots, \sigma_{K+1}, \dots, \sigma_N)$.

2. Solution to \mathbf{E} :

$$\mathbf{E}^* = \underset{\mathbf{E}}{\operatorname{argmin}} \lambda \|\mathbf{E}\|_1 + \frac{\mu_k}{2} \|\mathbf{E} - (\mathbf{X} - \mathbf{Z}_{k+1} + \mu_k^{-1} \mathbf{Y}_k)\|_F^2 \quad (2.23)$$

The solution to Eq:(2.23) sub-problem at k^{th} iteration is given by $\mathbf{E}_k = \mathcal{S}_{\lambda \mu_k^{-1}}[\mathbf{X} - \mathbf{Z}_{k+1} + \mu_k^{-1} \mathbf{Y}_k]$ where, $\mathcal{S}_\tau[x] = \operatorname{sign}(x) \max(|x| - \tau, 0)$ is a soft-thresholding operator [77].

3. Solution to \mathbf{Y} : The variable \mathbf{Y} is updated as follows over the iteration:

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k + \mu_k (\mathbf{X} - \mathbf{Z}_{k+1} - \mathbf{E}_{k+1}) \quad (2.24)$$

For proof of convergence and theoretical analysis of partial singular value thresholding operator kindly refer to Oh *et al.* [20] work. We solve for \mathbf{Z} , \mathbf{E} using ADMM until convergence for $K = 3$ and use the obtained surface normals for initializing the loss function of inverse rendering network.

2.5 EXPERIMENTS AND RESULTS

We performed evaluations of our method on DiLiGenT dataset [78]. DiLiGenT is a standard benchmark for photometric stereo, consisting of ten different real-world objects. For each object, it provides 96 images captured under different lighting conditions. Despite it provides surfaces of diverse reflectances, the subjects are not elegant for studying interreflections. Therefore, we generate a new dataset that is apt for analyzing such complex imaging phenomena. The acquisition is performed using two different setups. In the first setup, we designed a physical dome system to capture cultural artifacts. It is a 35cm hemispherical structure with 260 LEDs on the nodes for directed light projection, and with a camera on top, looking down vertically. The object under investigation lies at the center. Using it, we collected images of three historical artifacts (*Tablet1*, *Tablet2*, *Broken Pot*) with a spatial resolution of 180×225 . Ground-truth normals are acquired using active sensors with post-refinements. We noted that it is onerous to capture 3D surfaces with high precision. For this reason, we simulated the dome environment using Cinema 4D software with 100 light sources. Using this synthetic setup, we rendered images of three objects (*Vase*, *Golf-ball*, *Face*) with a spatial resolution of 256×256 . Our dataset introduces new

subjects with general reflectance properties to initiate a broader adaptation of the photometric stereo algorithm for extracting 3D surface information of real objects.

2.5.1 Implementation Details

This section provides a detailed description of our implementation. We start by introducing the light estimation network’s training phase. Then we focus on the testing phase, where the inverse rendering network is optimized to estimate the surface normals, depth, and BRDF values. Finally, we present details on training and testing run-times.

2.5.1.1 Training Details

As our inverse rendering network optimizes its learnable parameters at the test time, we apply a training stage only to the light estimation network. For training the network, we used Blobby and Sculpture datasets that are introduced by Chen *et al.* [43]. This dataset is created by using 3D geometries of Blobby [79], and Sculpture [80] shape datasets and combining them with different material BRDFs taken from MERL dataset [81]. In total, the complete dataset contains 85212 subjects. For each subject, there exist 64 renderings with different light source directions. The intensity of the light sources is kept constant during the whole data generation process. To simulate different intensities during training, image intensity values are randomly generated in the range of $[0.2, 2]$, and these intensity values are used to scale the image data linearly. In each training iteration, the input data is perturbed in the range of $[-0.025, 0.025]$ for augmentation.

The light estimation network is a multiple-input multiple-output (MIMO) system which requires images of the same object captured under different illumination conditions (see Fig.2.3). The core idea is that all input images have the same surface, and having more images helps the network extract better global features. During training, we use 32 images of the same object for global feature extraction. Note that all of the images are used for feature extraction at test time to achieve the best performance from the network.

2.5.1.2 Testing Details

Light Estimation Network. Given a set of test images \mathbf{X} and object mask \mathbf{O} , we first use the light estimation network to have light source directions and intensities. However, the light estimation network operates on 128×128

images because it uses fully connected layers for classification, and these layers process only fixed-length vectors. Consequently, we scale the input images into the resolution of 128×128 before feeding them to the network. We apply this pre-processing step only for the light estimation network and use the original image size for all other operations during testing. For objects like *Vase*, where the cast shadows and interreflections play a vital role in the object’s imaging, light estimation network can have questionable behavior. So, we use the light source directions and intensities estimated from a calibration sphere for testing our synthetic objects.

Inverse Rendering Network. Once we obtain the light source directions and intensities, we apply the robust initialization algorithm to get an initial surface normal matrix \mathbf{N}_{init} . It also provides an albedo map that is transformed into $\mathbf{P} \in \mathbb{R}^{m \times m}$ which is required for interreflection modeling. After the robust initialization process, we learn inverse rendering network’s parameters. First, we initialize all the network parameters ($\Theta_f, \Theta_{n1}, \Theta_{sp}, \Theta_{lg}, \Theta_{ri}$) which correspond to the weights of the convolution operations. In this step, we initialize the weights randomly by sampling from a Gaussian distribution with zero mean and 0.02 variance. To compute \mathcal{L}_{rec} of Eq.(2.18), we randomly sample 10% of the pixels in each iteration and compute it over these pixels to avoid local minimum. To provide weak-supervision, we set $\lambda_w = \mathcal{L}_{rec}(0, \mathbf{X})$ to balance the influence of \mathcal{L}_{rec} and \mathcal{L}_{weak} to network learning process. Note that λ_w is set to zero after 50 iterations to drop early stage weak-supervision. We perform 1000 iterations in total with initial learning rate of 8×10^{-4} . The learning rate is reduced by factor of 10 after 900 iterations for fine-tuning. Before feeding the images to the normal estimation network, we normalize them using a global scaling constant σ , i. e. the quadratic mean of pixel intensities $\mathbf{X}' = \mathbf{X}/(2\sigma)$. We also inject Gaussian noise with zero mean and 0.1 variance to the images before feeding them to f_{sp} for image reconstruction. We observed that this prohibits the network from generating degenerate solutions. During the learning of inverse rendering network, we repeatedly update the depth and the interreflection kernel \mathbf{K} using \mathbf{N}_o after every 100 iterations.

Surface Normal Integration. To compute the depth from normals, we use a gradient-based method with surface orientation constraint [74]. Given the surface normals, we first compute a gradient field $\hat{\mathbf{G}} \in \mathbb{R}^{h \times w \times 2}$ where h and w are the spatial dimensions. The idea is that the gradient field computed from surface normal map and the estimated depth $\mathbf{D} \in \mathbb{R}^{h \times w}$ should be consistent, i. e., $\nabla \mathbf{D} \approx \hat{\mathbf{G}}$. That corresponds to an overdetermined system

	GPU	Time
Training of Light Estimation Network	Titan X Pascal (12GB)	≈ 22 hours
Inference on DiLiGenT	GeForce GTX TITAN X (12GB)	53.41 ± 41.57 min per subject
Inference on our Dataset	GeForce GTX TITAN X (12GB)	29.08 ± 15.99 min per subject

TABLE 2.1: Measured training and testing time with respect to the utilized hardware. For our dataset, we have 100 to 260 images per subject and the DiLiGenT dataset has 96 images per subject. Note: Deep photometric stereo method processes a set of images rather than one image for estimating normals.

of linear equations and is solved by minimizing the following objective function i. e., Eq:(2.25) using the least-squares approach:

$$\min_{\mathbf{D}} \|\nabla \mathbf{D} - \hat{\mathbf{G}}\|^2 \quad (2.25)$$

Interreflection Modeling. To consider the effect of interreflection during the image reconstruction process, we define the function ξ_{n2} which uses the estimated normal $\mathbf{N}_o \in \mathbb{R}^{3 \times m}$, albedo matrix $\mathbf{P} \in \mathbb{R}^{m \times m}$ and the interreflection kernel $\mathbf{K} \in \mathbb{R}^{m \times m}$. Given all these components, Nayar *et al.* [68] relates the observed radiance (\mathbf{X}) and the radiance due to primary light source (\mathbf{X}_s) as shown in Eq:(2.6) Assuming the surface shows Lambertian reflectance property, we model the radiance in terms of facet matrices as follows:

$$\mathbf{X} = \mathbf{F}_{ny}\mathbf{L}, \quad \mathbf{X}_s = \mathbf{F}\mathbf{L}, \quad \Rightarrow \mathbf{F}_{ny} = (\mathbf{I} - \mathbf{P}\mathbf{K})^{-1}\mathbf{F} \quad (2.26)$$

Here $\mathbf{F}_{ny} \in \mathbb{R}^{m \times 3}$ and $\mathbf{F} \in \mathbb{R}^{m \times 3}$ are the facet matrices which contain surface normals \mathbf{N}_{ny} and \mathbf{N}_o scaled with local reflectance value. We use Eq:(2.26) to obtain \mathbf{F}_{ny} and normalize each row to unit vector to obtain \mathbf{N}_{ny} .

The computation of the interreflection kernel \mathbf{K} has the complexity of $\mathcal{O}(n^2)$ where n is the number of facets. Therefore, treating each pixel as a facet limits the application of our method. To approximate the effect of interreflections, we downsample the normal maps with the factor of 4 and calculated the kernel values accordingly. After the normal is updated, we scale it to the original size managing the image details appropriately.

2.5.1.3 Runtime Analysis

Our framework is implemented in Python using PyTorch version 1.1.0. Table (2.1) provides the light estimation network’s training time and the inference time of neural inverse rendering network on two datasets separately.

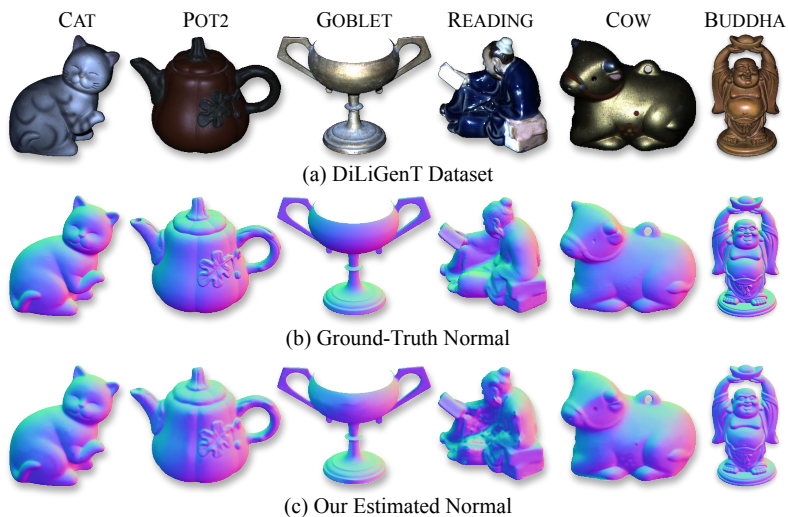


FIGURE 2.6: Qualitative results on DiLiGenT dataset using our method.

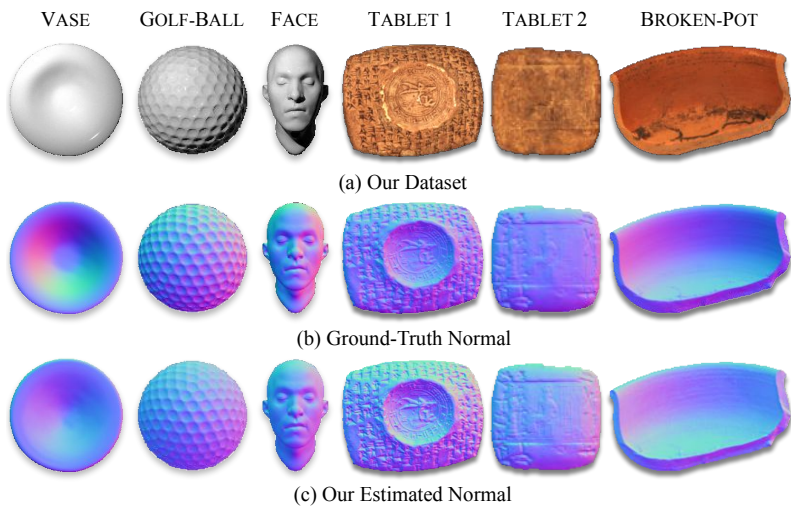


FIGURE 2.7: Qualitative results of our method on our dataset.

Type	G.T. Normal	Methods↓ Dataset →	Ball	Cat	Pot1	Bear	Pot2	Buddha	Goblet	Reading	Cow	Harvest	Average
Classical	\times	Alldrin <i>et al.</i> (2007) [27]	7.27	31.45	18.37	16.81	49.16	32.81	46.54	53.65	54.72	61.70	37.25
Classical	\times	Shi <i>et al.</i> (2018) [28]	8.90	19.84	16.68	11.98	50.68	15.54	48.79	26.93	22.73	73.86	29.59
Classical	\times	Wu <i>et al.</i> (2013) [34]	4.39	36.55	9.39	6.42	14.52	13.19	20.57	58.96	19.75	55.51	23.93
Classical	\times	Lu <i>et al.</i> (2013) [82]	22.43	25.01	32.82	15.44	20.57	25.76	29.16	48.16	22.53	34.45	27.63
Classical	\times	Pap. <i>et al.</i> (2014) [29]	4.77	9.54	9.51	9.07	15.90	14.92	29.93	24.18	19.53	29.21	16.66
Classical	\times	Lu <i>et al.</i> (2017) [55]	9.30	12.60	12.40	10.90	15.70	19.00	18.30	22.30	15.00	28.00	16.30
NN-based	✓	Chen <i>et al.</i> (2018) [43]	6.62	14.68	13.98	11.23	14.19	15.87	20.72	23.26	11.91	27.79	16.02
NN-based	✓	Chen <i>et al.</i> (2018) [†] [43]	3.96	12.16	11.13	7.19	11.11	13.06	18.07	20.46	11.84	27.22	13.62
NN-based	✓	Chen <i>et al.</i> (2019) [54]	2.77	8.06	8.14	6.89	7.50	8.97	11.91	14.90	8.48	17.43	9.51
NN-based	\times	Ours	3.78	7.91	8.75	5.96	10.17	13.14	11.94	18.22	10.85	25.49	11.62

TABLE 2.2: Without using ground-truth light or surface normals of this dataset at train time, our method supplies results that is comparable to the recent state-of-the-art [54]. The 1st and 2nd best performing methods are colored in light-red and dark-red respectively. ‘G.T. Normal’ column indicates the use of ground-truth normal at train time. Comparisons are done against well-known uncalibrated methods. † indicates the deeper version of the UPS-FCN model.

Type	G.T. Normal	Methods↓ Dataset →	Vase	Golf-ball	Face	Tablet 1	Tablet 2	Broken Pot	Average
Classical	\times	Nayar <i>et al.</i> (1991) [68]	28.82	11.30	13.97	19.14	16.34	19.43	18.17
NN-based	✓	Chen <i>et al.</i> (2018) [43]	35.79	36.14	48.47	19.16	10.69	24.45	29.12
NN-based	✓	Chen <i>et al.</i> (2019) [54]	49.36	31.61	13.81	16.00	15.11	18.34	24.04
NN-based	\times	Ours	19.91	11.04	13.43	12.37	13.12	18.55	14.74

TABLE 2.3: Comparison against recent uncalibrated deep photometric stereo methods and Nayar *et al.* [68] on our dataset. In contrast to our approach, Chen *et al.* [43] and Chen *et al.* [54] require ground-truth normal for training the network. We can observe that our method shows consistent behavior over a diverse dataset that is on average better than other methods. The two best-performing methods are shaded with light-red and dark-red color respectively.

2.5.2 Evaluation and Ablation Study

To measure the accuracy of the estimated surface normals, we adopt the standard mean angular error (MAE) metric as follows:

$$\text{MAE} = \frac{180}{\pi} \frac{1}{m} \sum_i^m \arccos(\tilde{n}_i^T n_i) \quad (2.27)$$

where, n is the number of photometric stereo images, and m is the number of object pixels. \tilde{n}_i and n_i denote the estimated and ground-truth surface normals. Following previous works [43, 54], we report MAE in degrees.

2.5.2.1 Results on DiLiGenT Dataset

Table(2.2) provides statistical comparison of our method against other uncalibrated methods on DiLiGenT benchmark. It can be inferred that our method achieves competitive results on this benchmark with an average MAE of 11.62 degrees, achieving the second best performance overall without ground-truth surface normal supervision. On the contrary, the best performing method [54] uses ground-truth normals during training, and therefore, it performs better for objects like *Harvest*, where imaging is deeply affected by discontinuities.

2.5.2.2 Results on Our Dataset

Table(2.3) compares our method with other deep uncalibrated methods on our proposed dataset. For completeness, we analyzed Nayar *et al.* [68] algorithm by using light sources data obtained using our approach. The results show that our method achieves the best performance overall. We observed that other deep learning methods cannot handle objects like *Vase* as they fail to model complex reflectance behavior. Similarly, Nayar *et al.* [68] results indicate that modeling interreflections alone is not sufficient. Since we not only model the effects of interreflections, but also the reflectance mapping associated with the geometry, our method consistently performs well.

2.5.2.3 Ablation Study

For this study, we validate the importance of robust initialization and interreflection modeling.

(a) Robust Initialization. To show the effect of initialization, we consider three cases. First, we use classical approach [11] to initialize inverse rendering network. Second, we replace the classical method with our robust initialization strategy. In the final case, we remove the weak-supervision loss from our method. Fig.2.9 shows MAE and image reconstruction loss curve per learning iteration obtained on *Cow* dataset. The results indicate that robust initialization allows the network to converge faster as outliers are separated from the images at an initial stage. Fig.2.8 shows the MAE of surface normals during initialization as compared to the results obtained using our method.

(b) Interreflection Modeling. To demonstrate the effect of interreflection modeling, we remove the function ζ_{n2} in Eq:(2.10) and use \mathbf{N}_o in image

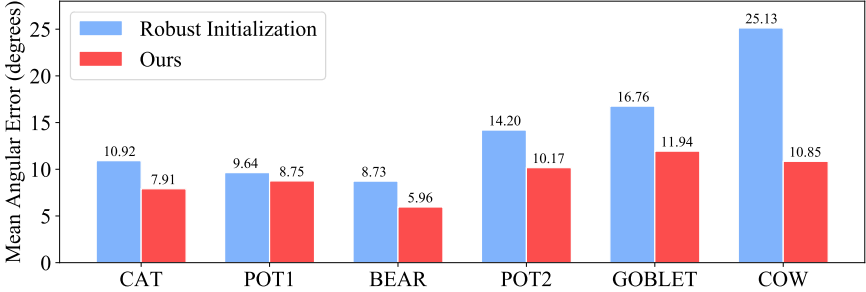


FIGURE 2.8: Surface normal accuracy achieved w.r.t its initialization.

reconstruction as in classical rendering. Fig.2.9 provides learning curves with and without interreflection modeling. As expected, excluding the effect of interreflections inherently impacts the accuracy of the surface normals estimates even if the image reconstruction quality remains consistent. Hence, it is important to explicitly constrain the geometry information.

2.5.2.4 Analysis on Estimated Light Source Directions

We aim to investigate the source directions' behavior predicted by the light estimation network. For that purpose, we use a well-known setup used for light calibration, i. e., a calibration sphere. Our renderings from the calibration sphere (see Fig.2.10(a)) has specular highlights and attached shadows, which provide useful cues for the light estimation network. Figures 2.10(b)-2.10(d) illustrate the x , y and z components of the estimated light source direction and ground-truth with respect to the images. We measured the MAE between these vectors as 6.31 degrees. We also observed that the x and y components match well with the ground-truth values. On the other hand, we observed fluctuations on the z component where the values slightly deviate from the ground-truth in a specific pattern. One possible explanation for this observation is that the network has a bias such that its behavior changes in the different regions of the lighting space. Since we generated the data by moving the light source on a circular pattern around z -axis, Fig. 2.10(d) also follows a similar pattern with the same frequency with x and y components' curves.

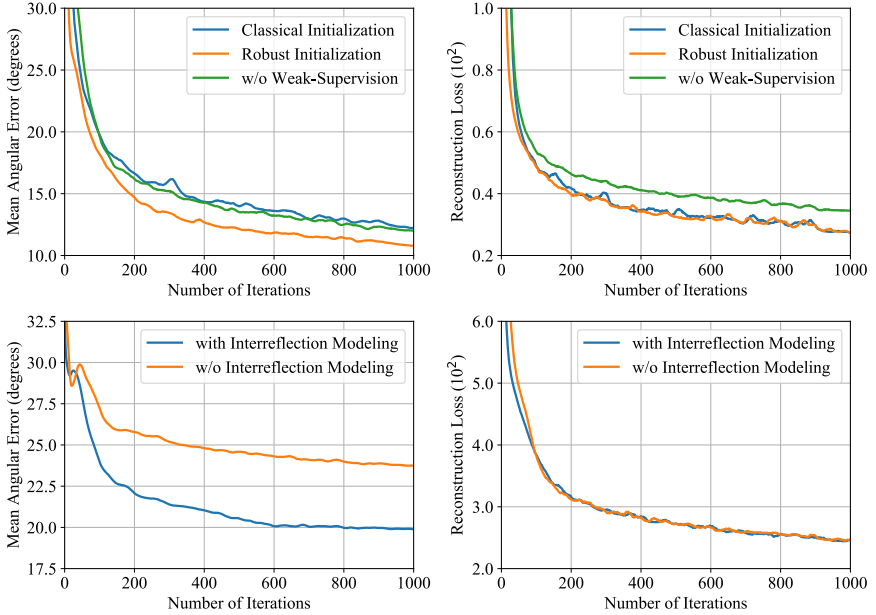


FIGURE 2.9: **Ablation Study:** We demonstrate the effect of robust initialization on *Cow* (top) and interreflection modeling on *Vase* (bottom).

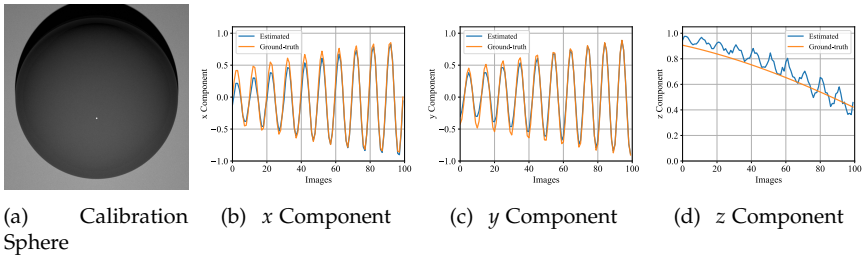


FIGURE 2.10: Light source directions obtained from the calibration sphere (a) using the light estimation network. We demonstrate the x, y and z components of the light direction vectors (b-d). The mean angular error between the ground-truth and estimated light directions is 6.31 degrees.

2.5.3 Case Study

This section provides the observation on the case study that we conducted for our proposed method. It is done to analyze the behavior of our method under different possible variations in our experimental setup. Such a study can help us understand the behavior, pros, and cons of our approach.

Case Study 1: *What if we use ground-truth light as input to inverse rendering network instead of relying on light estimation network?*

This case study investigates the reliability of our method. To conduct this experiment, we supplied ground-truth light source directions and intensities as input to the inverse rendering network and robust initialization. The goal is to study the expected deviation in the accuracy of surface normals when ground-truth light sources information is used, compared to the light calibration network. Table (2.4) compares our method’s performance with recent deep calibrated photometric stereo methods on our proposed dataset. The results show that our inverse rendering method achieves the best performance in the calibrated setting, although it does not use a training dataset like other deep-learning-based methods. Additionally, we observed that the CNN-PS model proposed by Ikehata [14] which performs per-pixel estimation using observation maps, may not provide accurate surface normals for interreflecting surfaces such as the *Vase* and the *Broken Pot*. Hence, we conclude that extracting information by utilizing the surface geometry is crucial for solving photometric stereo since all surface points affect each other.

Moreover, in Table (2.4), we show the comparison of our method’s performance under calibrated and uncalibrated settings. Our method achieves 12.68° MAE on average, using ground-truth light as input. At the same time, it reaches an average MAE of 14.74° utilizing the information of the light source obtained from the light estimation network. The difference between these two scores is 2.06 degrees, which indicates that the gap between the calibrated and uncalibrated settings is not substantial. Accordingly, we can conclude that our method is robust to the variations in the estimated lighting. Further, we observed that our method performs better with the network estimated light sources information in the categories like *Golf-ball*, *Face*. Hence, based on that observation, we can conclude that the availability of ground-truth calibration data is not a strict requirement for achieving better surface normals estimates in photometric stereo for all kinds of surface geometry.

Type	G.T. Normal	Methods, Dataset →	Vase	Golf-ball	Face	Tablet 1	Tablet 2	Broken Pot	Average
NN-based	✓	Ikehata (2018) [14]	34.00	14.96	16.61	16.64	12.32	18.31	18.81
NN-based	✓	Chen et al.(PS-FCN)(2018) [43]	27.11	15.99	16.17	10.23	5.79	8.68	14.00
NN-based	✗	Ours (Ground-truth light/ calibrated)	16.40	14.23	14.24	10.77	4.49	15.92	12.68
NN-based	✗	Ours (Estimated light/ uncalibrated)	19.91	11.04	13.43	12.37	13.12	18.55	14.74
		Diff. in MAE (Ours(Est)-Ours(GT))	+3.51	-3.19	-0.81	+1.60	+8.63	+2.63	+2.06

TABLE 2.4: Comparison of recent deep **calibrated** photometric stereo methods Ikehata [14] and Chen *et al.* [43] (PS-FCN) against our method under **uncalibrated** and **calibrated** setting. For testing our method under the calibrated setting, we evaluate the performances assuming that ground-truth light source directions and intensities are available. Note that Chen *et al.* [43] and Ikehata [14] additionally uses ground-truth surface normals for training, in contrast to our method. The last row shows the difference between our method results when used under uncalibrated and calibrated setting respectively. We can see that the average difference in MAE between the two settings of our method is not significant.

Case Study 2: *What if we use noisy images?* Photometric stereo uses a camera acquisition setup, and this implies that noise due to imaging is inevitable. This case study aims to investigate the behavior of our method on different noise levels. To study such a behavior, we synthesized images by adding noise to the images of our proposed dataset. Fig.2.11 compares the performance of our method under different noise levels. For this case study, we used zero-mean Gaussian noise with different standard deviations ($\sigma=0.05$, $\sigma=0.1$, $\sigma=0.2$). The quantitative results indicate that increasing the noise generally degrades the performance. We observed that the behavior under different noise levels varies among the subjects.

Case Study 3: *Photometric stereo on concentric surfaces with deep concavities and large surface discontinuity.*

To study our photometric stereo method’s boundary-condition, we took a complex geometric structure with concentric surfaces, deep-concavities, and large discontinuities for investigation. Accordingly, we synthesized the *Rose* dataset using the same dome-settings outlined before. Fig.2.12 shows the qualitative results obtained on this dataset. Our method achieves 60.82 degrees of MAE on this particular example. We observed that our approach could not handle this complex geometry because the surface is highly discontinuous with excessive gaps between the leaves. The scene is also affected by occlusions and cast shadows, and therefore, modeling the interreflections for this case seems very difficult.

Noise Std (σ)	Vase	Golf-ball	Face	Tablet1	Tablet2	Broken Pot	Average
0.0	19.91	11.01	13.43	12.37	13.12	18.55	14.73
0.05	21.96	11.54	12.94	17.25	11.22	17.22	15.36
0.1	25.01	11.83	15.12	18.80	11.55	19.06	16.90
0.2	24.41	14.25	19.62	21.27	10.07	18.16	17.96

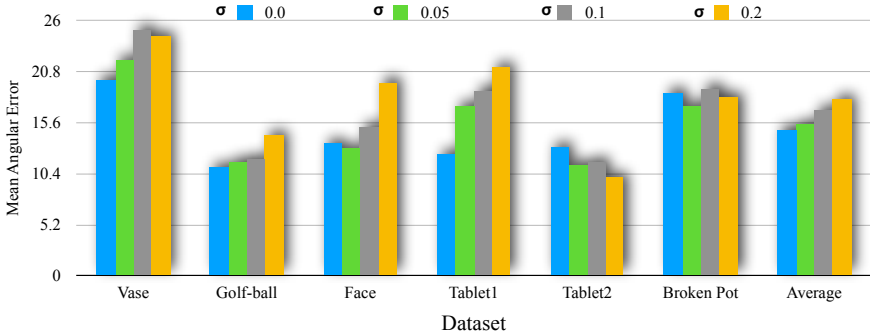


FIGURE 2.11: The performance of our method against different noise levels. We used zero-mean Gaussian noise ($\mu = 0$) with different standard deviations (σ). We observed that increasing the noise level generally degrades the performance. Still, the behavior under different noise levels varies among the subjects as the performance depends on the signal-to-noise ratio of the images.

Though our method applies to a broad range of objects, our interreflection modeling is inspired by Nayar *et al.* [68] formulation, which may not hold for all kinds of surfaces. The interreflection modeling computes depth from the normal map under the continuous surface assumption, which fails in this case study. Furthermore, it models continuous surfaces with discrete facets. Due to such limitations, our method may not be suitable for concentric surfaces with deep concavities and large discontinuities. In such cases, the interreflection effect is very complicated, and our approach may disappoint to model such complex light phenomena.

2.6 LIMITATIONS AND FUTURE EXTENSION

Discrete facets assumption of a continuous surface for computing depth and interreflection kernel may not be suitable where the surface is discontinuous in orientation, e. g., surface with deep holes, concentric rings, etc.. As a result, our method may fail on surfaces with very deep concavities

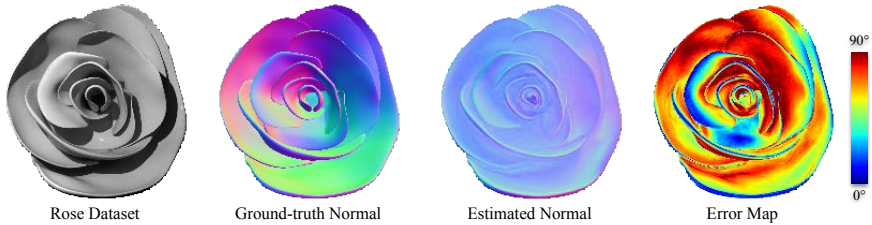


FIGURE 2.12: **Failure case:** Qualitative results on the Rose dataset.

and cases related to naturally occurring optical caustics. As a second limitation, the light estimation network may not resolve GBR ambiguity for all kinds of shapes. Presently, we did not witness such ambiguity with the light calibration network as it is trained to predict lights under non-GBR transformed surface material distribution.

Our proposed method enables the application of photometric stereo on a broader range of objects. Yet, we think that there are possible future directions to extend it. Firstly, our method is generally a two-stage framework that utilizes a light estimation network and inverse rendering network in separate phases during inference. As an extension of our work, we aim to combine those stages in an end-to-end framework where light, surface normals, and reflectance values are estimated simultaneously. Secondly, our method uses a physical rendering equation for image reconstruction that is not sufficient for modeling all physical interactions between the object and the light. We believe that an improved rendering equation with additional physical constraints will allow better normal estimates. In addition to that, our method utilizes a specular-reflectance map inspired by the Phong reflectance model. Using other sophisticated variants of specular-reflectance map such as the Blinn-Phong reflection model [83] may further advance our approach. Finally, we observed that our method is very convenient for practical usage as it doesn't require ground-truth normals for supervised training. However, it could be possible to improve performance by utilizing training data in a similar framework.

2.7 CONCLUSION

From this work, we conclude that uncalibrated neural inverse rendering approach with explicit interreflection modeling enforces the network to model complex reflectance characteristics of objects with different material

and geometry types. Without using ground-truth surface normals, we observed that our method could provide comparable or better results than the supervised approaches. And therefore, our work can enable 3D vision practitioners to opt for photometric stereo methods to study a broader range of geometric surfaces. That's said, image formation is a complex process, and additional explicit constraints based on the 3D surface geometry types, material, and light interaction behavior could further advance our work.

MULTI-VIEW PHOTOMETRIC STEREO

3.1 MOTIVATION

In the coming decade, dense 3D data acquisition of objects is likely to become one of the most important problems in computer vision and industrial machine vision. Moreover, it can be helpful for a wide range of other cutting-edge scientific disciplines such as metrology [1], geometry processing [2], forensics [3], etc. Many active and passive 3D reconstruction approaches or pipelines were proposed to solve 3D reconstruction of objects [1, 3–5, 84]. However, it is widely accepted that methods such as structure-from-motion [6, 7], multi-view stereo [1], photometric stereo [11, 61, 85], and other standalone approaches [86–90] are not sufficient on their own to provide detailed and precise 3D reconstruction for all kinds of surfaces [3]. Therefore, methods that combine complementary surface estimates by leveraging more than one modality are often preferred [57, 91].

Among the passive 3D shape acquisition methods, multi-view stereo (MVS) has become the most popular approach [6, 92, 93], especially after the proliferation of cheap digital cameras for high-quality imaging. Yet, MVS works best for Lambertian textured surfaces and gives unreliable results for non-textured objects with non-Lambertian surface reflectance property. Moreover, high-frequency surface details such as indentations and scratches are difficult to recover using MVS methods (see Fig.3.1(a)).

On the other hand, photometric stereo (PS) is magnificent at recovering high-frequency surface details using light-varying images [11]. It is also effective for non-textured, and non-Lambertian surfaces [54]. PS allows the recovery of per-pixel depth of the object by integration of the estimated surface normals [94]. However, it suffers from the main shortcoming: The recovered surface profile is globally deformed by a low-frequency distortion [57]. Such distortion is likely due to numerical integration of the surface normal map without explicit constraints between multiple disconnected regions of the object's surface [57, 95, 96] or non-directional lighting effects (see Fig.3.1(b)). Further, existing methods generally assume isotropic material objects and may fail to handle objects with anisotropic material like a piece of wood [14, 15, 71, 72].

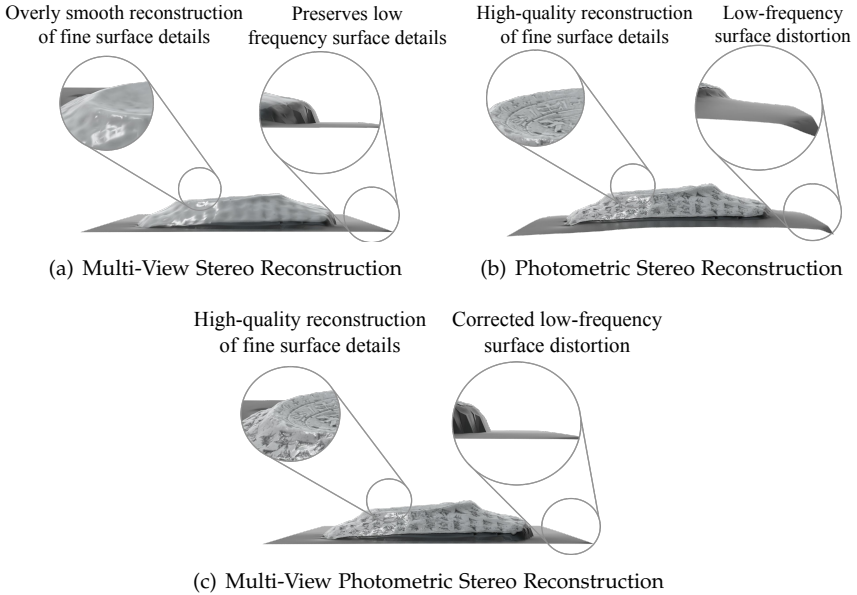


FIGURE 3.1: (a) MVS reconstruction preserves the plane geometry but loses the finer details. (b) PS captures the fine geometric details but introduces global distortion. (c) Multi-view photometric stereo (MVPS) can handle the high and low-frequency surface components quite well. It overcomes the high and low-frequency surface reconstruction problem by suitably utilizing the complementary surface estimates.

When it comes to the accuracy of recovered 3D shapes for its use in scientific and engineering purposes (metrology), methods that use only MVS or PS suffer [88, 93, 97, 98]. As a result, a mixed experimental setup such as multi-view photometric stereo (MVPS) is generally employed [91]. In such a setup, complementary modalities are used to obtain better surface measurements, which are otherwise unavailable from an individual sensor or method. Accordingly, similar fusion-based strategies gained popularity for surface estimation [5, 57, 99–101]. One may also prefer to use two or more active sensors to receive the surface data estimates for fusion. Nevertheless, this chapter focuses on the MVPS setup, where the subject is placed on a rotating base and for each rotation multiple images are captured using one LED light source at a time. The major motivation for such an approach is that the active range scanning strategies used for object’s 3D

acquisition, such as structured light [102–104], 3D laser scanners [105], RGB-D sensors [5] are either complex to calibrate or provide noisy measurements or both. Further, these measuring techniques generally provide incomplete range data with outliers that require serious efforts for refinement.

Existing state-of-the-art methods to the MVPS problem generally apply a sequence of steps [57, 91, 106] *(i)* procure the 3D position measurement using multi-view images or 3D scanner *(ii)* estimate surface orientation or iso-depth contours using photometric stereo methods, and *(iii)* fuse the surface orientation and 3D position estimates to recover 3D geometry using appropriate mathematical optimization. Now, several fusion strategies exist that can combine these alternate sources of information for better 3D shape reconstruction. [57, 91, 99, 106–108]. Of course, the precise steps taken by such approaches can provide better results, yet they rely heavily on explicit mathematical modeling [56, 57, 91, 101, 107–109], and complex multi-staged network design [106] which are complicated to execute. In contrast, this thesis presents three different approaches which are simple and general. Our approaches can suitably combine the surface details coming from PS with multi-view information for favorable gain in performance and applicability.

3.2 RELATED WORK

Here, we review important MVS, view-synthesis and MVPS methods related to our work.

3.2.1 *Multi-View Stereo (MVS) Methods*

It aims at reconstructing a plausible 3D geometry of the object from a set of images alone [1]. The working principle of MVS generally pivots around improving the dense image feature correspondence (local similarity) and camera parameters across images for triangulation [1, 89, 90, 93, 110, 111]. Recent developments in machine learning have led to the renovation of traditional MVS methods via deep-learning frameworks.

Roughly, it can be divided into four to five categories [1, 112]. *(i)* Volumetric methods require bounding box knowledge containing the subject. These methods estimate the relation between each voxel and the surface element, and their accuracy is greatly affected by voxel grid resolution [113–117]. *(ii)* Patch-based approach utilize the Barnes *et al.* [118] randomized correspondence idea in the scene space. Generally, these methods generate

random 3D planes in the scene space and refine the depth representation and normal fields based on photo-consistency measures [93, 98, 111, 119]. (iii) Depth map reconstruction-based methods use a reference image with different source images under calibrated settings to improve the overall depth estimation [98, 120–124]. (iv) Point cloud-based methods operate on the 3D points and process the initial sparse point sets to densify the results [93, 125]. (v) Distributed structure-from-motion methods utilize the notion of motion averaging to improve large-scale 3D reconstruction [59, 126, 127]. Recently, deep neural networks have been widely adopted for MVS, which provide better performance than traditional MVS methods [112]. Earlier work in this area uses CNN’s for two-view [128] and multi-view stereo [129]. Lately, the learning-based MVS rely on the construction of 3D cost volume and use the deep neural networks for regularization and depth regression [130–136]. As most of these approaches utilize 3D CNN for cost volume regularization—which in general is computationally expensive, the majority of the recent work is motivated to meet the computational requirement with it. Few methods attempt to address it by down-sampling the input [135, 136]. Other attempts to improve the computational requirements uses sequential processing of cost volume [137], cascade of 3D cost volumes [112, 138–140], small cost volume with point-based refinement [130], sparse cost volume with RGB and 2D CNN to densify the result [141], learning-based patch-wise matching [134, 140] with RGB guided depth map super-resolution [140].

3.2.2 View Synthesis for 3D Reconstruction

In recent years, view synthesis methods, in particular, the Neural Radiance Fields (NeRF) method for scene representation, have proposed an interesting idea to recover the 3D geometry of the scene using multi-view images [88]. NeRF has generated a new wave of interest in 3D computer vision and has led to several follow-ups in 3D data acquisition from images [142–152]. NeRF uses a fully-connected deep neural network to represent the scene geometry and radiance information implicitly. It renders photo-realistic views by implicitly encoding surface volume density and color via a multi-layer perceptron (MLP). Once MLP is trained, the 3D shape can be recovered by using the estimated volume density. One can model the implicit object’s surface normals and radiance field in a unified way but may need volume occupancy information. Further, to couple the surface normals term with the surface 3D point coming from the same mea-

surement source may not provide significant gain [152], especially when dealing with PS images, where shading plays a critical role.

3.2.3 Multi-View Photometric Stereo (MVPS) Methods

MVPS is a classical 3D shape acquisition setup introduced by Hernandez *et al.* [56]. The proposed setup is composed of a turn-table with an object placed at the center of the table for multi-view and photometric stereo image acquisition. The MVPS algorithm proposed by Hernandez *et al.* [56] combines the multi-view PS results and corrects its low-frequency surface distortion via multi-view geometric constraint. Yet, the method works well only for specific parametric BRDF models [91]. Later, Park *et al.* [62, 108] proposed an uncalibrated MVPS method to recover fine geometric details of the shape. It requires an initial coarse mesh with a 2D displacement map for optimization leading to shape recovery. Still, the method cannot handle objects with diverse surface reflectance properties. Further, it often fails on textureless regions with non-Lambertian reflectances [91]. Logothetis *et al.* [153] used a volumetric approach to solve MVPS via a variational framework. Recently, Li *et al.* [91] proposed a systematic geometric approach to MVPS showing state-of-the-art results. However, it consists of several carefully crafted explicit geometric modeling steps such as iso-depth contour estimation, tracing contours, multi-view depth propagation, point sorting, shape optimization, etc. It requires a successful execution of each of these steps applied in a sequel to recover the surface. Hence, re-implementing such an approach is complex, strained, and time-consuming. Furthermore, the use of classical PS and MVS in their pipeline has its limitations; for e.g., classical PS, MVS may not handle a wide range of objects with non-Lambertian properties, etc.

Our literature review shows a lack of a robust modern neural network approach to solve MVPS, which is excellent at learning the object's surface properties from data. Existing attempts for the task consider co-located lighting setups [157], have constraints on the object topology [158], or are restricted to certain material types [159]. Thus, it has become increasingly evident that a simple and effective learning method is essential for the MVPS problem.

Other related work uses an active 3D sensing modality with PS. For instance, Nehab *et al.* [57] used a structured lighted scanner, whereas Chatterjee *et al.* [101] relied on a RGB-D sensor to measure the 3D position data. Instead, our work focuses on the classical MVPS setup [56, 62, 91]. It

Method	Base data	Shape representation	Optimization
Mostafa <i>et al.</i> [99]	Laser, PS images	Neural Network	Back-Propagation + EKF [154]
Nehab <i>et al.</i> [57]	Scanner [105], PS images	Mesh	Mesh (Linear)
Hernandez <i>et al.</i> [56]	MVPS images	Mesh	Mesh (Coupled)
Park <i>et al.</i> [108]	MVPS images	Mesh + 2D displacement	Parameterized 2D (Sparse Linear)
Li <i>et al.</i> [91]	MVPS images	Depth Contour + 3D points	Poisson surface [155] + [57]
Logothetis <i>et al.</i> [153]	MVPS images and SDF	Parameterized SDF	Variational Approach
Ours	MVPS images	Multi-layer Perceptron	Adam [156]

TABLE 3.1: Previous work on passive approach to 3D shape recovery using orientation and range estimates. Despite Mostafa *et al.* [99] and Nehab *et al.* [57] use active modality, we included them for completeness.

has some apparent advantages over active 3D scanning methods. Firstly, it is easy and cost-effective to perform high-quality image acquisition, as regular cameras are sufficient. Secondly, it is relatively noise-free and gives dense per-pixel information compared to incomplete range data with outliers provided by structured light [102], 3D laser scanner [105], and depth sensors [5]. Although the developments in RGB-D and other portable active sensors have led to success with volumetric methods on shading-based refinement [107, 160, 161], our work relies on image data for this problem. Much of the existing work that utilizes the images for complementary measurements uses explicit mathematical modeling for precise 3D geometry [56, 91, 106, 108, 153, 162]. Table (3.1) summarizes some of the recent and early developments in the area of multi-view photometric stereo.

3.3 PRELIMINARIES

3.3.1 MVPS Setup

Hernández *et al.* [56] proposed the introductory MVPS acquisition setup. It is composed of a turntable arrangement, where light-varying images (PS images) of the object placed on the table are captured from a given viewpoint. Note that the camera and light sources' position remains fixed, and only the table rotates, providing a new viewpoint (v) of the object per rotation. For every table rotation, PS images for each light source are captured and stored (see Fig.3.2). In this chapter, we consider such a turntable acquisition setup for our analysis. We also assume a calibrated setting, i.e., all the light source directions and camera calibrations are known.

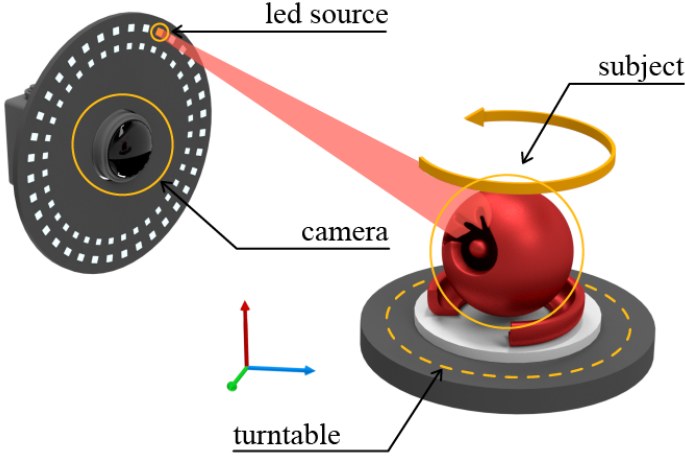


FIGURE 3.2: General setup for multi-view photometric stereo. The object is placed on a base with fixed rotation. A camera is placed at the center of the disk to capture images. LEDs are placed in a concentric ring for controlled light setup. The 3D model used for the above illustration is taken from [88] dataset.

3.3.2 Notation and Definitions

Before describing the details of our approach, we introduce the notation and the image formation model for the MVPS setting. We denote $\mathcal{I}^v = \{I_1^v, \dots, I_{N_p}^v\}$ as the set of N_p PS images for a given view $v \in [1, N_m]$. For simplicity, let's assume a single view case for an object with reflective surface, whose appearance can be encoded by a bidirectional reflectance distribution function (BRDF) Φ_s with surface normals $\mathbf{N} \in \mathbb{R}^{3 \times p}$. Here, p symbolizes the total number of pixels. When object's surface is illuminated by a point light source positioned in the direction $l_k \in \mathbb{R}^{3 \times 1}$, then the image $I_k^v \in \mathbb{R}^{p \times 1}$ captured by a camera in the view direction $\mathbf{v} \in \mathbb{R}^{3 \times 1}$ can be modeled as

$$I_k^v = e_k \cdot \Phi_s(\mathbf{N}, l_k, \mathbf{v}) \cdot \max(\mathbf{N}^T l_k, 0) + \epsilon_k \quad (3.1)$$

where, $e_k \in \mathbb{R}_+$ denotes the intensity, ϵ_k is an additive error and $\max(\mathbf{N}^T l_k, 0)$ accounts for the attached shadows. Using such an acquisition experimental setup, it is easy to recover two types of surface priors: (i) 3D position per pixel ($\mathbf{p}_i \in \mathbb{R}^{3 \times 1}$) of the object using multi-view stereo images (ii) surface normal for each surface point ($\mathbf{n}_i^{ps} \in \mathbb{R}^{3 \times 1}$) using light varying images [1, 7,

11, 110]. Hence, by design, the problem boils down to effective use MVS and PS surface priors, light varying images, light and camera calibration data for high-quality dense 3D surface recovery.

3.3.3 Benchmark

Datasets. We use DiLiGenT-MV benchmark dataset [91, 163] for our experiments, statistical evaluations, and ablation studies. The DiLiGenT-MV dataset consists of complex BRDF images taken from 20 viewpoints. For each viewpoint, 96 images are captured, each illuminated by a light source in a different known position (calibrated). The dataset includes 5 objects (BEAR, BUDDHA, COW, POT₂, READING) with complex surface profiles and its images are captured under large illumination changes. For creating this dataset, the distance between the camera and object is set to 1.5 m [91]¹.

Although the DiLiGenT-MV benchmark consists of challenging objects with non-Lambertian surfaces, all provided objects satisfy isotropic BRDF property. Therefore, we simulated a new dataset consisting for objects with anisotropic and glossy surfaces. Similar to classical setup, we simulated our dataset using a turntable setup with 36 angle rotations. We place 72 light sources in a concentric way around the camera (see Fig.3.2) and rendered images corresponding to each light source. We use licensed Houdini software to simulate our setup and render MVPS images of a single object 3D model taken from NeRF synthetic dataset [88] with three different material types (Wood, Gray, Red)². The Wood category is rendered to study anisotropic material behavior and the other two categories to analyze performance on texture-less glossy objects. We rendered images at 1280×720 resolution to better capture the object details.

Evaluation Metrics. Our quantitative analysis is based on Chamfer- L_1 distance, precision and \mathcal{F} -score on the reconstructed and ground-truth point sets: $\mathcal{R}, \mathcal{G} \subset \mathbb{R}^3$. For a single reconstructed point $r \in \mathcal{R}$, distance to the ground-truth is defined as follows:

$$d_{r \rightarrow \mathcal{G}} = \min_{g \in \mathcal{G}} \|r - g\|. \quad (3.2)$$

The individual distance measures are accumulated to define Chamfer- L_1 distance and \mathcal{F} -score as follows:

¹ This dataset is publicly available for research purpose at:
<https://sites.google.com/site/photometricstereodata/mv>

² CC-BY-3.0 license.

$$CD = \frac{1}{2|\mathcal{R}|} \sum_{x \in \mathcal{R}} d_{x \rightarrow \mathcal{G}} + \frac{1}{2|\mathcal{G}|} \sum_{x \in \mathcal{G}} d_{x \rightarrow \mathcal{R}}, \quad (3.3)$$

$$\mathcal{F}(\tau) = \frac{2P(\tau)R(\tau)}{P(\tau) + R(\tau)}, \quad (3.4)$$

where

$$P(\tau) = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} [d_{r \rightarrow \mathcal{G}} < \tau], \quad (3.5)$$

$$R(\tau) = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} [d_{g \rightarrow \mathcal{R}} < \tau], \quad (3.6)$$

stand for precision and recall measures respectively. Here, $[\cdot]$ is the Iverson bracket and τ is the distance threshold.

3.4 NEURAL RADIANCE FIELDS APPROACH TO MVPS

In this section, we present a simple and modern approach (NR-MVPS) that can suitably introduce the surface details coming from PS to neural radiance field representation of the scene for favorable performance gain. Our method is inspired by the idea of local region-based techniques for volumetric illumination, which can render more realistic images [164, 165]. The ray-traced volume rendering approximates the surface normals using the gradient of the density along each (x, y, z) direction in volumetric space [165, 166]. However, to use such a notion for our problem setup, we must know the occupancy of the point in the volume space. To keep it simple, we utilize the surface normal from PS as the gradient density information for each sample point along the ray to reconstruct MVPS images as close as possible and recover 3D geometry. While one could recover depth from surface normals and then infer the occupancy of the volume density, we know that normal integration may lead to inaccurate depth estimates, hence incorrect occupancy information. So, we adhere to the proposed idea of using local gradients and show the validity of our method via experimental evaluations.

Our approach first estimates the surface normal for all the views using a deep photometric stereo network which is trained independently in a supervised setting. We use sample spatial location \mathbf{x}_i , the viewing direction \mathbf{v} , and the object’s surface normal for our MVPS representation. Surface normals for each 3D sample point along all known view directions are

employed. For 3D reconstruction, our method optimizes a multi-layer perceptron that regresses from the position, view direction, and surface normal to single volume density and view-consistent RGB color. We use Fourier feature encoding on both positional and surface orientation data before passing them to the network [167]. Experimental results show that our method provides comparable or better results than previously proposed complex multi-view photometric stereo methods [56, 57, 99, 101, 107, 109], stand-alone state-of-the-art multi-view method [140], and view synthesis approach [88].

3.4.1 Contributions

In this section, we make the following contributions:

- While previous multi-stage fusion methods to MVPS are very complex, we propose a much simpler continuous volumetric rendering approach that uses the local density gradient effects in MVPS image formation.
- Our work takes an opportunistic approach that exploits the complementary source of information in MVPS setup via two sets of representation i. e., volumetric and surface.
- Despite being much simpler than fusion methods, our approach achieves better or comparable results than the state-of-the-art [91, 108], as well as stand-alone methods such as multi-view [140, 152], photometric stereo [14], and continuous volumetric rendering [145] methods.

3.4.2 Method

The proposed approach considers the notable image formation models used in computer vision and computer graphics i. e., photometric stereo image formation model [11] and the rendering equation [168]. These imaging models has its advantage depending on the experimental setup. Since we are solving the well-known MVPS problem, we can exploit the benefit of both photometric and multi-view stereo setup. For our work, we assume that the subject under study is a solid texture-less object, which is often studied in photometric stereo research [91]. Next, we describe the deep PS network, followed by our neural radiance field modeling for a multi-view photometric stereo setup.

3.4.2.1 Deep Photometric Stereo Network

As modeling unknown reflectance properties of different objects remains a fundamental challenge, we utilize deep neural networks to learn the complicated BRDF's from input data. We leverage the observation map based CNN model to estimate surface normal under a calibrated setting [14]. Unlike other supervised methods, it has rotational invariance property for isotropic material, handles unstructured images and lights well, and above all provides best performance known to us with acceptable inference time.

Observation map. For each pixel, this map contains the normalized observed intensity values due to all the light sources. In a general PS setup, the light sources are located in a concentric way. Thus, a one-to-one mapping between the light source position $(l_x, l_y, l_z) \in \mathbb{R}^3$ and corresponding x-y coordinate projection $(l_x, l_y) \in \mathbb{R}^2$ is possible. Note that $l_x^2 + l_y^2 + l_z^2 = 1 \forall l_k$ i. e., the unit vector in the direction of source. We construct the observation map $\Omega_i^v \in \mathbb{R}^{w \times w}$ for each pixel i using its intensity value across all the N_p images as outlined in **Algorithm 1**.

Algorithm 1: Observation Map Construction

```

1 for  $i \leftarrow \{1, \dots, p\}$ 
2   for  $j \leftarrow \{1, \dots, N_p\}$ 
3      $\Omega_i^v \left( \zeta \left( w \cdot \frac{(l_x+1)}{2} \right), \zeta \left( w \cdot \frac{(l_y+1)}{2} \right) \right) = \frac{I_j^v(i)}{\eta_i e_i}$ 
4   end
5 end
```

Here, w is the size of the observation map and the function $\zeta : \mathbb{R} \mapsto \mathbb{Z}_{0+}$. The scalar η_i for i^{th} pixel is $\eta_i = \max(I_1^v(i)/e_1, \dots, I_{N_p}^v(i)/e_{N_p})$ is the normalizing constant. Since the projected source vectors can take values from $[-1, 1]$, they are scaled appropriately to get positive integer values.

Architecture Details. Inspired by the DenseNet design [169], the PS network first performs a convolution with 16 output channels on the input observation maps. The other part of the network consists of two dense blocks, a transition layer, and two dense layers, which is then followed by a normalization layer to recover surface normals as output. The dense block is composed of one ReLU layer, one 3×3 convolutional layer and a 0.2 dropout layer. The transition layer is composed of a ReLU layer, 1×1 convolutional layer, 0.2 dropout layer, and an average pooling layer, which is placed in between the two dense blocks to modify the feature map size.

We train PS network end-to-end separately in a supervised setting. The l_2 (MSE) loss between estimated and the ground-truth normals is minimized using Adam optimizer [156] (see Fig. 3.3).

3.4.2.2 Neural Radiance Fields Representation

Recently, volume rendering techniques for view synthesis, in particular NeRF [88] has shown a great potential for learning 3D scene from multi-view images. It represents a continuous scene as a 5D vector-valued function i. e., $\mathbf{x} = (x, y, z)$ for each 3D location and $(\theta_{\mathbf{v}}, \phi_{\mathbf{v}})$ for every 2D viewing direction. Given multi-view images with known camera pose, NeRF approximates the assumed continuous 5D scene representation with a MLP that maps the $(\mathbf{x}, \theta_{\mathbf{v}}, \phi_{\mathbf{v}})$ to RGB color \mathbf{c} and volume density $\sigma \in \mathbb{R}_+$. Using the classical volume rendering work [170], it models the expected color $C(\mathbf{r})$ of the camera ray $r(t) = \mathbf{o} + t\mathbf{v}$ with near and far bound t_n, t_f as:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \quad (3.7)$$

$$\text{where } T(t) = \exp\left(-\int_{t_n}^t \sigma(\mathbf{r}(s))ds\right). \quad (3.8)$$

Here, \mathbf{v} is the unit viewing direction. $T(t)$ is the accumulated transmittance along ray from t_n to t which caters the notion that how much light is blocked earlier along the ray. Given N samples along the ray, the continuous integral in Eq:(3.7) is approximated using the quadrature rule [171]:

$$\tilde{C}(\mathbf{r}) \approx \sum_{i=1}^N T_i \alpha_i(\mathbf{x}_i) \mathbf{c}_i(\mathbf{x}_i, \mathbf{v}), \quad (3.9)$$

$$\text{where } \alpha_i(\mathbf{x}_i) = \left(1 - \exp(-\sigma(\mathbf{x}_i)\delta_i)\right), \quad (3.10)$$

$$\text{and } T_i = \prod_{j=1}^{i-1} (1 - \alpha_j). \quad (3.11)$$

Here, δ_i is the distance between the adjacent discrete samples and α_i encapsulate how much light is contributed by the ray i . By construction, Eq.(3.9) approximates the alpha composited color as a weighted combination of all sampled colors \mathbf{c}_i along the ray. For more details we refer the readers

to Mildenhall *et al.* work [88]. Now, lets have a closer look at the following rendering equation [152, 168]:

$$L_r(\mathbf{x}_i, \omega_o) = L_e(\mathbf{x}_i, \omega_o) + \int_S \Phi_r(\mathbf{x}_i, \mathbf{n}_i, \omega_j, \omega_o) L_j(\mathbf{x}_i, \omega_j) (\mathbf{n}_i \cdot \omega_j) d\omega_j. \quad (3.12)$$

The above equation suggests that the rendering of a surface element \mathbf{x}_i depends on the emitted light in the scene and the bidirectional reflectance distribution function (BRDF) Φ_r describing reflectance and the color property of the surface accumulated over the half-sphere S centered at \mathbf{n}_i . The significance of including surface normal to Φ_r in the rendering equation is put forward by Yariv *et al.* work [152]. Here, ω_o, ω_j is the outgoing light direction and negative direction of the incoming light, respectively. Φ_r accounts for proportion of light reflected from ω_j towards ω_o at \mathbf{x}_i . L_o the radiance directed outward along ω_o from a particular position \mathbf{x}_i , and L_e is the emitted radiance of the light. In general, the light does not always hit the surface orthogonally, and so the dot product between the \mathbf{n}_i and ω_j attenuates the incoming light at \mathbf{x}_i . Hence, by restricting Φ_r and light radiance functions (L_o, L_e), which the radiance fields approximation can represent, we condition the color function to include the notion of density gradient for image rendering. Consequently, we use the normal estimated from PS to condition \mathbf{c}_i in Eq.(3.9). We rely on a deep photometric stereo network to estimate surface normals, overcoming BRDF modeling complications and providing us an excellent surface detail. Hence, our method has the inherent benefit over an entangled surface normal representation in image rendering [152]. Accordingly, we modify the Eq.(3.9) as follows:

$$\tilde{\mathbf{C}}(\mathbf{r}) \approx \sum_{i=1}^N T_i \alpha_i(\mathbf{x}_i) \mathbf{c}_i(\mathbf{x}_i, \mathbf{n}_i^{ps}, \mathbf{v}) \quad (3.13)$$

Further, adding image features in Eq.(3.13) could be advantageous as in [142, 151]. However, MVPS setup generally deals with non-textured surfaces where using image features is not much of help. Still, an obvious advantage with the setup is that better surface details can be captured from shading. For simplicity, we use surface normals to condition volume rendering and refrain from relying on image features. So, our approach blends density gradient information into the continuous volume rendering formulation bypassing the explicit volume occupancy information. Concretely, we feed surface normals for each 3D sample point along the viewing direction to the neural rendering network.

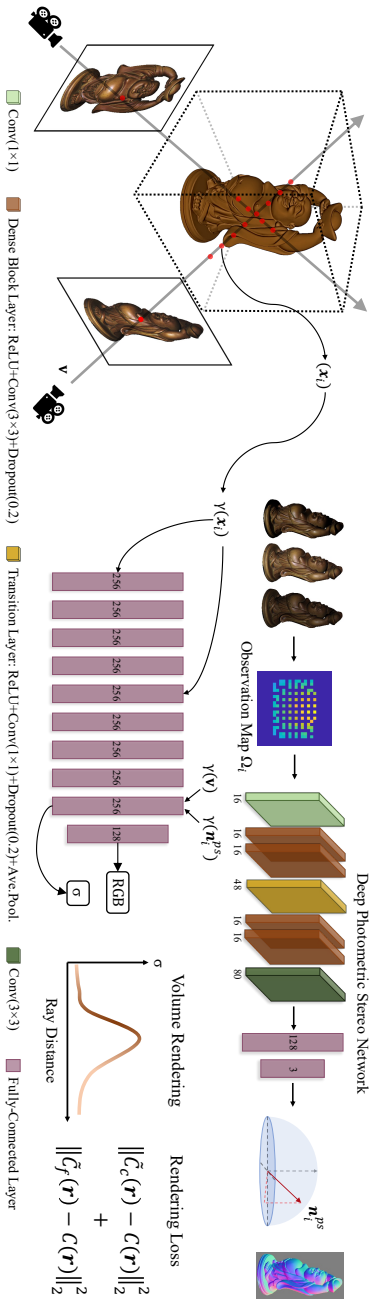


FIGURE 3-3: **NR-MVPS Overview:** The deep photometric stereo network predicts surface normals of the object from each viewpoint using PS images. We model multi-view neural radiance fields by introducing gradient knowledge from PS network output in the density space for solving MVPS. Our work takes a much simpler approach than existing state-of-the-art multi-staged MVPS methods showing comparable accuracy.

Optimization and Loss Function. Following neural radiance fields optimization strategy [88], we encode each sampled position \mathbf{x}_i along the ray, viewing direction \mathbf{v} , and photometric stereo surface normal \mathbf{n}_i^{ps} using the Fourier features $\gamma(\mathbf{x}) = [\sin(\mathbf{x}), \cos(\mathbf{x}), \dots, \sin(2^{L-1}\mathbf{x}), \cos(2^{L-1}\mathbf{x})]$. We used $L = 10$ for $\gamma(\mathbf{x}_i)$, $L = 4$ for $\gamma(\mathbf{n}_i^{ps})$ and $L = 4$ for $\gamma(\mathbf{v})$. For efficient estimation of continuous integral (Eq.(3.13)) using quadrature rule, we use the stratified sampling approach to partition the near and far bound $[t_n, t_f]$ into N evenly-spaced discrete samples [171].

$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n), t_n + \frac{i}{N}(t_f - t_n)\right]. \quad (3.14)$$

We employed MLP (Multi-layer Perceptron) to optimize the following loss function:

$$\mathcal{L}_{mvps} = \sum_{\mathbf{r} \in \mathcal{B}} \|\tilde{C}_c(\mathbf{r}) - C(\mathbf{r})\|_2^2 + \|\tilde{C}_f(\mathbf{r}) - C(\mathbf{r})\|_2^2. \quad (3.15)$$

where, \mathcal{B} denotes set of all rays in the batch. We used hierarchical volume sampling strategy to densely evaluate neural radiance field network for N query point along each ray. To that end, we first sample N_c points using the stratified sampling strategy and optimize the coarse network $\tilde{C}_c(\mathbf{r})$ (Eq.(3.13)). With the known output distribution of the coarse network, we sample N_f points using inverse transform sampling to optimize the fine network $\tilde{C}_f(\mathbf{r})$. In Eq.(3.15), the variable $C(\mathbf{r})$ is the observed color for the pixel (see Fig.3.3).

3.4.3 Experiments and Results

3.4.3.1 Implementation Details

(a) Deep-Photometric Stereo Network. We train the deep photometric stereo network on the CyclesPS dataset [14]. We used the Adam optimizer [156] with a learning rate of 10^{-3} and trained for 10 epochs. A per-pixel observation map with a size of 32×32 is used at train and test time. During testing, we applied the network on 96 PS images per subject from DiLiGenT-MV dataset.

(b) MLP Optimization for MVPS. For each object in DiLiGenT-MV dataset, we optimize a dense fully-connected neural network that is composed of 8 fully-connected ReLU layers. Each layer is composed of 256 channels. In addition to the density output from the 8th layer, 24 dimensional view-direction and surface normal Fourier features are then fed at the 9th layer for

rendering (see Fig.3.3). We use $N_c = 64$ points for the coarse network and $N_f = 128$ points for the fine network. We optimize the network parameters for 30 epochs with a batch size of 1024 rays and an initial learning rate of 10^{-4} . This takes 7 hours per object on an NVIDIA GeForce RTX 2080 Ti with 11GB RAM.

3.4.3.2 Baseline Comparison

For comparison against the baseline methods, we classified them into two categories: (a) Standalone methods: It either uses MVS or PS set up to recover the 3D shapes of an object. (b) Multi-Staged Fusion methods: It first recovers the sparse 3D point cloud of the object using multi-view images and surface orientation using fixed viewpoint PS images for each view. These spatial positions and orientations are then fused using a different method/pipeline to recover the 3D geometry. To compare against the standalone methods, we pick the 4th light source in DiLiGenT-MV setup. To recover the 3D shape using our method, we run our fine model by sampling points in 512^3 volumetric space uniformly. The recovered density is queried using marching-cube algorithm [172] with $\sigma = 10$. Table 3.2 compares the 3D reconstruction accuracy of our method to standalone and multi-stage fusion methods. We evaluate accuracy using the standard Chamfer-L1 distance metric between the recovered shape and the ground-truth shape after registration.

(a) Standalone Methods. For standalone baselines comparison, we compare our method with the recent state-of-the-art in MVS, PS, and View-Synthesis. (1) *PatchMatch Network* [140]: This method has shown state-of-the-art performance in MVS. PatchMatch-Net proposed an end-to-end trainable network that has fast inference time and works well even for high-resolution images. It is trained on the DTU dataset [173]. We empirically observed that using two source frames per reference view on the DiLiGenT-MV dataset results in more accurate depth estimation. After getting the depth map from the network for each view, we transform the results to a point cloud by back-projecting the depth values to 3D space. More details on the PatchMatch Network is provided in Section §3.5 of the thesis.

(2) *NeRF* [145]: Even though it is developed for novel view synthesis, recently NeRF has been widely used as a baseline for multi-view 3D reconstruction [142]. By sampling the volume density σ , it is possible to recover 3D geometry. We use an initial learning rate of 10^{-4} and batch size of 1024 rays. We run the optimization for 30 epochs and threshold the density values (512^3) at 10 to recover the 3D geometry.

Method Type →	Standalone Methods (↓)			Multi-Stage Fusion Methods (↓)		
Dataset	PM-Net [140]	NeRF [88]	Ours	R-MVPS [108]	B-MVPS [91]	Ours
BEAR	2.13	0.62	0.66	0.89	0.63	0.66
BUDDHA	0.72	0.99	1.00	0.64	0.40	1.00
COW	1.68	0.92	0.71	0.42	0.54	0.71
POT2	1.30	0.64	0.63	1.29	0.55	0.63
READING	1.64	1.22	0.82	0.98	0.85	0.82

TABLE 3.2: Quantitative 3D reconstruction accuracy comparison against the competing methods on DiLiGenT-MV benchmark. We used Chamfer- L_1 metric to compute the accuracy. The statistics show that our method is better and comparable to the stand-alone methods and Multi-Stage fusion methods respectively. Generally, Multi-Stage fusion methods are heuristic in nature and require careful execution at different stages. On the contrary, our method is much simpler to realize and implement.

Method	BEAR	BUDDHA	COW	POT2	READING
CNN-PS [14]	0.78	0.83	0.87	0.86	0.89
Ours	0.09	0.11	0.11	0.07	0.06

TABLE 3.3: Multi-view depth error comparison against CNN-PS [14]. For comparison, we integrate the surface normals from CNN-PS to recover its depth which is scaled appropriately to compute the l_1 depth accuracy. For ours, we projected the recovered shape reconstruction to the estimated depth and corresponding depth accuracy.

(3) *CNN-PS* [14]: This method proposes a dense convolution neural network to learn the mapping between PS images and surface normals directly. It can handle non-convex surfaces and complicated BRDFs. We integrate the obtained surface normals using Horn and Brooks’ method [94] to get the depth map, and scale it to $[-1, 1]$. Next, we also project our 3D shape to the different cameras and recover the depth on a similar scale for statistical comparison. Table 3.3 shows our performance comparison against CNN-PS.

(b) Multi-Stage Fusion Methods. Fusion approaches to MVPS usually comprise several steps that are heuristic in nature, and proper care must be taken to execute all the steps well. For evaluation, we compared against the two well-known baselines in MVPS (see Table 3.2).

(1) *Robust MVPS (R-MVPS)* [108]. It employs a series of different algorithms to solve MVPS. It first uses multi-view images to recover the coarse 3D mesh of the object using structure from motion. Next, this mesh is projected to 2D planar space for parameterization to estimate multi-view consistent

Dataset →	BEAR		BUDDHA		COW		POT ₂		READING	
Method	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓
NeRF [88]	29.97	0.0235	29.00	0.0455	30.80	0.0192	28.88	0.0269	28.12	0.0346
Ours	37.16	0.0122	33.59	0.0162	34.49	0.0134	30.47	0.0258	30.46	0.0311

TABLE 3.4: Quantitative image rendering quality comparison on DiLiGenT-MV benchmark. Our method can render much better images as it utilizes the surface normal information acquired with PS.

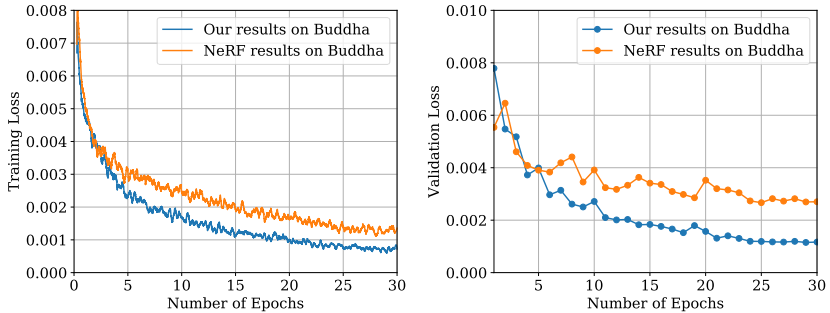


FIGURE 3.4: Training and validation loss curve on BUDDHA sequence.

surface normals using photometric stereo setup. Finally, the estimated surface normals are used for mesh refinement to recover the fine-detailed geometry of the object.

(2) *Benchmark MVPS (B-MVPS)* [91]. This method takes a set of steps to estimate a fine-detailed 3D reconstruction of an object from MVPS images. It first estimates iso-depth contours from the PS images [174] and then uses a structure-from-motion algorithm to recover a sparse point cloud from the MVS images. Later the depth of these 3D points is propagated along the iso-depth contour to recover the complete 3D shape. The spatially varying BRDF is computed once the 3D shape is recovered.

3.4.3.3 Analysis and Ablation

(a) Training and Validation Analysis. To demonstrate the effect of surface normal information on image rendering, we compare our method with NeRF. Fig.3.4 provides training and validation curves for both methods on BUDDHA. As expected, our method provides much higher image rendering quality during the learning process. Additionally, Table 3.4 provides PSNR

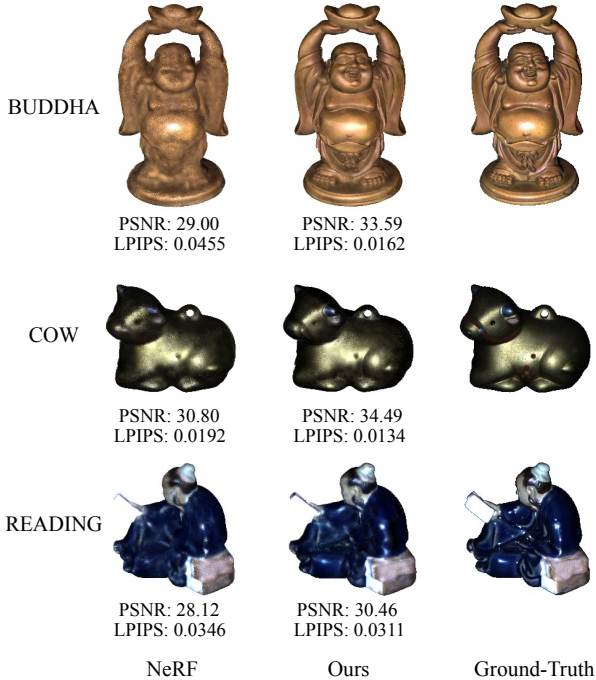


FIGURE 3.5: Visual comparison on DiLiGenT-MV renderings achieved by NeRF [88] and our method. Without surface normals, NeRF lacks in details and produces blurry renderings. On the other hand, our method is able to recover fine details and render accurate images by blending surface normal information in volume rendering process. PSNR (higher the better), LPIPS (lower the better).

and LPIPS [175] scores for NeRF and our method. These results show that our rendering quality is much better than standard view-synthesis approaches.

(b) Effect of Volume Sampling. Our method uniformly samples points along the ray between near and far bounds t_n, t_f . Increasing the number of these query points enables a denser evaluation of the network. Still, it is computationally not feasible to sample a lot of points uniformly. To make the process more efficient, we use a two-stage hierarchical volume sampling strategy by optimizing coarse and fine networks simultaneously. For that, we first consider the coarse network rendering:

Volume Sampling	BEAR	BUDDHA	COW	POT ₂	READING
$N_c = 64, N_f = 0$	0.85	1.45	0.86	0.63	1.38
$N_c = 256, N_f = 0$	0.68	0.92	1.01	0.64	1.64
$N_c = 64, N_f = 128$	0.66	1.00	0.71	0.63	0.82

TABLE 3.5: Reconstruction accuracy achieved with different number of points. We provide scores using Chamfer-L1 distance metric.

Method	BEAR	BUDDHA	COW	POT ₂	READING
Ours without view dependence	31.71	29.76	30.26	30.28	29.41
Ours with view dependence	37.16	33.59	34.49	30.47	30.46

TABLE 3.6: Quantitative image rendering quality measurement with PSNR metric with and without view dependence (The higher the better).

$$\tilde{C}_c(\mathbf{r}) = \sum_{i=1}^{N_c} \mathbf{w}_i(\mathbf{x}_i) \mathbf{c}_i(\mathbf{x}_i, \mathbf{n}_i^{ps}, \mathbf{v}), \text{ where } \mathbf{w}_i(\mathbf{x}_i) = T_i \left(1 - \exp(-\sigma(\mathbf{x}_i) \delta_i) \right) \quad (3.16)$$

We calculate weights $\hat{\mathbf{w}}_i(\mathbf{x}_i) = \mathbf{w}_i(\mathbf{x}_i) / \sum_{j=1}^{N_c} \mathbf{w}_j(\mathbf{x}_j)$ to have a probability density function on the ray. Then, we sample fine points from this distribution using inverse transform sampling. For the coarse network, we sample $N_c = 64$ points uniformly. For the fine network, we sample $N_f = 128$ points by taking the coarse network weights into account.

To show the effectiveness of our two-stage sampling strategy, we simulate an experiment. To that end, we remove fine sampling from our approach and evaluate the performance by using only the uniformly sampled points. Table 3.5 reports the 3D reconstruction accuracy using 64 and 256 coarse samples only, as well as the two-stage approach of using both coarse and fine sampling. The results suggest that choosing N_c and N_f introduces a better trade-off between computation time and accuracy.

(c) Effect of Viewing Direction. Here, we want to study the effect of viewing direction on rendered image quality obtained using our method for this experiment. To that end, we remove view direction information $\gamma(\mathbf{v})$ from our MLP. Table 3.6 compares the quality of image rendering with and without view dependence. As expected, the image quality sharply decreases without the view direction. So we conclude that similar to surface normals, view direction is also crucial for the rendering.

Dataset	BEAR	BUDDHA	COW	POT ₂	READING
NeRF [88]	0.80	1.07	0.78	0.81	1.18
Ours	0.70	1.06	0.79	0.73	0.98

TABLE 3.7: Quantitative 3D reconstruction accuracy against NeRF [88]. We tested both approaches with 20 different light configurations. We provide average scores using Chamfer-L1 metric.

(d) Limitations and Further Study. Our NR-MVPS approach combines two independent research fields that practice precise 3D reconstruction of an object from images. We demonstrated that the proposed method provides favorable results against the competing methods. However, we make a few assumptions, such as calibrated MVPS setup and solid objects, which limit our approach to broader adoption. Further, we do not explicitly model interreflections. Consequently, it would be interesting to extend our work to the uncalibrated settings. Exploring joint modeling of PS normal and surface normal from the implicit surface representation is not studied in this approach. In Section §3.5 and Section §3.6, we further investigate neural implicit representations for modeling of surface normals and better extraction of the object geometry.

3.4.4 Conclusion

We introduced a straightforward method that takes an opportunistic approach to solve the multi-view photometric stereo problem. We conclude that by exploiting the photometric stereo image formation model and the recent continuous volume rendering for multi-view image synthesis, we can reconstruct the 3D geometry of the object with high accuracy. Further, our formulation inherently utilizes the notion of the density gradient by leveraging the photometric stereo response and hence bypasses the explicit modeling of 3D occupancy information. We demonstrated that introducing knowledge about the density gradient to the neural radiance field representation provides encouraging improvements in MVPS. We assessed the suitability of our work via extensive experiments on the DiLiGenT-MV benchmark dataset.

3.5 UNCERTAINTY-AWARE DEEP MVPS

In this section, we propose an uncertainty-aware approach (UA-MVPS) that can effectively exploit such complementary surface information in multi-view photometric stereo setup. Our work leverages recent advances in deep neural networks. To this end, we use a PatchMatch-based deep-MVS network [176] to infer the per-pixel depth and a CNN-based deep-PS network [14] to infer per-pixel surface normal. But, we know that such deep network models have their accuracy limits and can predict erroneous depth or surface normals in certain parts of the object. In that case, if we naively combine the output predicted by these networks, we may end up having a bad overall result. To resolve this, we extend the deep-PS and deep-MVS networks with per-pixel uncertainty estimation capability. Using the prediction uncertainty as a measure, we select and combine only reliable surface estimates at each pixel, that is either deep-PS normal, deep-MVS depth, both, or none of the prediction.

Using our approach of selecting and discarding surface estimates may result in the loss of some pixels' corresponding 3D surface details. To recover those lost geometric details, we introduce a neural network (MLP layers only) based optimization to recover the overall dense shape from those selected surface predictions by representing the object's shape as level sets of the neural network. Our overall loss function optimization encourages the zero-level set of the neural network to converge to the confident surface estimates. To that end, we first convert the depth estimate to point cloud while keeping the predicted surface normals representation as it is. Our approach then optimizes for parameters of an MLP so that it approximates a signed-distance-function (SDF) to a plausible surface based on the point cloud, surface normals, and an implicit geometric regularization term developed on the Eikonal partial differential equation [177].

3.5.1 Contributions

In this section, we make the following contributions:

- We present an effective and easy-to-use deep neural network-based solution to the classical MVPS problem for dense, detailed, and precise recovery of 3D shapes.
- We introduce uncertainty-aware deep-PS and deep-MVS modeling in the MVPS pipeline. Modeling uncertainty helps as a measure in

automatic discarding of unreliable surface estimates at a pixel, hence improving robustness.

- We propose an implicit neural shape representation based on the Eikonal term in the MLP loss for natural zero level set surface recovery [177]. It reliably infers the object’s dense surface geometry defined by the confident deep-PS and deep-MVS prediction with better memory foot-print than the methods based on mesh processing [91, 108].

3.5.2 Method

As previously noted, $\mathcal{I}^v = \{I_1^v, \dots, I_{N_p}^v\}$ denotes the set of N_p input PS images for a given view $v \in \{1, \dots, N_m\}$, where N_m is the number of views. For MVS, we follow Li *et al.* [91] work, which take the median of all PS images per camera view to have MVS images. Concretely, $Y^v = \text{median}(\mathcal{I}^v)$ gives us the set $\mathcal{Y} = \{Y^1, \dots, Y^v, \dots, Y^{N_m}\}$ of MVS images. Since we consider a fully calibrated MVPS setup, we assume all light source directions, intensities and camera calibration matrices are provided.

The method section is organized as follows: First, we introduce our uncertainty-aware deep-PS network. Next, we describe the uncertainty-aware deep-MVS network pipeline. Finally, we explain our neural shape representation approach and level set optimization for dense, detailed, and accurate 3D surface recovery from high-fidelity surface normals and depth estimates.

3.5.2.1 Uncertainty-Aware Deep Photometric Stereo

We adhere to using a supervised deep learning framework to have high-fidelity surface normal predictions at test time for a diverse set of objects with different material properties. To that end, we adopt an observation map based modeling in deep-PS [14] due to its simplicity and notable performance on PS benchmark datasets [78, 163] as in NR-MVPS approach presented in Section §3.4. However, such a deep-PS network is not apt for estimating the uncertainty of the predicted surface normals at test time. For our problem, it is imperative to have that information as perfect prediction is not always possible. Accordingly, we modify the deep-PS architecture to provide uncertainty of the predicted surface normals by leveraging the Bayesian neural network (NN) approach [178, 179]. Generally, Bayesian NNs are a simple extension of NNs by placing a prior distribution

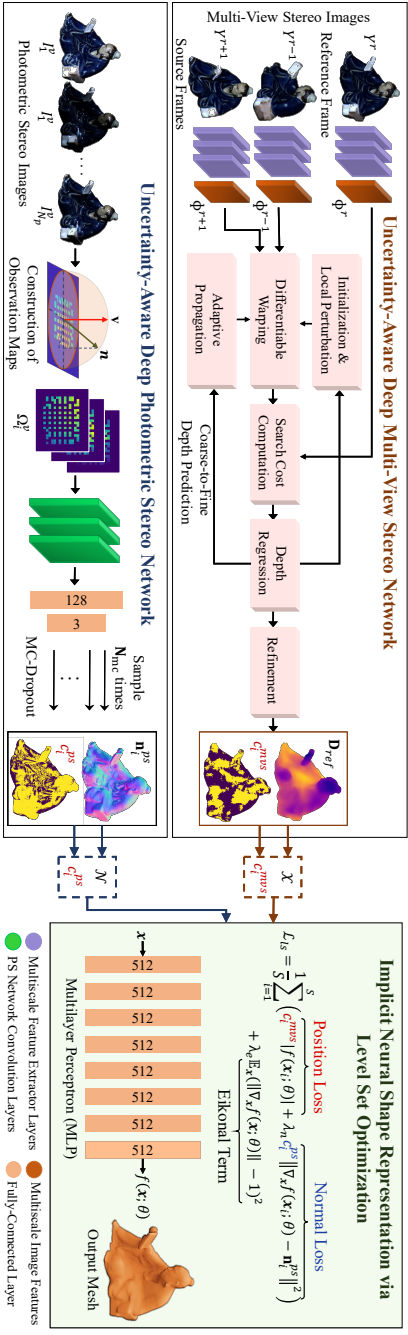


FIGURE 3.6: **UA-MVPS Overview:** We first predict per-pixel depth and surface normals via deep-MVS and deep-PS networks. Then, we recover dense, detailed 3D shape by using these surface estimates in neural level set optimization. Our approach utilizes prediction uncertainty as a measure of reliability and uses highly confident surface estimates in optimization for better surface recovery.

(generally Gaussian) over the NN's weights. Compared to standard NN, it has the advantage of providing an uncertainty measure of the network prediction [180, 181]. For completeness, let's briefly review the standard Bayesian NN framework.

Let $\{\mathbf{A}, \mathbf{B}\}$ be the training dataset with \mathbf{A} , \mathbf{B} as input and output sets, respectively. Assume, a Bayesian NN with L layers parameterized by weight $\mathbf{w} = \{\mathbf{W}_j\}_{j=1}^L$ with \mathbf{W}_j as the weight matrix for layer j . The predictive distribution $P(\mathbf{b}^* | \mathbf{a}^*, \mathbf{A}, \mathbf{B})$ for a new input \mathbf{a}^* is formulated as:

$$P(\mathbf{b}^* | \mathbf{a}^*, \mathbf{A}, \mathbf{B}) = \int P(\mathbf{b}^* | \mathbf{a}^*, \mathbf{w}) P(\mathbf{w} | \mathbf{A}, \mathbf{B}) d\mathbf{w} \quad (3.17)$$

However, to determine predictive distribution value is intractable as $P(\mathbf{w} | \mathbf{A}, \mathbf{B})$ is hard to solve analytically [182]. Generally, variational inference (VI) is used to approximate $P(\mathbf{w} | \mathbf{A}, \mathbf{B})$. By introducing the variational distribution $Q_\gamma(\mathbf{w})$ parameterized by γ , the KL divergence between $Q_\gamma(\mathbf{w})$ and $P(\mathbf{w} | \mathbf{A}, \mathbf{B})$ is minimized as:

$$\mathcal{L}_{VI} = - \int Q_\gamma(\mathbf{w}) \log(\mathbf{B} | \mathbf{A}, \mathbf{w}) d\mathbf{w} + KL(Q_\gamma(\mathbf{w}) || P(\mathbf{w})) \quad (3.18)$$

While VI based on KL divergence is widely used, the assumption of Gaussian distribution on the network parameters increases the complexity of the model. That, in turn, reduces the model efficiency without a significant gain in predictive power. Not long ago, Gal *et al.* [183] proposed a Bernoulli distribution-based VI approach which provides a simpler and fruitful approximation to the posterior distribution. To adopt Bernoulli distribution approach to our deep-PS network, we model $\mathbf{W}_j = \mathbf{M}_j \cdot \text{diag}([z_{j,u}]_{u=1}^{K_j})$ and $z_{j,u}$ as Bernoulli(q_j) for $j = 1, \dots, L$, $u = 1, \dots, K_{j-1}$. Here, \mathbf{M}_j denotes variational parameters of $\mathbf{W}_j \in \mathbb{R}^{K_j \times K_{j-1}}$ with K_j the number of units at layer j , and $z_{j,u}$ denotes the Bernoulli random variables with probability q_j .

With such parameterization, the integral term in \mathcal{L}_{VI} is approximated by sampling \mathbf{W} from a Bernoulli distribution, also known as Monte Carlo (MC) integration approach. Likewise, the KL divergence term in \mathcal{L}_{VI} is replaced with a weight decay on network parameters [183]. Using these approximations, we train our uncertainty-aware deep-PS network using the following loss function:

$$\mathcal{L}_{ps} = \frac{1}{N_{mc}} \sum_{j=1}^{N_{mc}} \|\tilde{\mathbf{n}}_j - \mathbf{n}_{gt}\|_2^2 + \lambda_w \sum_{j=1}^L \|\mathbf{W}_j\|_2^2 \quad (3.19)$$

where, N_{mc} is the number of MC samples, $\tilde{\mathbf{n}}_j$ is the estimated surface normal in each forward pass and \mathbf{n}_{gt} is the ground-truth surface normal.

As shown in Gal *et al.* [183] work, we can realize the approximation to Bernoulli distribution by introducing dropout layers in the neural network. Accordingly, we apply dropout with $q_j = 0.2$ after each convolution and fully-connected layer in deep-PS network. We keep dropout layers active during train and test time.

Due to the introduction of dropout layers, we now have a stochastic network. At test time, we run the trained model multiple times, recording the (potentially varying) surface normal prediction at each i^{th} pixel. We then calculate the mean and variance of these multiple predictions at every pixel. The mean is taken as the final prediction $\mathbf{n}_i^{ps} \in \mathbb{R}^{3 \times 1}$ and the variance as its uncertainty $\sigma_i^2 \in \mathbb{R}^{3 \times 1}$. Since our approach is focused on highly confident predictions, we convert the per-pixel variance to a single binary variable c_i^{ps} . We set $c_i^{ps} = 1$, if $\|\sigma_i^2\|_1 < \tau_{ps}$ and $c_i^{ps} = 0$, otherwise.

3.5.2.2 Uncertainty-Aware Deep Multi-View Stereo

Similar to deep-PS network, we aim to have a per-pixel uncertainty measure but now on the depth prediction. One natural way is to similarly use MC dropout strategy to deep-MVS network. Fortunately, there already exist deep MVS frameworks which have the intrinsic ability to implicitly provide uncertainty measure via confidence values of their depth predictions [130, 135, 176, 184]. Hence, it is inefficient to add extra complexity by introducing MC dropout layers. Among [130, 135, 176, 184], we use [176], i. e., PatchMatch based deep-MVS network due to its recent state-of-the-art performance on MVS benchmarks and fast inference on large scale images.

PatchMatch based Deep-MVS Network. Similar to PatchMatch algorithm [118], the PatchMatch based deep-MVS network employs [118] via similar three steps (but in 3d scene space) as follows: (i) Initialization step: Generating depth hypotheses, (ii) Propagation step: Propagate the hypotheses to neighbors, and (iii) Evaluation step: Compute the similarity cost and search for best solution. We apply these steps on per-pixel multi-scale features that are hierarchically extracted from MVS images \mathcal{Y} at M different resolution scales [176, 185]. This allows us to estimate depth in a coarse-to-fine manner. Before providing more details on these iterative steps, we reintroduce the notation for clarification. We denote the reference frame by $Y^r \in \mathbb{R}^{w \times h}$, coordinates of the i^{th} pixel by \mathbf{y}_i , frame r feature by Φ^r , and camera r intrinsic calibration matrix by \mathbf{K}_r . For each reference frame, we pick N_s source frames where $Y^s \in \mathbb{R}^{w \times h}$ denotes a source frame. $(\mathbf{R}_{r,s}, \mathbf{t}_{r,s})$ denotes the relative motion between frame r and s . We skip to add extra notation for stage number for simplicity of writing.

(i) *Initialization.* In the first iteration, we randomly sample per pixel \mathcal{D}_f depth hypotheses in the pre-defined inverse depth range $[d_{min}, d_{max}]$. Our sampling strategy ensures that the inverse depth range interval sampled into \mathcal{D}_f hypotheses is proper, and one hypothesis is covered at each interval. Once initialized, local perturbations are invoked in the subsequent iteration at each stage to diversify the hypotheses and make the method robust to front-to-parallel surface issues [111]. For local perturbation, per pixel, N_f^m hypotheses are generated at stage m in the normalized inverse depth range R_m .

(ii) *Propagation.* Let Φ^r denote the reference feature map, ε_j the fixed 2D offset for depth hypothesis j , and $\tilde{\varepsilon}_j(\mathbf{y}_i)$ the learnable 2D offset for pixel i at coordinates \mathbf{y}_i . A 2D CNN is applied on Φ^r to learn the 2D offset for each pixel. The depth hypotheses \mathbf{D}_p at pixel i is obtained as follows:

$$\mathbf{D}_p(\mathbf{y}_i) = \{\mathbf{D}(\mathbf{y}_i + \varepsilon_j + \tilde{\varepsilon}_j(\mathbf{y}_i))\}_{j=1}^{N_d^m} \quad (3.20)$$

where, N_d^m denotes the number of depth hypotheses at stage m and \mathbf{D} denotes the depth map in the last iteration. The learnable offset idea based on features allows to gather the hypotheses from the same surface rather than in the fixed set of neighbors, hence it is faster and more accurate.

(iii) *Evaluation.* Let $\Phi^r(\mathbf{y}_i)$, $\Phi^s(\mathbf{y}_i^{s,j}) \in \mathbb{R}^C$ be the reference feature and the warped source feature maps of pixel i and depth hypothesis d_j , respectively. Here, C is the number of feature channels. We get $\mathbf{y}_{i,j}$ via warping as follows:

$$\mathbf{y}_i^{s,j} = \mathbf{K}_s \left(\mathbf{R}_{r,s}(d_j(\mathbf{y}_i)) \cdot \mathbf{K}_r^{-1} \mathbf{y}_i + \mathbf{t}_{r,s} \right) \quad (3.21)$$

Next $\Phi^s(\mathbf{y}_i^{s,j})$ is obtained using differentiable bi-linear interpolation. To get the matching cost, we must sum per pixel cost from all the views and the depth hypotheses. For that, the cost per depth hypothesis is computed using group-wise correlation and aggregated over the number of views with per-pixel visibility weight [6, 139]. If G denotes the number of groups into which the feature maps are divided along channel dimension, then g^{th} group similarity $\Delta_s^g \in \mathbb{R}$ for source view s is given by:

$$\Delta_s^g(\mathbf{y}_i, j) = \Lambda \langle \Phi_g^r(\mathbf{y}_i), \Phi_g^s(\mathbf{y}_i^{s,j}) \rangle \quad (3.22)$$

Here, $\Lambda \in \mathbb{R}$ is the ratio of number of group to number of channels. Collecting the group similarity for all the pixels and over hypotheses gives $\Delta_s \in \mathbb{R}^{w \times h \times \mathcal{D} \times G}$. For vectorized usage, let $\Delta_s(\mathbf{y}_i, j) \in \mathbb{R}^G$ denote

the respective group similarity vector. To incorporate the visibility information per pixel $\mathbf{w}_s(\mathbf{y}_i)$ in the source image Y^s , a network composed of 3D convolutional layer with $1 \times 1 \times 1$ kernels and sigmoid activation is used. This simple pixel-wise network takes the initial set of group similarity Δ_s to provide the visibility weight measure $\mathcal{W}_s \in \mathbb{R}^{w \times h \times D}$ for a pixel in the range 0 to 1. Accordingly, the view weight is computed as $\mathbf{w}_s(\mathbf{y}_i) = \max(\{\mathcal{W}_s(\mathbf{y}_i, j)\}_{j=0}^{D-1})$. Using the visibility weight, the weighted group similarity $\tilde{\Delta}(\mathbf{y}_i, j)$ for pixel i and j^{th} depth hypothesis is computed as:

$$\tilde{\Delta}(\mathbf{y}_i, j) = \left(\sum_{s=1}^{N_s} \mathbf{w}_s(\mathbf{y}_i) \right)^{-1} \left(\sum_{s=1}^{N_s} \mathbf{w}_s(\mathbf{y}_i) \Delta_s(\mathbf{y}_i, j) \right) \quad (3.23)$$

The weighted group similarity over all the pixels and hypotheses is computed as $\tilde{\Delta} \in \mathbb{R}^{w \times h \times D \times G}$. To get the cost $\mathbf{J} \in \mathbb{R}^{w \times h \times D}$ per pixel and depth hypothesis, a 3D convolution network with $1 \times 1 \times 1$ kernel is applied on $\tilde{\Delta}$.

For aggregating the matching cost, an adaptive propagation strategy is followed. Similar to the propagation strategy per pixel, an additional spatial offset $\tilde{\mathbf{y}}_i^t$ per pixel i is learnt based on the AANet [111, 186]. For a spatial window with N_w pixels, the spatial cost aggregation is computed as

$$\tilde{\mathbf{J}}(\mathbf{y}_i, j) = \left(\sum_{t=1}^{N_w} w_t \cdot \tilde{d}_t \right)^{-1} \left(\sum_{t=1}^{N_w} w_t \cdot \tilde{d}_t \cdot \mathbf{J}(\mathbf{y}_i + \mathbf{y}_i^t + \tilde{\mathbf{y}}_i^t, j) \right) \quad (3.24)$$

\mathbf{y}_i^t is the pixel coordinates within the window. d_t and $w_t \forall t \in [1, N_w]$ are the weights per pixel based on the depth hypotheses and feature similarity, respectively. Feature weight at a sampled location is based on the feature similarity between corresponding features in Φ^r and \mathbf{y}_i , which is computed via group-wise correlation [187]. Whereas, the depth weights are based on the absolute difference in the inverse depth between the sampled location and \mathbf{y}_i using j^{th} hypotheses. To regress the depth per pixel, we apply softmax function to $\tilde{\mathbf{J}}(\mathbf{y}_i, j)$ which gives the confidence measures \mathcal{C} of the estimation.

$$\mathbf{D}(\mathbf{y}_i) = \sum_{j=0}^{D-1} d_j(\mathbf{y}_i) \cdot \text{softmax}(\tilde{\mathbf{J}}(\mathbf{y}_i, j)) \quad (3.25)$$

Subsequently, per-pixel confidence measure ρ_i is computed using the predicted probability of the most likely depth hypothesis, i.e. $\rho_i = \text{softmax}(\tilde{\mathbf{J}}(\mathbf{y}_i, j^*))$. Further, an independent depth residual network based

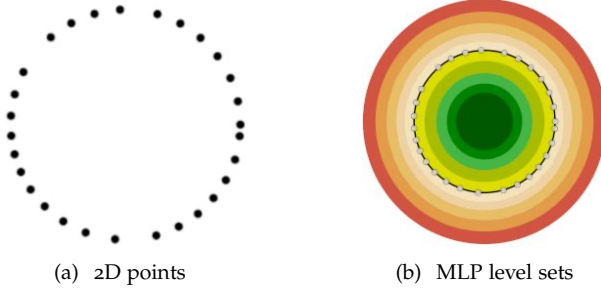


FIGURE 3.7: (a) Sparse 2D point cloud. (b) The Eikonal term based implicit geometric regularization in the optimization gives plausible zero level set for (a) —shown in black. The different colors in (b) show the level sets.

on Hui *et al.* work [188] is used to obtain the refined depth map \mathbf{D}_{ref} . It extracts the features Φ^D from \mathbf{D} , the Φ^I from Y^r , and upscale Φ^D to image size via deconvolution. Both of these features are concatenated and subsequently multiple 2D convolution layers are used to compute the depth residual. For more details on the PatchMatch based deep-MVS network, refer [176].

Deep-MVS Loss Function. We use l_1 loss between the estimated depth and the ground-truth depth at the same resolution. The total MVS loss takes into account the PatchMatch loss \mathcal{L}_{pm} at each stage along with refined depth loss \mathcal{L}_{ref} .

$$\mathcal{L}_{mvs} = \mathcal{L}_{pm} + \mathcal{L}_{ref}, \text{ where } \mathcal{L}_{pm} = \sum_{m=1}^M \sum_{t=1}^{N_{iter}^m} \mathcal{L}_t^m \quad (3.26)$$

Here, N_{iter}^m denotes the total number of iterations at stage m .

Uncertainty Modeling in Deep-MVS Network. To have the notion of per-pixel depth uncertainty, we convert depth prediction confidence value (ρ_i 's) to a binary variable c_i^{mvs} . We set $c_i^{mvs} = 1$ if $\rho_i > \tau_{mvs}$ and $c_i^{mvs} = 0$, otherwise.

3.5.2.3 Implicit Neural Shape Representation

Once we have the complementary information of the object shape, i. e., surface normals, depth, and the fidelity of prediction at hand, our goal is

to combine them effectively for dense surface reconstruction. It is pretty natural to go for regular volumetric fusion approaches [189–192] since surface normals can also provide depth by simple integration [193]. However, we know volumetric fusion uses a fixed size cubic grid independent of the object’s geometry, and therefore, may not obey the geometry of the shapes we want to model. Consequently, we propose to work directly on the confident raw surface estimates. For that, we convert deep-MVS network per-pixel depth prediction to point cloud $\mathcal{X} = \{\mathbf{p}_i\}_{i=1}^S \subset \mathbb{R}^3$, where S is the total number object pixel across all views. Additionally, we use per-pixel surface normal prediction from deep-PS network $\mathcal{N} = \{\mathbf{n}_i^{ps}\}_{i=1}^S \subset \mathbb{R}^3$. We propose to recover the 3D surface by optimizing the parameters of a MLP $f_\theta(\mathbf{x})$. The suggested MLP approximates the signed distance function (SDF) defined by \mathcal{X} and \mathcal{N} . We consider the following loss function for MLP optimization:

$$\begin{aligned} \mathcal{L}_{ls} = & \frac{1}{S} \sum_{i=1}^S \left(\underbrace{c_i^{mvs} |f_\theta(\mathbf{p}_i)|}_{\text{position loss}} + \lambda_n \underbrace{c_i^{ps} \|\nabla_{\mathbf{x}} f_\theta(\mathbf{p}_i) - \mathbf{n}_i^{ps}\|}_{\text{normal loss}} \right) \\ & + \lambda_e \underbrace{\mathbb{E}_{\mathbf{x}} (\|\nabla_{\mathbf{x}} f_\theta(\mathbf{x})\| - 1)^2}_{\text{Eikonal term}} \end{aligned} \quad (3.27)$$

The first term encourages the zero level set to converge to high-fidelity position estimates (i. e., for $c_i^{mvs} = 1$). The second term forces the local gradients to be consistent with the reliable normal estimates (i. e., for $c_i^{ps} = 1$). The final term stands for the Eikonal regularization, and it is computed by taking the expectation \mathbb{E} over probability distribution $\mathbf{x} \sim \mathcal{P}$. It is noted over several experiments that the Eikonal term is impressive at implicitly regularizing the zero level set by favoring smooth and plausible surfaces. For more details, refer [177, 194].

3.5.3 Experiments and Results

Train set. We use CyclesPS synthetic dataset [14] to train our *deep-PS network*. It consists of 15 shapes each of which is rendered with diffuse, specular, and metallic BRDF’s using 1300 light sources. We use 90% of the data for training and 10% for validation. For training the *deep-MVS network*, we use DTU MVS dataset [173]. It provides images of 80 scenes captured from 49 or 64 views (depending on the subject) with their ground-truth (GT) depth maps. We keep the training and validation splits same as outlined in [113].

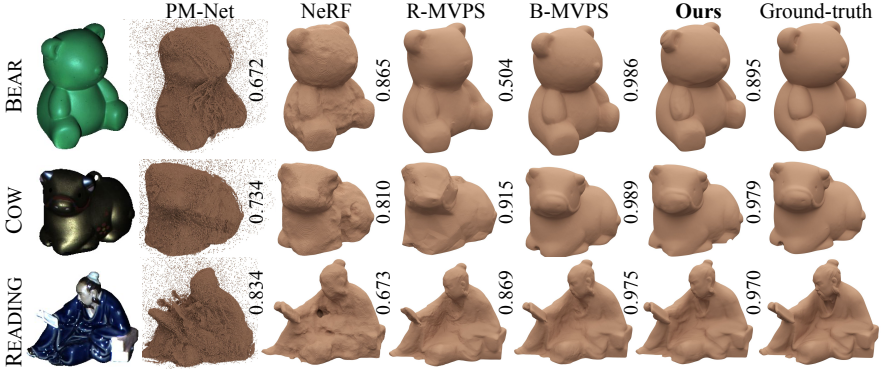


FIGURE 3.8: Comparison of the reconstruction quality with PM-Net [176], NeRF [88], R-MVPS [108], and B-MVPS [91] using \mathcal{F} -score metric.

Test set. We used DiLiGenT-MV benchmark dataset [91] as the test set to perform all our experiments, statistical evaluations, and ablations.

3.5.3.1 Implementation Details

We implemented our approach in Python 3.8 using PyTorch 1.7.1 library. We conducted all the experiments on a commodity desktop supported with NVIDIA GPU with 11GB of RAM. We trained deep-PS and deep MVS networks independently in a supervised setting.

(i) *Uncertainty-Aware Deep Photometric Stereo.* We first generate pixel-wise observation maps $\Omega_i^v \in \mathbb{R}^{32 \times 32}$ using CyclesPS images to train our deep-PS network. For each observation map, we randomly pick between 50 to 1300 light sources. We train the network for 10 epochs using Adam optimizer [156] with a learning rate of 0.1. During training, we set $N_{mc} = 10$ and $\lambda_w = 10^{-4}$ in our loss function (see Eq.(3.19)). After training, we perform uncertainty based inference on DiLiGenT-MV images. For that, we first generate observation maps for each pixel. Then, we run the deep-PS network on each observation map 100 times following MC-Dropout approach [183]. We calculate, for each pixel, the mean and variance of the outputs to obtain surface normal \mathbf{n}_i^{ps} and its uncertainty σ_i^2 . We set $\tau_{ps} = 0.03$ to obtain c_i^{ps} ($c_i^{ps} = 1$ if $\|\sigma_i^2\|_1 < \tau_{ps}$ and $c_i^{ps} = 0$, otherwise).

(ii) *Uncertainty-Aware Deep Multi-View Stereo.* The deep-MVS network is trained using DTU MVS dataset [173]. It is trained for 8 epochs using Adam optimizer [156] and learning rate of 0.001. To predict depth with

Method Type →		Deep Multi-View Stereo		View-Synthesis		Photometric Stereo			
Dataset ↓	Method →	MVSNet [135]	PM-Net [176]	NeRF [88]	IDR [152]	Robust PS [20]	SDPS-Net [54]	CNN-PS [14]	Ours
	BEAR	0.135	0.672	0.865	0.053	0.266	0.239	0.293	0.895
	BUDDHA	0.147	0.799	0.713	0.150	0.367	0.298	0.363	0.922
	COW	0.095	0.734	0.810	0.098	0.245	0.447	0.511	0.979
	POT2	0.126	0.666	0.859	0.079	0.231	0.464	0.632	0.907
	READING	0.115	0.834	0.673	0.073	0.242	0.188	0.508	0.970
	AVERAGE	0.124	0.741	0.784	0.091	0.270	0.327	0.461	0.935

TABLE 3.8: \mathcal{F} -score comparison with standalone methods on DiLiGenT-MV dataset [91]. The statistics show that our method achieves better results compared to the standalone multi-view and PS methods, thereby showing the advantage of using the complementary surface information in a MVPS setup.

coarse-to-fine approach, we set number of stages $M = 3$ for performing PatchMatch at different scales with $N_{iter}^3 = 2$, $N_{iter}^2 = 2$, $N_{iter}^1 = 1$ (higher M indicates coarser scale). $\mathcal{D}_f = 48$ depth hypotheses are used at initialization. For local perturbation $N_l^3 = 16$, $N_l^2 = 8$, $N_l^1 = 8$, and $N_d^3 = 16$, $N_d^2 = 8$, $N_d^1 = 0$ at propagation steps. At last, depth refinement is performed at the original image resolution [176]. For testing, we use DiLiGenT-MV images with $N_s = 2$ source images per reference image. We predict per-pixel depth and confidence measure ρ_i for all views using the above parameters. We set $\tau_{mvs} = 0.9$ to obtain c_i^{mvs} , where $c_i^{mvs} = 1$ if $\rho_i > \tau_{mvs}$ and $c_i^{mvs} = 0$ otherwise.

(iii) *Overall Loss Optimization.* We optimize for a zero-level set surface defined by highly confident estimates in \mathcal{X} and \mathcal{N} (see Section §3.5.2.3). For that, we first perform a precautionary multi-view consistency check to eliminate a few spurious 3D points. MLP with 8 layers is then used on remaining highly-confident estimates to learn suitable implicit neural shape representation. Here, each MLP layer contains 512 hidden units. A skip connection combines the input to the 4th layer to speed up learning [194]. We set the parameters $\lambda_n = 10$ and $\lambda_e = 1$ for our loss function (Eq.(3.27)). The distribution \mathcal{P} for the expectation in Eq.(3.27) is taken as average of (a) sum of Gaussian centered at points of \mathcal{X} locally and (b) a uniform distribution globally. The standard deviation of Gaussian at a point is taken as the distance to the 50th nearest neighbor. For optimization, we used Adam optimizer [156] with a learning rate of 0.001. We train the MLP for 10^5 epochs by sampling 2^{14} elements from \mathcal{X}, \mathcal{N} in each batch. We run the trained MLP on a volumetric grid of size 512^3 , which is then used by marching cubes algorithm [172] to extract the mesh corresponding to the zero level set of MLP based neural shape representation.

Dataset↓ Method →	NR-MVPS [195]	R-MVPS [108]	B-MVPS [91]	Ours	Difference with [91]
BEAR	0.856	0.504	0.986	0.895	0.091
BUDDHA	0.690	0.935	0.934	0.922	0.012
COW	0.844	0.915	0.989	0.979	0.010
POT2	0.858	0.458	0.984	0.907	0.077
READING	0.720	0.869	0.975	0.970	0.005
AVERAGE	0.794	0.736	0.974	0.935	0.039

TABLE 3.9: \mathcal{F} -score comparison with different MVPS methods on DiLiGenT-MV dataset [91]. The statistics show that our method performs far better than R-MVPS and compares favorably with the B-MVPS [91]. The point to note is that we can get results close to the state-of-the-art with a simple and easy-to-implement method.

Method Type →		TSDF Fusion [189]		Ours	
Dataset↓ Metric →	\mathcal{F} -score (↑)	Chamfer- L_1 (↓)	\mathcal{F} -score (↑)	Chamfer- L_1 (↓)	
BEAR	0.129	4.624	0.895	0.415	
BUDDHA	0.398	2.069	0.922	0.455	
COW	0.192	3.392	0.979	0.329	
POT2	0.056	6.100	0.907	0.515	
READING	0.314	2.238	0.970	0.355	
AVERAGE	0.218	3.685	0.935	0.414	

TABLE 3.10: Comparison of the reconstruction quality with TSDF Fusion [189], which is a standard method of choice for robust 3D fusion (outlier removal). We use \mathcal{F} -score (higher is better) and Chamfer- L_1 (lower is better) metrics for statistical evaluation.

3.5.3.2 Baseline Comparisons

Here, we present our baseline comparison results, which we have divided into three sub-categories.

(a) With standalone methods. Here, we compare our performance with methods that use either PS or MVS setup. Such an experiment helps us understand the benefit of using PS and MVS information together and how accurately we can reconstruct the shape with standalone methods. We used \mathcal{F} -score to compare our method’s performance with state-of-the-art PS, MVS, and view-synthesis methods. Table(3.8) provides the statistics for the same, indicating the clear advantage of our approach against the standalone methods.

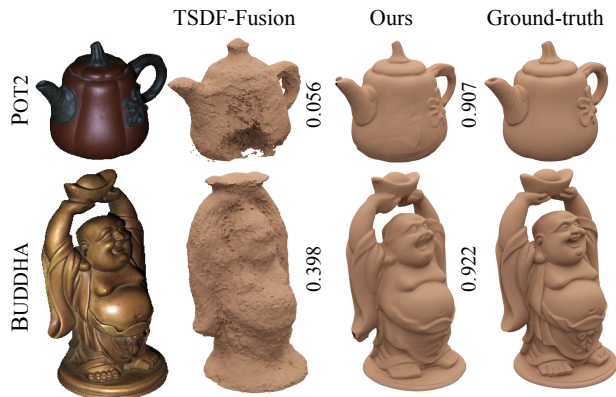


FIGURE 3.9: Comparison of the reconstruction quality with TSDF Fusion [189], which is a standard method of choice for robust 3D fusion (outlier removal).

(b) With MVPS methods. We compare our method with well-known MVPS methods, i. e., R-MVPS [108], B-MVPS [91], and NR-MVPS [195]. Table(3.9) provides \mathcal{F} -score comparison with these methods. Our method performs better than R-MVPS [108], NR-MVPS [195] and compares favorably with B-MVPS [91] with the minor difference in \mathcal{F} -score values i. e., less than 10^{-1} in all object categories. Additionally, we want to emphasize again that B-MVPS [91] relies on several carefully crafted explicit geometric steps and refinements that are complex and time-consuming, while our deep MVPS approach is easy to implement and realize.

(c) With standard volumetric fusion method. For estimating the dense 3D surface, it is possible to combine the deep-MVS depth map and depth from PS surface normal map via widely used robust 3D fusion technique, i. e., TSDF fusion [189]. To show that our approach is better at estimating the object’s surface than TSDF fusion, we executed the TSDF fusion algorithm on our deep-MVS depth and depth from deep-PS output. The qualitative comparison is presented in Fig.3.9. Table(3.10) provides \mathcal{F} -score and Chamfer- L_1 metric stats for the rest of the DiLiGenT-MV subjects. The results clearly show that TSDF fusion provides undesirable output. On the other hand, our approach takes proper care of the surface estimates and provides much better 3D surface reconstruction.

Settings↓ Dataset →	BEAR	BUDDHA	COW	POT ₂	READING	AVG
w/o Uncertainty Modeling	0.468	0.485	0.365	0.557	0.380	0.451
w/o PS Uncertainty Modeling	0.443	0.481	0.381	0.484	0.377	0.433
w/o MVS Uncertainty Modeling	0.457	0.473	0.339	0.636	1.024	0.586
Ours (LCNet [54] light)	0.481	0.465	0.346	0.481	0.381	0.431
Ours	0.415	0.455	0.329	0.515	0.355	0.414

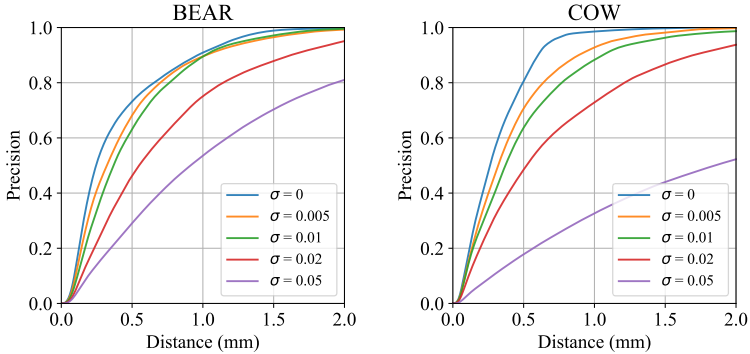
TABLE 3.11: Effect of uncertainty modeling and light calibration on the reconstruction quality of our approach. The results show Chamfer- L_1 metric (lower is better). The numbers confirm that utilizing the estimated uncertainty of both PS and MVS produces the best results. Further, our approach performs well without the exact light sources i. e., LCNet light sources [54].

3.5.3.3 Ablation Study and Further Analysis

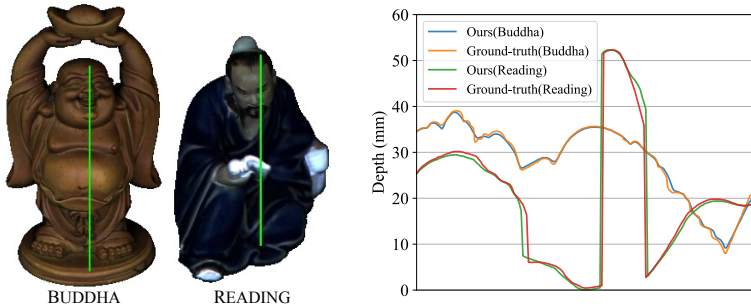
(a) Effect of uncertainty modeling. To understand the impact of uncertainty modeling on our method’s performance, we analyzed our results under three distinct settings of overall loss function i. e., Eq.(3.27): (i) We remove both deep-MVS and deep-PS confidence variable, i. e., c_i^{mvs} , c_i^{ps} from Eq.(3.27) (w/o uncertainty modeling), (ii) We keep the deep-MVS confidence variable (c_i^{mvs}) and drop the deep-PS confidence variable (c_i^{ps}) from Eq.(3.27) (w/o PS uncertainty modeling), and (iii) We keep the deep-PS confidence variable (c_i^{ps}) and drop the deep-MVS confidence variable (c_i^{mvs}) from Eq.(3.27) (w/o MVS uncertainty modeling). Table(3.11) shows the Chamfer- L_1 metric results obtained under the three settings. The results indicate that incorporating uncertainty information to the loss helps handle the erroneous estimations, and therefore, results in better 3D reconstruction.

(b) Effect of light sources. Here, we study the behavior of our approach under the uncalibrated-PS setting where GT light sources are not given. Instead, we used a pre-trained neural network to have an initial estimate of light sources. Precisely, we used LCNet [54] model to have light source direction and intensity values. Table(3.11) shows Chamfer- L_1 metric obtained using LCNet predicted light sources information. Our method performs almost equivalently well, showing robustness to small errors in the light calibration.

(c) Effect of noise. We consider the MVPS acquisition setup in our work, where imaging noise is inevitable. So, we analyze the behavior of our approach under imaging noise. To that end, we add zero-mean Gaussian noise



(a) Precision w.r.t. noise levels



(b) Recovered surface profile

FIGURE 3.10: (a) Surface reconstruction accuracy for Bear and Cow objects under different noise levels. We report precision as a function of distance threshold (τ) to show the fraction of accurately reconstructed points. (b) Estimated surface profile using our approach, showing how the recovered 3D shape follows the ground-truth surface profile curve when compared across the arbitrarily chosen geodesics.

to images with different standard deviations. Fig.3.10(a) shows the precision curve of the recovered surface as a function of the distance threshold τ . Precisely, it measures the fraction of points that are reconstructed accurately. The plots show that increasing the noise level degrades the performance. Further, we infer that behavior among subjects varies as signal-to-noise ratio changes. We can observe that our method is robust, and the performance drop is not random.

(d) Quality of reconstructed surface geometry. To perform this experiment,

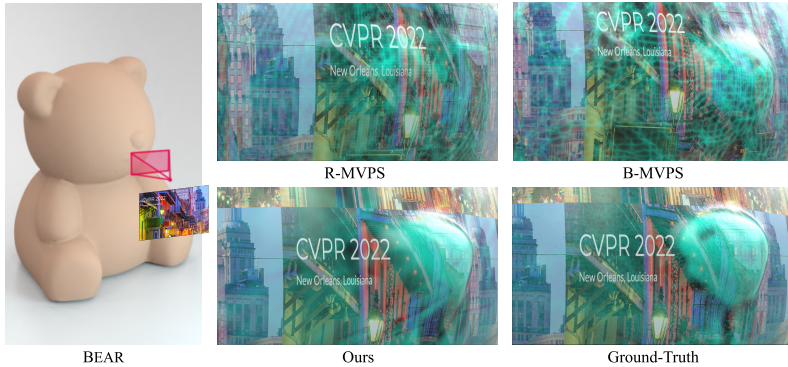


FIGURE 3.11: We transfer the CVPR’22 logo texture on the local region (around the nose) of the mesh recovered using MVPS methods. It can be observed that the texture pattern on our recovered mesh is closer to GT compared to B-MVPS [91] and R-MVPS [108] (notice the shift of the text). We want to emphasize that if the local topology is same, it must place the text at similar location as can be seen in ours result and GT.

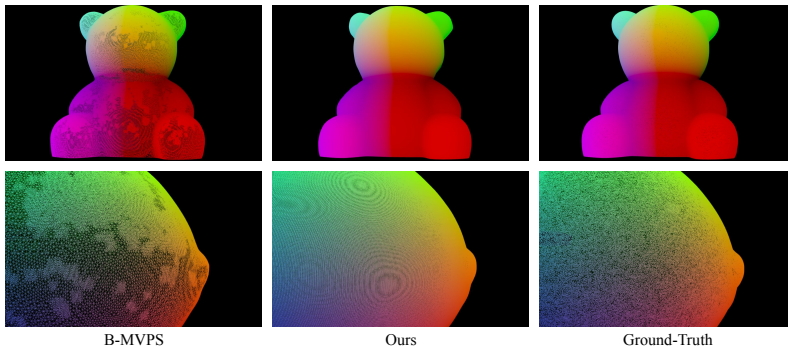


FIGURE 3.12: Colored Wireframe comparison with state-of-the-art B-MVPS [91]. Clearly, the distribution of geometric primitives on B-MVPS result is uneven and unbalanced compared to ours.

we first analyzed the surface profile of our reconstructed shape across randomly chosen geodesics on the object. Fig.3.10(b) shows that our recovered surface profile closely follows GT. Next, we performed a local surface analysis of recovered mesh. Although evaluation of recovered shape based on global performance metric is already discussed, it may not reflect the

true picture of surface topology since the actual distribution of mesh on GT shape is not known a priori. So, as a second experiment, we transferred texture on a local mesh topology and qualitatively compared the results. Fig.3.11 shows texture transfer results on the mesh recovered using different methods, and clearly, our textured mesh reflects fine text details and appears close to GT. Further, we analyzed the colored Wireframe of the recovered shape compared to B-MVPS [91]. Fig.3.12 Wireframe model shows that the distribution of geometric primitives in our recovered shape is smooth, regular, and close to GT, whereas B-MVPS [91] has an unbalanced distribution of geometric primitives.

Limitations. This approach assumed light sources and camera calibration information are given as input. Further, we assumed that the surfaces exhibit isotropic BRDF properties and tested our method on [91] dataset, which is generally composed of isotropic material objects. In Section §3.6, we show that our UA-MVPS method may fail on anisotropic surface reflectances and propose a volume rendering based approach to overcome this limitation.

3.5.4 Conclusion

This section explores the field of MVPS using the concepts from deep learning, geometry, and uncertainty. Unlike existing MVPS methods, which treat 3D shape reconstruction as point estimation and geometric optimization problems, we propose learning the fidelity of surface estimation and recovering the shape based on the implicit neural shape representation. Without using complex geometric steps, we observed that our simple neural network based approach could provide results comparable to the best available algorithm. Thus, we believe our work will enable broader use of the MVPS approach in precise 3D data acquisition.

3.6 UNCERTAINTY-AWARE NEURAL VOLUME RENDERING

In this section, we extend the UA-MVPS method presented in Section §3.5 where we introduced uncertainty modeling in multi-view stereo and photometric stereo neural networks for reliable inference of the 3D position and surface normals, respectively. Although uncertainty estimation helps us filter wrong predictions, it can lead to incomplete recovery of an object’s 3D shape. To this end, we introduce neural volume rendering of the implicit 3D shape representation. It has couple of key advantages over the pipeline presented in Section §3.5: (i) It helps extending the application of MVPS to a

wider class of object with different material type (see Fig.3.13). *(ii)* It further enhances the performance and use of implicit neural shape representation in MVPS leading to state-of-the-art results on benchmark datasets.

Recent multi-view stereo approaches have shown that neural volume rendering using the implicit neural 3D shape representation can effectively model a diverse set of objects via multi-view image rendering techniques [88, 143, 152]. Therefore, introducing it to MVPS can assist in handling challenging objects' material types. Intuitively, rendering-based geometry modeling can succeed where both the MVS and PS methods fail to estimate the surface geometry [57, 91, 108]. Further, contrary to the standard practice in MVPS of performing optimization or filtering on explicit geometric primitives [57, 91, 108], i. e., mesh, neural volume rendering relies on neural implicit shape representation, which is memory efficient and is scalable [143].

3.6.1 Contributions

This section makes the following contributions:

- We present a simple, efficient, scalable, and effective MVPS method for the detailed and complete recovery of the object's 3D shape.
- Our proposed uncertainty-aware neural volume rendering uses confident priors from deep-MVS and deep-PS networks and encapsulates them with an implicit geometric regularizer to solve MVPS demonstrating state-of-the-art reconstruction results on the benchmark dataset [91].
- Contrary to the current state-of-the-art methods, our method applies to a broader class of object material types, including anisotropic and glossy materials. Hence, widen the use of MVPS for 3D data acquisition.

3.6.2 Method

On the one hand, we have the state-of-the-art geometric methods that are composed of several complex steps, hence not suitable for automation. Further, they cannot meet the modern demand of scalability, and thus, less convincing for the current challenge of handling a large set of object data. On the other hand, our method proposed on Section §3.5 (UA-MVPS) is simple and scalable but works well only for isotropic material objects.

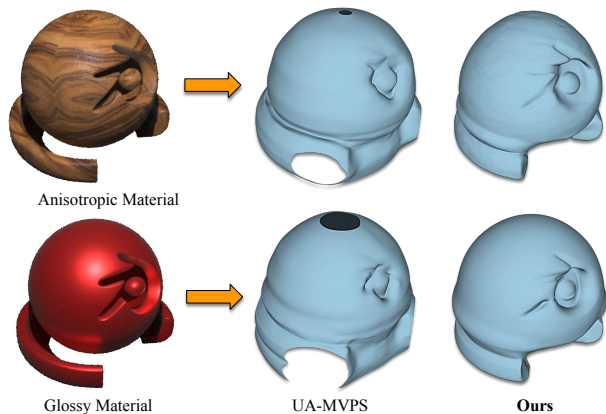


FIGURE 3.13: The advantage of uncertainty-aware volume rendering approach over UA-MVPS [58]. It can be observed that our method is able to correctly recover the fine object’s details for anisotropic and glossy material object. The 3D model used for the above illustration is taken from [88] dataset.

In this section, we introduce a simple, scalable, and effective approach that can handle a much broader range of objects. We first recover the 3D position and surface normal priors from MVS and PS images (MVPS setup) using uncertainty-aware deep multi-view stereo [176] and deep photometric stereo networks [14, 58], respectively. The uncertainty-aware network measures the suitability of the predicted surface measurements for its reliable fusion. However, the filtering of unreliable predictions based on the uncertainty measures leads to the loss of local surface geometry. Thus, we introduce a geometric regularization term in the overall loss function to recover the complete 3D geometry of the object. To that end, we represent the object’s shape as level sets of a neural network and recover it by optimizing the parameters of a multi-layer perceptron (MLP). The MLP approximates a signed-distance-function (SDF) to a plausible surface based on the point cloud, surface normals, and an implicit geometric regularization term developed on the Eikonal partial differential equation [177].

The above pipeline generally works well but cannot model anisotropic or glossy surfaces. Hence, not a general solution and is unsuitable for large applications. On a different note, we observed that representing the light fields and density of the object as a neural network in a multi-view volume rendering algorithm improves the 3D reconstruction of general

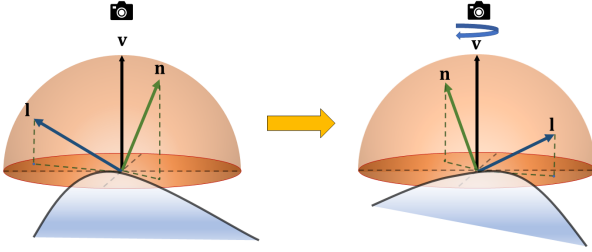


FIGURE 3.14: Under the isotropy assumption, the measured pixel intensity is invariant to the joint rotation of surface normal \mathbf{n} and light source direction \mathbf{l} around the viewing direction \mathbf{v} . Such an assumption though suited for a reasonable category of objects but is not applicable for anisotropic material objects.

objects. Further, as well-studied, volume rendering generalizes well to diverse objects with different material types. Such an observation lead us to introduce an uncertainty-aware volume rendering approach to the MVPS problem. As we will show, it not only helps achieve state-of-the-art results on isotropic material objects but also provide accurate 3D surface reconstruction on challenging subjects such as glossy texture-less surface objects. Next, we describe each component of our approach in detail, leading to the final loss.

3.6.2.1 Shape Representation and Regularization

Using deep-MVS and deep-PS networks —as described in previous section, we filter confident 3D positions and surface normals $\{\mathbf{p}_i, \mathbf{n}_i^{ps}\}_{i=1}^S \subset \mathbb{R}^3$ prediction $\forall i \in [1, \dots, S]$. Our goal is to recover object’s dense 3D reconstruction combining those reliable intermediate priors. To this end, we propose to learn the signed distance function (SDF) of the object surface defined by a implicit function $f_\theta(\mathbf{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}$ using the reliable prediction estimates. We model the function using an MLP parameterized by θ , assuming its zero level set approximates the object surface.

To find the optimal θ , we consider the Eikonal equation ($\|\nabla_{\mathbf{x}} f_\theta(\mathbf{x})\| = 1$). It establishes a constraint on $f_\theta(\mathbf{x})$ to represent a true SDF. Note that even if the boundary conditions imposed by the given surface estimates are satisfied (i. e., $f_\theta(\mathbf{p}_i) = 0$, $\nabla_{\mathbf{x}} f_\theta(\mathbf{p}_i) = \mathbf{n}_i^{ps}$), a unique solution to the zero level set surface may not exist. Nevertheless, describing an incomplete set of surface 3D estimates using Eikonal condition as a regularizer favors

smooth and plausible surfaces [194]. Hence, we consider the following regularization term in our optimization:

$$\mathcal{L}_{\text{Eikonal}} = \lambda_e \mathbb{E}_{\mathbf{x}} (\|\nabla_{\mathbf{x}} f_{\theta}(\mathbf{x})\| - 1)^2 \quad (3.28)$$

where the expectation is computed w.r.t. a probability distribution $\mathbf{x} \sim \mathcal{P}$. Note that we have considered the Eikonal regularization to interpolate the surface from MVS and PS network predictions in Section §3.5. However, the question we ask in this section, *did we utilize all the imaging prior provided by MVPS well or can we do better?* In this approach, we show that by cleverly using multi-view image prior, we can perform better than UA-MVPS [58]. To accomplish that, we introduce neural volume rendering method to MVPS.

3.6.2.2 Neural Volume Rendering

Recent work on volume rendering techniques has shown outstanding results in learning scene representations from multi-view images [88]. Although such techniques are impressive with novel view synthesis, they can not faithfully provide the object’s geometry from the learned volume density, leading to inaccurate and noisy reconstructions. Therefore, for our work, we use SDF-based volume rendering approach [143] which models volume density as a function of the signed distance value as follows:

$$\begin{aligned} \sigma(\mathbf{x}) &= \alpha \Phi_{\beta}(-f_{\theta}(\mathbf{x})), \\ \text{where } \Phi_{\beta}(s) &= \begin{cases} \frac{1}{2} \exp\left(\frac{s}{\beta}\right), & \text{if } s \leq 0 \\ 1 - \frac{1}{2} \exp\left(-\frac{s}{\beta}\right), & \text{if } s > 0 \end{cases} \end{aligned} \quad (3.29)$$

Here, $\alpha, \beta > 0$ are trainable parameters and $\Phi_{\beta}(\cdot)$ is the cumulative distribution function of a zero-mean Laplace distribution. Eq:(3.29) ensures a smooth transition of density values near the object boundary, and at the same time allows a suitable extraction of zero level set after optimization for surface recovery. Inspired by the classical volume rendering techniques [170, 171], the expected color $I(\mathbf{o}_i, \mathbf{v}_i)$ of a camera ray $\mathbf{x}_i(t) = \mathbf{o}_i + t\mathbf{v}_i$ with camera center $\mathbf{o}_i \in \mathbb{R}^3$ and viewing direction vector $\mathbf{v}_i \in \mathbb{R}^3$ can be modeled as:

$$I(\mathbf{o}_i, \mathbf{v}_i) = \int_{t_n}^{t_f} T(\mathbf{x}_i(t)) \sigma(\mathbf{x}_i(t)) \mathbf{r}_{\psi}(\mathbf{x}_i(t), \mathbf{n}_i(t), \mathbf{v}_i) dt, \quad (3.30)$$

where $T(\mathbf{x}_i(t)) = \exp\left(-\int_0^t \sigma(\mathbf{x}_i(s)) ds\right)$ is the transparency, $\mathbf{n}_i(t) = \nabla_{\mathbf{x}} f_{\theta}(\mathbf{x}_i(t))$ is the level set’s normal at $\mathbf{x}_i(t)$, \mathbf{r}_{ψ} is the radiance field function

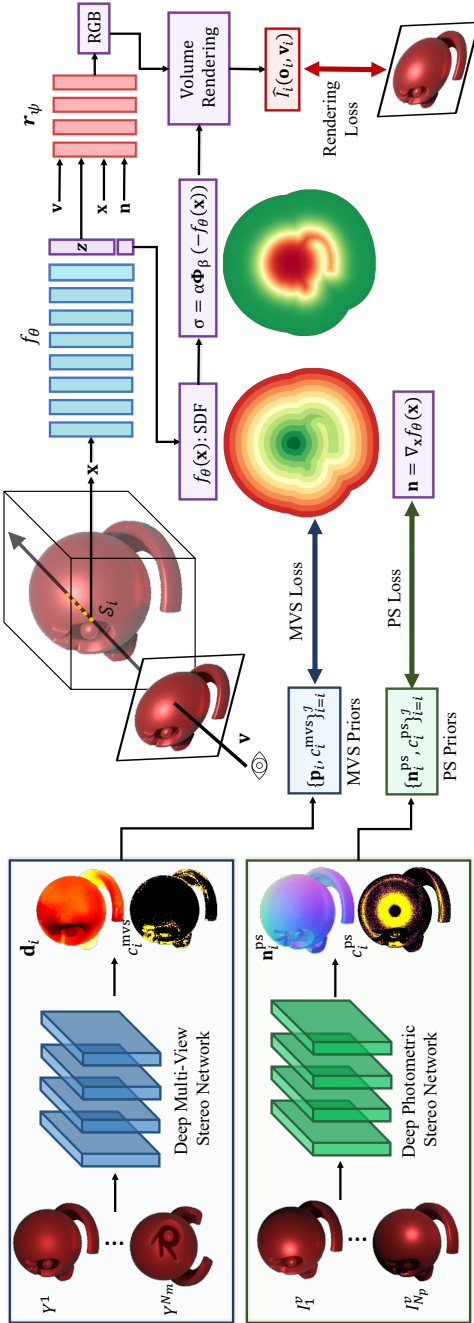


FIGURE 3.15: **Method overview (Left to Right):** We obtain highly confident 3D position and surface normal predictions of the object via uncertainty-aware deep-MVS and deep-PS networks, respectively. Then, we learn the signed distance function representation of the object surface. Finally, our optimization uses the volume rendering technique to recover the missing details of the surface, providing high-quality 3D reconstructions of challenging material types.

and (t_n, t_f) are the bounds of the ray. Using the quadrature rule for numerical integration [171] and the ray sampling strategy in [152], we approximate the expected color as :

$$\hat{I}(\mathbf{o}_i, \mathbf{v}_i) = \sum_{j \in \mathcal{S}_i} T_j (1 - \exp(-\sigma_j \delta_j)) \mathbf{r}_\psi(\mathbf{x}_j, \mathbf{n}_j, \mathbf{v}) \quad (3.31)$$

Here, \mathcal{S}_i is the set of samples along the ray, δ_j is the distance between each adjacent samples and T_j is the approximated transparency [152]. To realize \mathbf{r}_ψ , we introduce a second MLP with learnable parameters ψ . The radiance fields network \mathbf{r}_ψ is placed subsequent to the signed distance field network f_θ (see Fig. 3.15). Furthermore, we introduce a feature vector $\mathbf{z} \in \mathbb{R}^{256}$ that is extracted from f_θ using a fully connected layer. This feature vector is fed to the radiance field network r_ψ to account for global illumination effects. We optimize f_θ and \mathbf{r}_ψ network on the test subject together. After optimization, we extract the zero level set of f_θ and recover the shape mesh using marching cubes algorithm [172]. For more details, refer to Section §3.6.3.1 and [143].

Optimization. Our overall training loss is as follows:

$$\begin{aligned} \mathcal{L}_{\text{mvps}} = & \frac{1}{S} \sum_{i=1}^S \left(\overbrace{c_i^{\text{mvs}} |f_\theta(\mathbf{p}_i)|}^{\text{MVS Loss}} + \overbrace{c_i^{\text{ps}} \|\mathbf{n}_i^r - \mathbf{n}_i^{\text{ps}}\|}^{\text{PS Loss}} \right. \\ & \left. + \overbrace{(1 - c_i^{\text{mvs}} c_i^{\text{ps}}) \|I_i - \hat{I}(\mathbf{o}_i, \mathbf{v}_i)\|_1}^{\text{Rendering Loss}} \right) \quad (3.32) \\ & + \frac{\lambda_m}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \overbrace{CE(\max_{j \in \mathcal{S}_i}(\sigma_j / \alpha), 0)}^{\text{Mask Loss}} + \overbrace{\lambda_e \mathbb{E}_{\mathbf{x}}(\|\nabla_{\mathbf{x}} f_\theta(\mathbf{x})\| - 1)^2}^{\text{Eikonal Regularization}} \end{aligned}$$

Eq.(3.32) consists of five terms. Here, the first term forces the signed distance to vanish on the high fidelity position predictions of deep-MVS network. Similarly, the second term encourages the expected surface normal on a ray $\mathbf{n}_i^r = \sum_{j \in \mathcal{S}_i} T_j (1 - \exp(-\sigma_j \delta_j)) \mathbf{n}_i(t)$ to align with the highly confident deep-PS predictions. The third term introduces an uncertainty-aware rendering loss to the optimization for the pixels where either MVS or PS fails. Intuitively, this allows the optimization to recover the missing surface details using rendering. We further improve the geometry using the object masks. For that, we first find the maximum density on rays outside the object mask (i.e. $i \in \mathcal{M}$). Then, we apply cross-entropy loss (CE) to minimize ray and geometry intersections as in [152]. The final term applies Eikonal

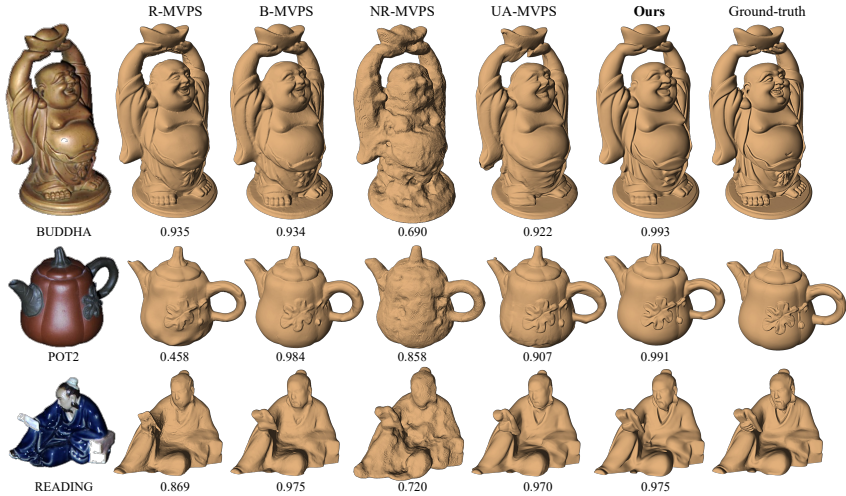


FIGURE 3.16: Comparison of MVPS reconstructions on DiLiGenT-MV benchmark [91]. We report F -score metric results for numerical comparison. We can observe that our method recovers fine details and provides high-quality reconstructions of challenging objects.

regularization for plausible surface recovery as discussed in Section §3.6.2.1. Fig.(3.15) shows the overall pipeline of our proposed approach.

3.6.3 Experiments and Results

3.6.3.1 Implementation Details

We implemented our method in Python 3.8 using PyTorch 1.7.1 [196] and conducted all our experiments on a single NVIDIA GPU with 11GB of RAM. We first train uncertainty-aware deep-MVS and deep-PS networks under a supervised setting. For these networks, we use the same hyperparameters as described in §3.5.3.1. Then, we use these networks to have 3D position and surface normal predictions at test time. Finally, MVS images, along with the network predictions and their per-pixel confidence values, are used to optimize the proposed loss function (Eq.(3.32)).

As described in Section §3.6.2.2, we optimize two networks during optimization: signed distance field network (f_θ) and radiance field network (\mathbf{r}_ψ). f_θ consists of 8 MLP layers with a skip connection connecting the first layer to the 4th. On the other hand, \mathbf{r}_ψ has four MLP layers (see Fig.3.15). All the

Method Category →		Deep Multi-View Stereo		Photometric Stereo			View-Synthesis		
Dataset ↓	Method →	MVSNet [135]	PM-Net [176]	Robust PS [20]	SDPS-Net [54]	CNN-PS [14]	NeRF [88]	VolSDF [143]	Ours
	BEAR	0.135	0.672	0.266	0.239	0.293	0.865	0.962	0.965
	BUDDHA	0.147	0.799	0.367	0.298	0.363	0.713	0.786	0.993
	COW	0.095	0.734	0.245	0.447	0.511	0.810	0.985	0.987
	POT2	0.126	0.666	0.231	0.464	0.632	0.859	0.946	0.991
	READING	0.115	0.834	0.242	0.188	0.508	0.673	0.683	0.975
	AVERAGE	0.124	0.741	0.270	0.327	0.461	0.784	0.873	0.982

TABLE 3.12: F -score comparison of standalone method reconstructions on DiLiGenT-MV benchmark [91]. Our method outperforms standalone multi-view stereo, photometric stereo and view synthesis methods in all of the object categories.

Dataset ↓	Method →	R-MVPS [108]	B-MVPS [91]	NR-MVPS [195]	UA-MVPS [58]	Ours
	BEAR	0.504	0.986	0.856	0.895	0.965
	BUDDHA	0.935	0.934	0.690	0.922	0.993
	COW	0.915	0.989	0.844	0.979	0.987
	POT2	0.458	0.984	0.858	0.907	0.991
	READING	0.869	0.975	0.720	0.970	0.975
	AVERAGE	0.736	0.974	0.794	0.935	0.982

TABLE 3.13: F -score comparison of MVPS reconstructions on DiLiGenT-MV benchmark [91]. Our method performs consistently well on various objects and is better than others on average.

layers of both networks have 256 units. We apply Fourier feature encoding to the inputs (position \mathbf{x} and view direction \mathbf{v}) to improve the networks’ ability to represent high-frequency details [88]. For the loss function in Eq:(3.32), we set $\lambda_m = 0.1$ and $\lambda_e = 1$. We use a set of multi-view images which are captured under the illumination of the same randomly chosen light source to compute the rendering loss. We use Adam optimizer [156] with learning rate 10^{-4} and train for 10^4 epochs. In each epoch, we use batches of 1024 rays from each view and sample 64 points along each ray [143]. To compute the Eikonal regularization as in Eq:(3.28), we also uniformly sample points globally. So, the distribution \mathcal{P} stands for the collection of these ray samples and global samples. After the optimization, we extract zero level set of the learned SDF representation by f_θ and recover the shape mesh using marching cubes algorithm [172] on a 512^3 grid.

3.6.3.2 Statistical Analysis

We performed comparative analysis on the DiLiGenT-MV dataset [91]. To evaluate the quality of the shape reconstructions, we use well-known

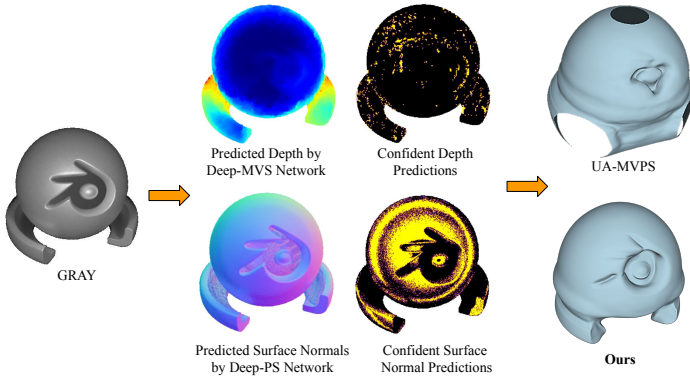


FIGURE 3.17: We show depth and surface normal predictions on texture-less object. Pixels marked with yellow color indicate confident MVS or PS predictions (c_i^{MVS} and c_i^{PS}). Note that MVS cannot predict depth reliably on texture-less surface, which leads to inferior results in UA-MVPS [58]. On the other hand, our uncertainty-aware volume rendering approach can recover missing surface information, and therefore, provides better reconstructions.

Chamfer- L_2 and F -score [197] metric. For better understanding, we present the performance comparison result in two different categories depending on the method type.

(a) Standalone Method Comparison. By the standalone method, we refer to the approaches that use only one modality i. e., either MVS or PS images for 3D reconstruction. We consider state-of-the-art MVS, PS, and view-synthesis methods for this comparison. Note that we use Horn and Brooks algorithm [94] for normal integration to recover depth maps. We then back-project the recovered depths to 3D space to evaluate reconstruction performance. Table 3.12 presents the F -score comparison of these methods on DiLiGenT-MV [91]. The statistics show that our method consistently outperforms the standalone approaches. Further, we observed that none of the standalone methods could reliably recover the object’s 3D shape. On the contrary, our method gives accurate reconstruction by effectively exploiting the complementary surface and image priors.

(b) MVPS Methods Comparison. Table 3.13 provides the F -score comparison results with SOTA MVPS methods on the DiLiGenT-MV benchmark dataset. For our comparison, we consider both explicit geometry modeling-based classical approaches [91, 108], and neural implicit representation

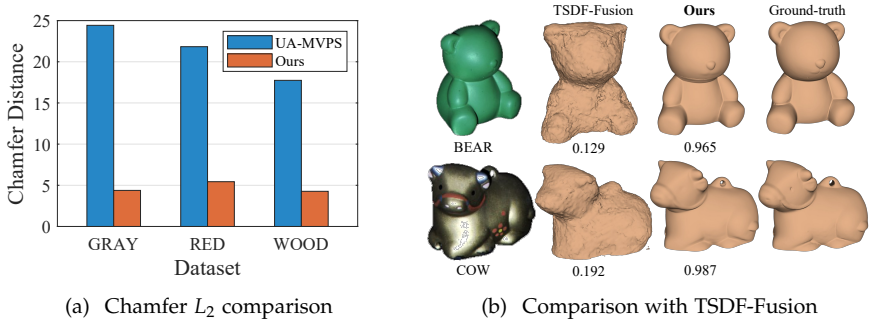


FIGURE 3.18: (a) Chamfer L_2 comparison of our method with UA-MVPS [58] on our synthetic dataset (lower is better). (b) Comparison of our method with TSDF Fusion algorithm [189]. We report F -score metric for numerical comparison.

Settings ↓ Dataset →	BEAR	BUDDHA	COW	POT2	READING	AVERAGE
w/o MVS Loss	0.189	0.089	0.202	0.156	0.353	0.198
w/o PS Loss	0.301	0.572	0.184	0.262	0.428	0.349
w/o Rendering Loss	0.154	0.471	0.269	0.235	0.374	0.301
w/o Uncertainty-Aware.	0.267	0.085	0.313	0.137	0.251	0.211
Ours	0.213	0.088	0.176	0.198	0.253	0.186

TABLE 3.14: Contribution of MVS, PS, rendering loss terms and uncertainty modeling to our reconstruction quality. We report Chamfer L_2 metric for comparison (lower is better). Clearly, our proposed loss in Eq.(3.32) produces best results on average.

based deep approaches [58, 195]. The numerical results show that our method provides the highest scores on three objects categories. Moreover, it outperforms all the existing MVPS methods on average. Some important point to note is that (i) Our approach provides a scalable and easy-to-execute implementation, without requiring tedious sequential steps as in classical methods [91], (ii) Our MLP based shape representation requires only 3.07MB of memory, while explicit geometric methods may require up to 90MB. Such advantages make our method an efficient and effective algorithmic choice for solving MVPS.

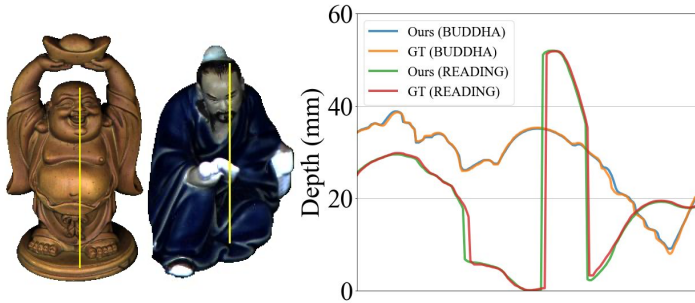


FIGURE 3.19: Surface profile of our reconstructions on a randomly chosen path. Clearly, our surface profile overlaps with the ground-truth(GT), which indicates the high quality of our reconstructions.

3.6.3.3 Further Analysis

(a) Anisotropic and Textureless Glossy Surfaces. We perform evaluations on our synthetic dataset to analyze the efficiency of our approach on anisotropic and texture-less glossy surfaces. In Fig.3.18(a), we provide Chamfer L_2 metric comparison of our method with the recent UA-MVPS [58]. The results show that our method performs much better than its competitor on glossy (Gray, Red) and anisotropic surfaces (Wood). In Fig.3.17, we show qualitative results of the uncertainty-aware deep-MVS and deep-PS networks on the Gray category. It can be observed from visual results that deep-MVS cannot provide reliable position estimates on texture-less glossy surfaces. For this reason, methods relying on the fusion of only MVS and PS priors (such as UA-MVPS) cannot handle all kinds of surfaces. On the other hand, our volume rendering based method can recover the missing surface information; hence, it can suitably work for anisotropic and glossy surface profiles.

(b) Optimization. Here, we investigate the effectiveness of our proposed optimization loss in Eq:(3.32) with an ablation study. For that, we compare the reconstruction quality of our method by removing (i) MVS loss term, (ii) PS loss term, (iii) rendering loss term and (iv) uncertainty modeling (c_i^{mvs} and c_i^{ps}) from the overall loss. In Table 3.14, we provide Chamfer L_2 metric comparison of the reconstruction quality achieved under each of these configurations. The numerical results verify that uncertainty modeling based integration of MVS, PS and rendering loss terms provides best results on DiLiGenT-MV [91].

(c) Surface Profile. To show the quality of our recovered 3D reconstructions, we study the surface topology across an arbitrarily chosen curve on the surface. Fig.3.19 shows a couple of examples of such surface profile on Buddha and Cow sequences. Clearly, our recovered surface profiles align well with the ground truth.

(d) Volumetric Fusion Approach. Of course, one can use robust 3D fusion method such as TSDF fusion [189] to recover the object’s 3D reconstruction. And therefore, we conducted this experiment to study the results that can be recovered using such fusion techniques. Accordingly, we fuse deep-MVS depth and the depth from deep-PS normal integration [94] using the TSDF fusion. Fig.3.18(b) shows that TSDF fusion provide inferior results compared to ours.

(e) Limitations. Although our method works well on glossy objects, it may fail on materials with mirror reflection. Furthermore, SDF representation of the object shape restricts our approach to solid and opaque materials. Finally, our work considers a calibrated setting for MVPS setup, and it would be interesting to further investigate our approach in an uncalibrated setup.

3.6.4 Conclusion

The proposed method addresses the current limitations of well-known MVPS methods and makes it work well for diverse object material types. Experimental studies on anisotropic and texture-less glossy objects show that existing MVS and PS modeling techniques may not always extract essential cues for accurate 3D reconstructions. However, by integrating incomplete yet reliable MVS and PS information into a rendering pipeline and leveraging the generalization ability of the modern view synthesis approach to model complex BRDFs, it is possible to make MVPS setup work well for anisotropic materials and glossy texture-less objects with better accuracy. Finally, the performance on the standard benchmark shows that our method outperforms existing methods providing exemplary 3D reconstruction results. To conclude, we believe that our approach will open up new avenues for applying MVPS to real-world applications such as metrology, forensics, etc.

NEURAL ARCHITECTURE SEARCH FOR UNCALIBRATED PHOTOMETRIC STEREO

4.1 MOTIVATION

In recent years, deep neural networks have significantly improved the performance of many computer vision tasks, including photometric stereo. Their powerful ability to learn from data has helped in modeling surfaces with unknown reflectance properties, which was a challenge for traditional PS methods. Further, neural networks can implicitly learn the image formation process and global illumination effects from data, which classical algorithms cannot pursue. As a result, several deep learning architectures were proposed for PS [14, 15, 17, 43, 50, 51, 67]. Hence, by leveraging a deep neural network, we can overcome the shortcoming of PS due to the Lambertian object assumption. However, these methods still rely on the other assumption of calibrated setting i.e., the light source directions are given at test time, limiting their practical application. Accordingly, uncalibrated deep PS methods that can provide results comparable to calibrated PS networks are becoming more and more popular [16, 54, 85].

The impressive results demonstrated by deep uncalibrated PS methods have a few critical issues: the network architecture is manually designed, and therefore, such networks are typically not optimally efficient and have a large memory footprint [16, 43, 54, 85]. Moreover, the authors of such networks conduct many experiments to explore the effect of empirically selected operations and tune hyperparameters. But, we know from the popular research in machine learning that not only the type of operation but sometimes their placement (ordering) matters for performance [48, 199]. And therefore, a separate line of research known as Neural Architecture Search (NAS) has gained tremendous interest to tackle such challenges in architecture design. NAS methods automate the design process, greatly reducing human effort in searching for an efficient network design. NAS algorithms have shown great success in many high-level computer vision tasks such as object detection [200, 201], image classification [202], image super-resolution [203], action recognition [204], and semantic segmentation [205]. Yet, its potential for low-level 3D computer vision problem such as uncalibrated PS remains unexplored.

4.2 CONTRIBUTIONS

In this chapter, we make the following contributions:

- We propose the first differentiable NAS-based framework to solve uncalibrated photometric stereo problem.
- Our architecture search methodology considers the task-specific constraints of photometric stereo during search, train, and test time to discover meaningful architecture.
- We show that automatically designed architecture outperforms the existing traditional uncalibrated PS performance. The experiments reveal that our approach discovers lightweight architectures, which provides results comparable to the state-of-the-art manually designed deep uncalibrated networks [43, 54, 85] with significantly less parameters.

4.3 METHOD

This section describes our task-specific neural architecture search (NAS) approach. Among architecture search methods, evolutionary algorithms [206, 207] and reinforcement learning-based methods [208, 209] are computationally expensive and need thousands of GPU hours to find architecture. Hence they are not suitable for our problem. Instead, we adhere to the cell-based differentiable NAS formulation. It has proven itself to be computationally efficient and demonstrated encouraging results for many high-level vision problems [210, 211]. However, in those applications, differentiable NAS is used without any task-specific treatment. Unfortunately, this will not work for the uncalibrated PS problem. There exists GBR ambiguity [26] due to the lack of light source information. Moreover, certain task-specific constraints must be satisfied (e. g., unit normal, unit light source direction), and the method must operate on unordered image sets. Unlike typical NAS-based methods, we incorporate human knowledge in our search strategy to address those challenges. To resolve GBR ambiguity, we first search for an efficient light calibration network, followed by a normal estimation network’s search [54]. To handle PS-related constraints, we fix some network layers and define our discrete search space for both networks accordingly. We model our PS architecture search space via a continuous relaxation of the discrete search space, which can be optimized efficiently using a gradient-based algorithm. We utilize the seminal classical photometric

stereo formulation [11, 26] and previous handcrafted deep neural network design [54] as the basis of our NAS framework. Utilizing previous methods knowledge in the architecture design process not only helps in reducing the architecture’s search time but also provides an optimal architecture with better performance accuracy [54, 85].

4.3.1 *Architecture Search for Uncalibrated PS*

Leveraging the recent one-shot cell-based NAS method i.e., DARTS [210], we first define different discrete search spaces for light calibration and normal estimation networks. Next, we perform a continuous relaxation of these search spaces, leading to differentiable bi-level objectives for optimization. We perform an end-to-end architecture search for light calibration and normal estimation networks separately to obtain optimal architectures. Contrary to high-level vision problems such as object detection, image classification, and others [209, 210, 212], directly applying the one-shot NAS to existing uncalibrated PS networks [54, 85] may not necessarily lead to a good solution. Unfortunately, for our task, a single end-to-end NAS seems challenging. It may lead to unstable behavior due to GBR ambiguity [26]. And therefore, we search for an optimal light calibration first and then search for a normal estimation network by keeping some of the necessary operations or layers fixed—such a strategy is used in other NAS based applications [213]. The searched architectures are then trained independently for inference.

4.3.1.1 *Background on Differentiable NAS.*

In recent years, Neural Architecture Search (NAS) has attracted a lot of attention from the computer vision research community. The goal of NAS is to automate the process of deep neural network design. Among several promising approaches proposed in the past [206, 209, 210, 214–216], the DARTS [210] has shown promising outcomes due to its computational efficiency and differentiable optimization formulation. So, in this chapter, we use it to design an efficient deep neural network to solve uncalibrated PS.

DARTS searches for a computational cell from a set of defined search spaces, which is a building block of the architecture. Once the optimal cells are obtained, it is stacked to construct the final architecture for training and inference. To find the optimal cell, we define search space \mathcal{O} , that is a set

of possible candidate operations. The method first performs continuous relaxation on the search spaces and then searches for an optimal cell. A cell is a directed acyclic graph (DAG) with N nodes and E edges. Each node is a latent feature map representation say $x^{(i)}$ for the i^{th} node and each edge is associated with an operation say $o^{(i,j)}$ between node i and node j (see Fig.4.1(a)). In a cell, each intermediate node is computed from its preceding nodes as follows:

$$x^{(j)} = \sum_{i < j} o^{(i,j)}(x^{(i)}) \quad (4.1)$$

Let $o^{(i,j)}$ be some operation among K candidate operations $\mathcal{O} = \{o_1^{(i,j)}, o_2^{(i,j)}, \dots, o_K^{(i,j)}\}$. The categorical choice of a specific operation is replaced by the continuous relaxation of the search space by taking softmax over all the defined candidate operations as follows:

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x) \quad (4.2)$$

Here, $\alpha^{(i,j)}$ is a vector of dimension $|\mathcal{O}|$ which denotes the operation mixing weights on edge (i, j) (see Fig.4.1(b)). As a result, the search task for DARTS reduces to a learning set of continuous variable $\alpha^{(i,j)} \forall (i, j)$. The optimal architecture will be determined replacing each mixed operation $\bar{o}^{(i,j)}$ on edge (i, j) with: $o^{(i,j)} = \arg \max_{o \in \mathcal{O}} \alpha_o^{(i,j)}$ corresponding to the operation which is the ‘‘most probable’’ among the ones listed in \mathcal{O} (see Fig.4.1(c)-Fig.4.1(d)). The introduced relaxation allows joint learning of architecture α and its weight ω within the mixture of operations. So, the goal of architecture search now becomes to search for an optimal architecture α using the validation loss with the weights ω that minimizes the training loss for a given α . This leads to following bi-level optimization problem.

$$\begin{aligned} & \underset{\alpha}{\text{minimize}} \quad \mathcal{L}_{val}(\omega^*(\alpha), \alpha); \\ & \text{subject to: } \omega^*(\alpha) = \underset{\omega}{\arg \min} \quad \mathcal{L}_{train}(\omega, \alpha) \end{aligned} \quad (4.3)$$

where, \mathcal{L}_{val} and \mathcal{L}_{train} are the validation and training losses respectively. This optimization problem is solved iteratively until convergence is reached. The architecture α is updated by substituting the lower-level optimization gradient approximation. Concretely, update α by descending $\nabla_{\alpha} \mathcal{L}_{val}(\omega - \xi \nabla_{\omega} \mathcal{L}_{train}(\omega, \alpha), \alpha)$. Subsequently update ω by descending $\nabla_{\omega} \mathcal{L}_{train}(\omega, \alpha)$, where:

$$\nabla_{\alpha} \mathcal{L}_{val}(\omega - \xi \nabla_{\omega} \mathcal{L}_{train}(\omega, \alpha), \alpha) \approx \nabla_{\alpha} \mathcal{L}_{val}(\omega^*(\alpha), \alpha) \quad (4.4)$$

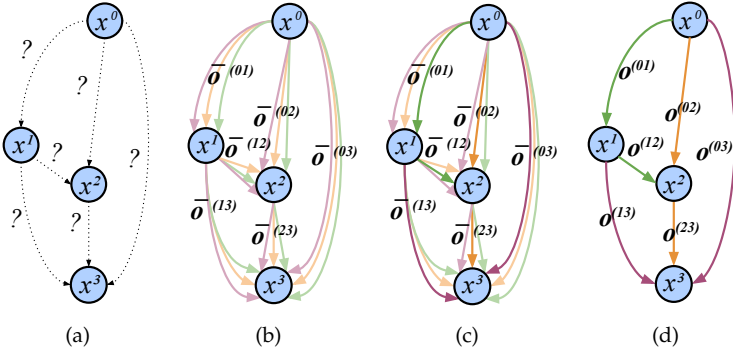


FIGURE 4.1: Illustration of a cell. (a) Initially, the optimal operations $\bar{o}^{(i,j)}$ between nodes $x^{(i)}$ and $x^{(j)}$ are unknown. (b) Each node is computed by a mixture of candidate operations. (c) Architecture encoding is obtained by solving the continuous relaxation of the search space. (d) Optimal cell obtained after selection of most probable candidate operation.

$\xi > 0$ is the learning rate of the inner optimization. The idea is that, $\omega^*(\alpha)$ is approximated with a single learning step which allows the searching process to avoid solving the inner optimization in Eq.(4.3) exactly. We refer this formulation as second-order approximation [210]. To speed up the searching process, common practice is to apply first-order approximation by setting $\xi = 0$. For more details on the bi-level optimization refer Liu *et al.* work [210].

4.3.1.2 Our Cell Description

For our problem, we search for both light calibration and normal estimation networks. Our cells consist of two input nodes, four intermediate nodes, and one output node for both of the networks. Each cell at layer k uses the output of two preceding cells (\mathcal{C}_{k-1} and \mathcal{C}_{k-2}) at input nodes and outputs \mathcal{C}_k by channel-wise concatenation of the features at the intermediate nodes. To adjust the spatial dimensions, we define two cells i.e., *normal cells* and *reduction cells*. Normal cells preserve the spatial dimensions of the input feature maps by applying convolution operations with stride 1. The reduction cells use operations with stride 2 adjacent to input nodes, reducing the spatial dimension by half. Although the cell definition for both networks is the same, the network-level search spaces are different

due to the problem’s constraints. Next, we describe our procedure to obtain optimal network architecture for uncalibrated PS.

4.3.2 Light Calibration Network

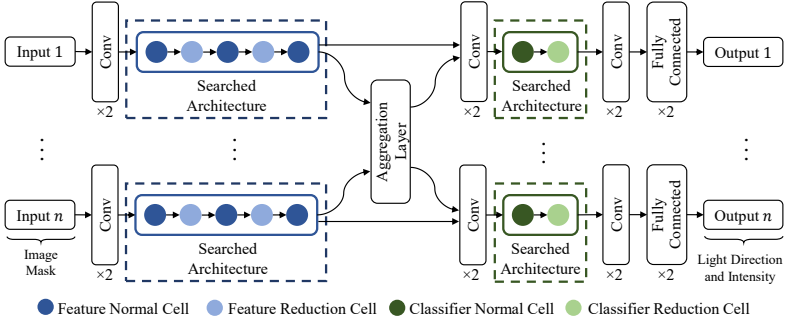
Light calibration network predicts all the light source’s direction and intensity from a set of PS images. Here, we assume the object mask is known. One obvious way to estimate light is to regress a set of images with the source direction vectors and intensities in a continuous space. However, converting this task into a classification problem is more favorable for our purpose. It stems from the fact that learning to classify light source directions to predefined bins of angles is much easier than regressing the unit vector itself. Further, using discretized light directions makes the network robust to small input variations.

We represent the light source direction in the upper-hemisphere by its azimuth $\phi \in [0, \pi]$ and elevation $\theta \in [-\pi/2, \pi/2]$ angles. We divide the angle spaces into 36 evenly spaced bins ($K_d = 36$). Our network perform classification on azimuth and elevation separately. For the light intensities, we assign the values in the range of $[0.2, 2]$ divided uniformly into 20 bins ($K_e = 20$) [54].

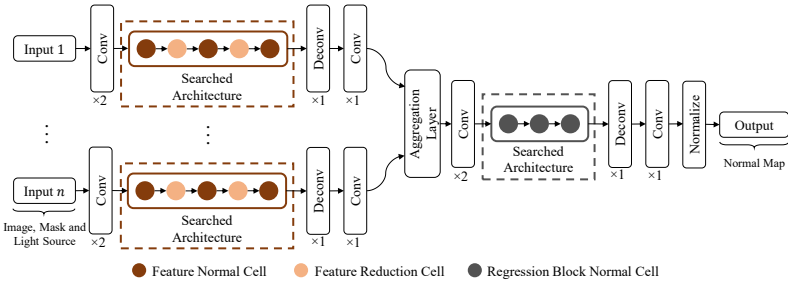
NAS for Light Calibration Network. To perform NAS for light calibration network, we use the backbone shown in Fig.4.2(a). The backbone consists of three main parts (*i*) local feature extractor (*ii*) aggregation layer and (*iii*) classifier. The feature extraction layers provide image-specific information for each input image. The weights of these feature extraction layers are shared among all input images. The image-specific features are then aggregated to a global feature representation with the max-pooling operation. Later, global feature representation is combined with the image-specific information and fed to the subsequent layers for classification. The fully connected layers provide softmax probabilities for azimuth, elevation, and intensity values.

We use the NAS algorithm to perform search only over the feature extraction layer and classifier layers for architecture search (shown with dashed box Fig.4.2(a)), while keeping other layers fixed. For NAS to provide optimal architecture over the searchable blocks in the light-calibration network backbone, we define our search space as follows:

1. *Search Space.* Our candidate operations set in search space for light calibration network is composed of $\mathcal{O}^{light} = \{“1 \times 1 \text{ separable conv.}”, “3 \times 3 \text{ separable conv.}”, “5 \times 5 \text{ separable conv.}”, “\text{skip connection}”, “\text{zero}”\}$.



(a) Light Calibration Network



(b) Normal Estimation Network

FIGURE 4.2: Our pipeline consists of two networks: (a) Light Calibration Network predicts light source directions and intensities from images. Our search is confined to feature extraction module and classification module. (b) Normal Estimation Network outputs the surface normal map from images and estimated light sources. Our search is confined to feature extraction module and regression module.

The “zero” operation indicates the lack of connection between two nodes. Each convolutional layer defined in the set first applies ReLU [69] and then convolution with given kernel size followed by batch-normalization [73]. As before, our cells consist of two input nodes, four intermediate nodes, and one output node §4.3.1.2. Just for the initial cell, we use stem layers as its input for better search. These layers apply fixed convolutions to enrich the initial cell input features.

2. *Continuous Relaxation and Optimization.* We perform the continuous relaxation of our defined search space using Eq.(4.2) for differentiable optimiza-

tion. During searching phase, we perform alternating optimization over weights ω and architecture encoding values α as follows:

- Update network weights ω by $\nabla_{\omega} \mathcal{L}_{train}(\omega, \alpha)$.
- Update architecture mixing weights α by $\nabla_{\alpha} \mathcal{L}_{val}(\omega - \xi \nabla_{\omega} \mathcal{L}_{train}(\omega, \alpha), \alpha)$. (see Eq.(4.4))

\mathcal{L}_{train} and \mathcal{L}_{val} denote the loss computed over training and validation datasets, respectively. We use multi-class cross-entropy loss on azimuth, elevation, and intensity classes to optimize our network [54]. The total light calibration loss is:

$$\mathcal{L}_{light} = \mathcal{L}_{az} + \mathcal{L}_{el} + \mathcal{L}_{in} \quad (4.5)$$

where, \mathcal{L}_{az} , \mathcal{L}_{el} , and \mathcal{L}_{in} are the losses for azimuth, elevation, and intensity respectively. We utilize the synthetic Blobby and Sculpture datasets [43] for this optimization where ground-truth labels for lighting are provided.

Once the searching phase is complete, we convert the continuous architecture encoding values into a discrete architecture. For that, we select the strongest operation on each edge (i, j) with: $o^{(i,j)} = \arg \max_{o \in \mathcal{O}^{light}} \alpha_o^{(i,j)}$. We preserve only the strongest two operations preceding each intermediate node. We train our designed architecture with optimal operations from scratch on the training dataset again to optimize weights before testing §4.4.1.

4.3.3 Normal Estimation Network

We independently search for optimal normal estimation network using the backbone shown in Fig.4.2(b). To use the light source information into the network, we first convert n light direction vectors into a tensor $\mathcal{X} \in \mathbb{R}^{n \times 3 \times h \times w}$, where each 3-vector is repeated over spatial dimensions h and w . This tensor is then concatenated with the input image to form a tensor $\mathcal{I} \in \mathbb{R}^{n \times 6 \times h \times w}$. Similar to the light calibration network, we use a shared-weight feature extraction block to process each input. After image-specific information is extracted, we combine them in a fixed aggregation layer with the max-pooling operation and obtain a global representation. Keeping the aggregation layer fixed allows the network to operate on an arbitrary number of test images and improves robustness. The global information is finally used to regress the normal map, where a fixed normalization layer is used to satisfy the unit-length constraint.

NAS for Normal Estimation Network. Similar to light calibration network, the cells here consist of two input nodes, four intermediate nodes, and one output node. To efficiently search for architectures at initial layers, we make use of stem layers prior to each search space [210]. These layers apply fixed convolutions to enrich the input features.

1. *Search Space.* It is a well-known fact that the kernel size has great importance in vision problems. Recent work on photometric stereo has verified that using bigger kernel size helps to explore the spatial information, but stacking too many of them leads to over-smoothing and degrades the performance [48]. Therefore, we selectively use different kernel sizes in the candidate operations set $\mathcal{O}^{normal} = \{ "1 \times 1 \text{ separable conv.}", "3 \times 3 \text{ separable conv.}", "5 \times 5 \text{ separable conv.}", "skip connection", "zero" \}$. Here also, each convolutional layer defined in the set first applies ReLU [69] and then convolution with given kernel size followed by batch-normalization [73].

2. *Continuous Relaxation and Optimization.* Similar to light calibration network, we use Eq.(4.2) to make the search space continuous. We then jointly search for the architecture encoding values and the weights using the ground-truth surface normals and light source information during optimization. The optimization is performed using the same bi-level optimization approximation strategy (see Eq.(4.3) and Eq.(4.4)). We normalize the images before feeding them to the network. The normalization ensures the network is robust to different intensity levels. To search normal estimation network, we use the following cosine similarity loss:

$$\mathcal{L}_{normal} = \frac{1}{m} \sum_i^m (1 - \tilde{\mathbf{n}}_i^T \mathbf{n}_i) \quad (4.6)$$

where, m is the number of pixels, $\tilde{\mathbf{n}}_i$ is the estimated normal by our network and \mathbf{n}_i is the ground-truth normal at pixel i . Note that $\tilde{\mathbf{n}}_i$ is a unit-vector due to the fixed normalization layer.

After the search optimization for normal estimation network is done, we obtain optimal discrete architecture by keeping the operation $o^{(i,j)} = \arg \max_{o \in \mathcal{O}^{normal}} \alpha_o^{(i,j)}$ on each edge (i, j) . Similar to [210], we only preserve the two preceding operations with highest weight for each node. Finally, we train our normal estimation network from scratch using the searched architecture. Our normal estimation network uses the light directions and intensities estimated by the light calibration network to predict normals at test time.

4.4 EXPERIMENTS AND RESULTS

This section first describes our procedure in preparing the dataset for the searching, training, and testing phase. Later, we provide the implementation of our method, followed by statistical evaluations and ablation.

4.4.1 Dataset Preparation

We used three well-known photometric stereo datasets for our experiments, statistical analysis, and comparisons, namely, Blobby [217], Sculpture [80], and DiLiGenT [163].

Search and Train Set Details. For architecture search and optimal architecture training, we used 10 objects from the Blobby dataset [217] and 8 from the Sculpture dataset [80]. We considered the rendered photometric stereo images of these datasets provided by Chen *et al.* [43]. It uses 64 random lights to render the objects. In search and train phase, we randomly choose 32 light source images. Following Chen *et al.* [43], we considered 128×128 sized images for both Blobby and Sculpture dataset.

(a) Preparation of Search Set. Searching for an optimal architecture using one-shot NAS [210] can be computationally expensive. To address that, we use only 10% of the dataset such that it contains subjects from all the categories present in the Blobby and Sculpture dataset. Next, we resized all those 128×128 resolution images to 64×64 . We refer this dataset as Blobby search set and Sculpture search set. Our search set is further divided into search train set and search validation set. This train set is prepared by taking eight shapes from Blobby search set and six shape from Sculpture search set. The search validation set is composed of two shapes from Blobby and Sculpture search sets, respectively. Hence, approximately 80% of the search set is used as search train set and 20% is used as search validation set. This is done in a way that there is no common subject between train and validation sets. We used a batch size of four at train and validation time during search phase. The search set is same for the light calibration and normal estimation network’s search.

(b) Preparation of Train Set. Once the optimal architectures for light calibration and normal estimation are obtained, we use the train set for training these networks from scratch. Since, we searched architecture using 64×64 size images, we use convolution layer with stride 2 at the train time for the light calibration network’s training. Following Chen *et al.* [43], we use 99% of the Blobby and Sculpture dataset for training and 1% for the validation.

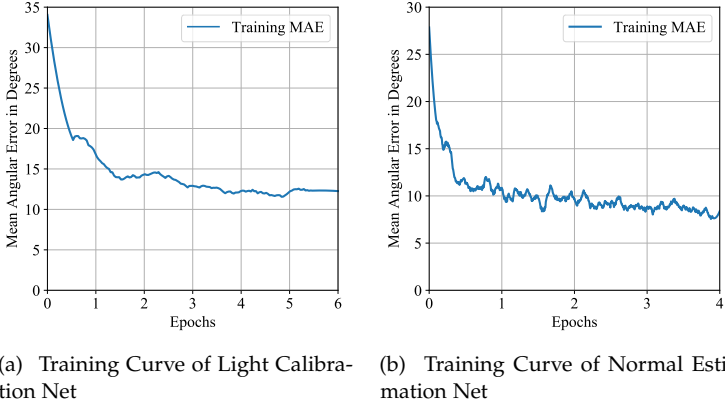


FIGURE 4.3: (a) Training curve of the light calibration network. (b) Training curve of the normal estimation network.

For light calibration we used batch size of thirty-two at train time and eight for validation. For normal estimation, instead, we considered batch size of four both at training and validation.

4.4.2 Implementation Details

The proposed method is implemented with Python 3.6, and PyTorch 1.1 [196]. For both networks, we employ the same optimizer, learning rate, and weight decay settings. The architecture parameters α and the network weights ω are optimized using Adam [156]. During the architecture search phase, the optimizer is initialized with the learning rate $\eta_{\alpha} = 3 \times 10^{-4}$, momentum $\beta = (0.5, 0.999)$ and weight decay of 1×10^{-3} . At model train time, the optimizer is initialized with the learning rate $\eta_w = 5 \times 10^{-4}$, momentum $\beta = (0.5, 0.999)$ and weight decay of 3×10^{-4} . We conducted all the experiments on a computer with a single NVIDIA GPU with 12GB of RAM.

We search for two types of cells, namely normal cell and reduction cell. We use the loss function defined in Eq.(4.5) and Eq.(4.6) during search phase to recover optimal cells for each network independently. Fig.4.2(a) and Fig.4.2(b) show the light calibration and the normal estimation backbone and its searchable parts, respectively. For light calibration network, we have two searchable blocks (i) Feature block and (ii) Classification block.

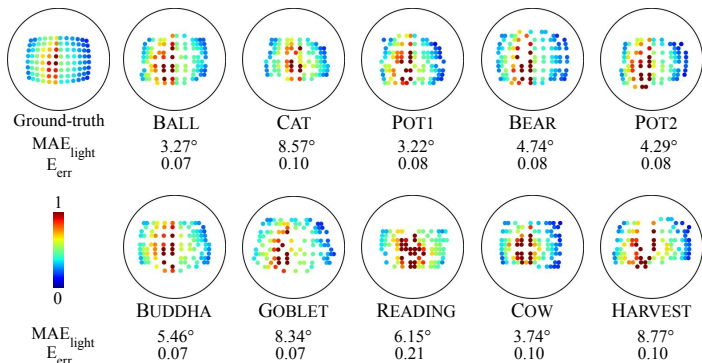


FIGURE 4.4: Light calibration network results on DiLiGenT objects. We show the light direction by projecting the vector $[x, y, z]$ to a corresponding point $[x, y]$. The color of the point shows the light intensity value in $[0, 1]$ range. MAE_{light} is the mean angular error in the estimation of light source direction and E_{err} stands for the intensity error.

Methods _↓	Dataset →	Ball	Cat	Potr	Bear	Potz	Buddha	Goblet	Reading	Cow	Harvest	Average
Alldrin et al. (2007) [65]		7.27	31.45	18.37	16.81	49.16	32.81	46.54	53.65	54.72	61.70	37.25
Shi et al. (2010) [28]		8.90	19.84	16.68	11.98	50.68	15.54	48.79	26.93	22.73	73.86	29.59
Wu & Tan (2013) [34]		4.39	36.55	9.39	6.42	14.52	13.19	20.57	58.96	19.75	55.51	23.93
Lu et al. (2013) [218]		22.43	25.01	32.82	15.44	20.57	25.76	29.16	48.16	22.53	34.45	27.63
Papadh. et al. (2014) [29]		4.77	9.54	9.51	9.07	15.90	14.92	29.93	24.18	19.53	29.21	16.66
Lu et al. (2017) [35]		9.30	12.60	12.40	10.90	15.70	19.00	18.30	22.30	15.00	28.00	16.30
Ours		3.46	8.94	7.76	5.48	7.10	10.00	9.78	15.02	6.04	17.97	9.15

TABLE 4.1: Quantitative comparison with the traditional uncalibrated photometric stereo methods on DiLiGenT benchmark. Our searched architecture estimates accurate surface normals of the object with general reflectance property.

Here, we design our feature block using three normal cells, two reduction cells, and the classification block using one normal cell and one reduction cell. Similarly, we have two searchable blocks (i) Feature block and (ii) Regressor block for normal estimation network. Here, the feature block comprises three normal cells and two reduction cells, while the regressor block is composed of three normal cells. To construct the network design for searchable blocks, each normal cell is concatenated sequentially to the reduction cell in order. We use 3 epochs to search architecture for each network.

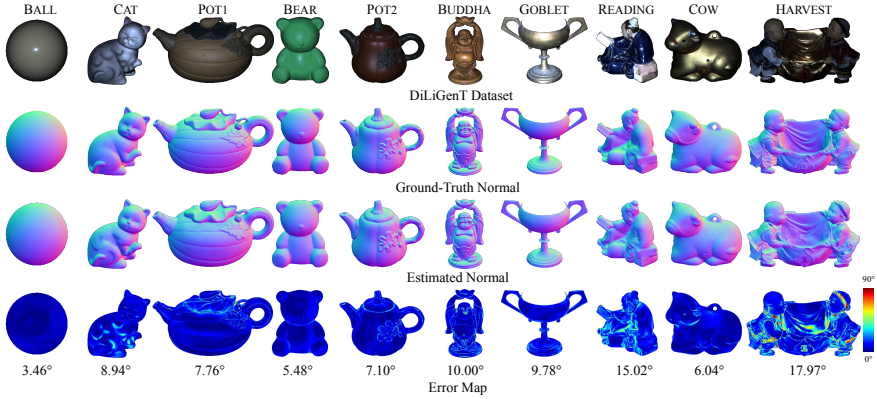


FIGURE 4.5: Qualitative surface normal results on the DiLiGenT benchmark. The bottom row demonstrates the angular error maps and mean angular errors of our results.

Methods	Param.(M)	Ball	Cat	Pot1	Bear	Pot2	Buddha	Goblet	Reading	Cow	Harvest	AVG
UPS-FCN [†] (2018) [43]	6.1	3.96	12.16	11.13	7.19	11.11	13.06	18.07	20.46	11.84	27.22	13.62
SDPS-Net (2019) [54]	6.6	2.77	8.06	8.14	6.89	7.50	8.97	11.91	14.90	8.48	17.43	9.51
GCNet (2020) [16] + PS-FCN [43]	6.8	2.50	7.90	7.20	5.60	7.10	8.60	9.60	14.90	7.80	16.20	8.70
Neural Inv. Rendering(2021) [85]	8.1	3.78	7.91	8.75	5.96	10.17	13.14	11.94	18.22	10.85	25.49	11.62
Ours (w/o auxiliary)	4.4	4.86	9.79	9.98	4.97	8.95	10.29	9.46	15.59	8.06	18.20	9.98
Ours	4.4	3.46	8.94	7.76	5.48	7.10	10.00	9.78	15.02	6.04	17.97	9.15

TABLE 4.2: Quantitative comparison of deep uncalibrated photometric stereo methods on DiLiGenT benchmark [78]. Our searched architecture on average provides results that are better compared to other deep networks not only in surface orientation accuracy (MAE) but also in model size. The blue show the statistics where our method has the second best performance. We used deeper version of UPS-FCN [43].

At train time, we regularize the normal estimation network loss function using the concept of auxiliary tower [210] for performance gain. Consequently, we modify its loss function at train time as follows:

$$\mathcal{L}_{normal} = \frac{1}{m} \sum_i^m (1 - \tilde{\mathbf{n}}_i^T \mathbf{n}_i) + \lambda_{aux} \frac{1}{m} \sum_i^m (1 - \hat{\mathbf{n}}_i^T \mathbf{n}_i) \quad (4.7)$$

where, λ_{aux} is a regularization parameter, and $\hat{\mathbf{n}}_i$ is the output surface normal at pixel i due to auxiliary tower. We set $\lambda_{aux} = 0.4$. We observed that the auxiliary tower improves the performance of the normal estimation network. It can be argued that a similar regularizer could be used for the light calibration network. However, in that case, we have to incorporate that

regularizer for each image independently, which can be computationally expensive. Fig.4.3(a) and Fig.4.3(b) show the training curve for the light calibration and normal estimation network respectively. We trained the light calibration and normal estimation networks for six and three epochs, respectively for inference.

4.4.3 Qualitative and Quantitative Evaluation

Evaluation Metric. We use the well-known mean angular error (MAE) metric to compute the light calibration and normal estimation accuracy, respectively. Unlike light directions and surface normals, light intensity can only be estimated up to a scale factor. For this reason, instead of using the exact intensity values for evaluation, we use a scale-invariant relative error metric [54]:

$$E_{err} = \frac{1}{n} \sum_i^n \left(\frac{|s\tilde{e}_i - e_i|}{e_i} \right) \quad (4.8)$$

Here, \tilde{e}_i and e_i are the estimated and ground-truth light intensities, respectively with s as the scale factor. Following Chen *et al.* [17], we solve $\operatorname{argmin}_s \sum_i^n (s\tilde{e}_i - e_i)^2$ using the least squares to compute s for intensity evaluation.

Inference. Once optimal architectures are obtained, we train these networks for inference. We test their performance using the defined metric on the Test set. For each test object, we first feed the object images at 128×128 resolution to the light calibration network to predict the light directions and intensities. Then, we use the images and estimated light sources as input to the normal estimation network to predict the surface normals.

(a) Performance of Light Calibration Network. To show the validity of our searched light calibration network, we compared its performance on DiLiGenT ground-truth light direction and intensity. Fig.4.4 shows the quantitative and qualitative results obtained using our network. Concretely, it provides light directions MAE_{light} and intensity error (E_{err}) for all object categories. The results indicate that the searched light calibration network can reliably predict light source direction and intensity from images of object with complex surface profile and different material properties.

(b) Comparison of Surface Normal Accuracy. We documented the performance comparison of our approach against the traditional uncalibrated photometric stereo methods in Table 4.1. The statistics show that our method

performs significantly better than such uncalibrated approaches for all the object categories. That is because we don't explicitly rely on BRDF model assumptions and the well-known matrix factorization approach. Instead, our work exploits the benefit of the deep neural network to handle complicated BRDF problems by learning from data. Rather than using matrix factorization, our work independently learns to estimate light from data and use it to solve surface normals.

Further, we compared our method with the state-of-the-art deep uncalibrated PS methods. Table 4.2 shows that our method achieves competitive results with an average MAE_{normal} of 9.15° , having the second best performance overall. The best performing method [16] uses a four-stage cascade structure, making it complex and deep. On the contrary, our searched architecture is light and it can achieve such accuracy with 2.4M fewer parameters. Fig.4.6 provides additional visual comparison of our results with several other approaches from the literature [29, 34, 43, 54]. Table 4.2 also shows the benefit of using an auxiliary tower at train time.

(c) Ablation Study. *Analysing the performance with the change in number of input images at test time.* Our light calibration and normal estimation network can work with an arbitrary number of input images at test time. In this experiment, we analyse how the number of images affects the accuracy of the estimated lighting and surface normals. Fig. 4.7(a) and 4.7(b) show the variation in the mean angular error with different number of images. As expected, the error decreases as we increase the number of images. Of course, feeding more images allows the networks to extract more information, and therefore, the best results are obtained by using all 96 images provided by the DiLiGenT dataset [78].

4.5 LIMITATIONS AND FURTHER STUDY

In this chapter, we have successfully demonstrated a favorable way to exploit the differentiable NAS to the uncalibrated photometric stereo. By respecting the inherent geometric constraint of uncalibrated PS, we utilize NAS that provides commendable performance and a lightweight neural network design. Still, we believe there are possible future directions to explore and handle the limitations of the current method. Firstly, our method considers a setup where each point is illuminated only by a directional light source. However, each surface element mutually illuminates each other on concave parts, and therefore, our method suffers on such interreflecting surfaces (see Fig. 4.8(a)). As future work we aim to apply NAS on an inverse rendering

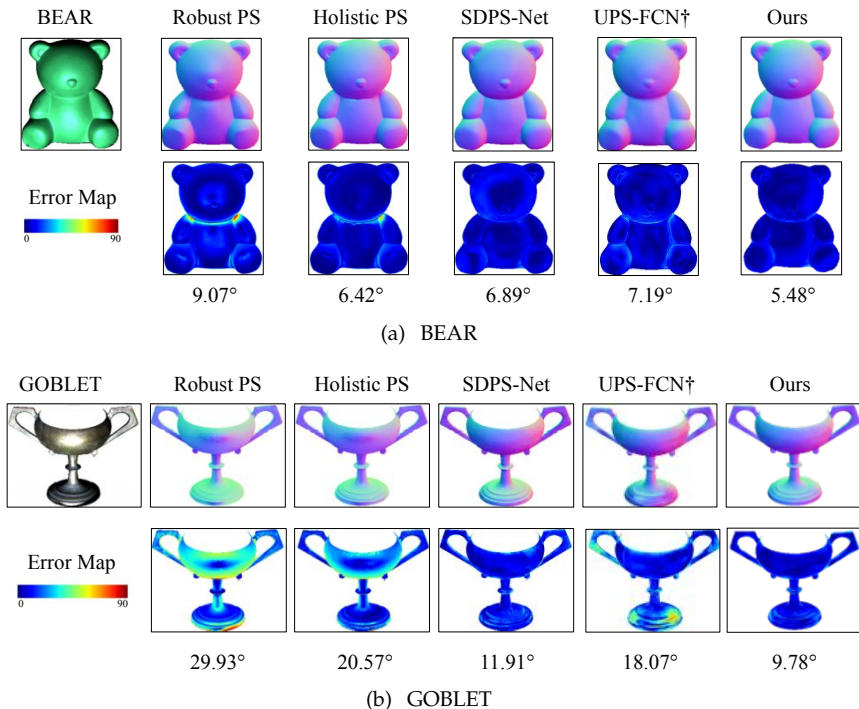


FIGURE 4.6: Visual comparison against Robust PS [29], Holistic PS [34], SDPS-Net [54] and UPS-FCN [43] on (a) BEAR and (b) GOBLET objects from DiLiGenT dataset. The statistics show the superiority of our searched architecture.

pipeline with explicit interreflection modeling as introduced in Chapter 2. Secondly, our method works well on the surfaces with homogeneous BRDF, and we observe degradation in performance on textured regions (see 4.8(b)). That is because the dataset used in training consists of texture-less surfaces. We believe that extending the dataset to spatially varying BRDFs will enhance the performance on such surfaces. As creating a large-scale dataset is not an easy task, it could be possible to improve the performance by exploring the applicability of techniques like channel-wise normalization.

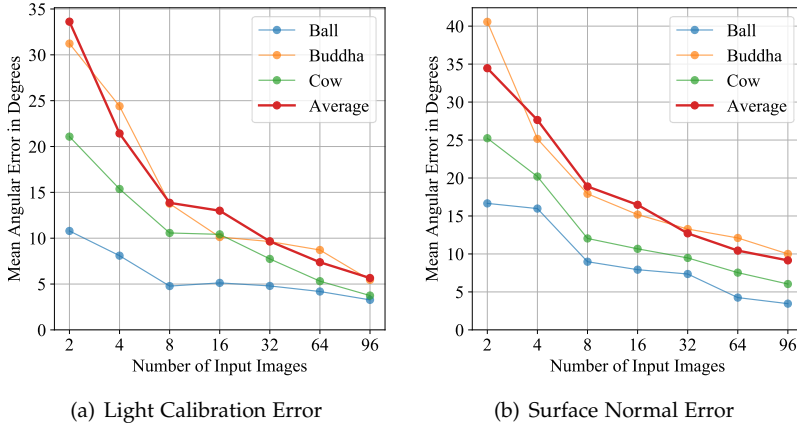
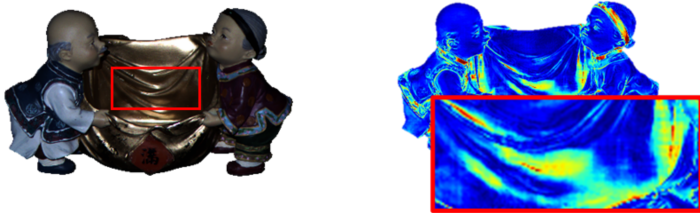


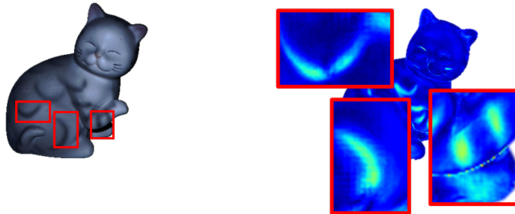
FIGURE 4.7: Variation in MAE w.r.t the change in the number of input images at test time. Observation with (a) light calibration and (b) normal estimation network, respectively.

4.6 CONCLUSION

In this chapter, we demonstrated the effectiveness of applying differentiable NAS to deep uncalibrated PS. Though using the existing differentiable NAS framework directly to our problem is not straightforward, we showed that we could successfully utilize NAS provided PS-specific constraints are well satisfied during the search, train, and test time. We search for an optimal light calibration network and normal estimation network using the one-shot NAS method by leveraging hand-crafted deep neural network design knowledge and fixing some of the layers or operations to account for the PS-specific constraints. The architecture we discover is lightweight, and it provides comparable or better accuracy than the existing deep uncalibrated PS methods.



(a) HARVEST scene



(b) CAT scene

FIGURE 4.8: Failure cases: (a) interreflecting surfaces, (b) textured surfaces.

DISCUSSION

This thesis addresses the problem of surface recovery by taking a modern approach to photometric 3D reconstruction techniques. Chapter 2 presents an uncalibrated inverse rendering approach to estimate the surface normals of objects. Our experimental analysis shows that our method can perform comparable to or better than the fully-supervised methods that require the acquisition of ground-truth normals for training. It is important to note that creating a large-scale dataset of real-world objects for photometric stereo is highly challenging. As a remedy, existing methods heavily rely on synthetic renderings. On the other hand, our approach benefits from the physics to analyze light-surface interactions. To this end, it uses Nayar’s interreflection kernel and a specular mapping based on the Phong model. Furthermore, it exploits an initialization step based on robust photometric stereo. Therefore, we consider our approach a big step for physics-based deep learning. We believe that our method can be further improved by adding more physical constraints, such as using more complicated reflection models, adopting Monte Carlo methods for handling specular interreflections, or utilizing additional training data.

In Chapter 3, we introduced simple and modern deep learning-based solutions to the multi-view photometric stereo (MVPS) problem. Firstly, we want to indicate that MVPS is not an ordinary 3D data acquisition setup that can be realized with common commodity cameras. It requires sophisticated hardware, and special care must be taken to calibrate cameras and lights. Only then, it becomes possible to acquire 3D and render scenes accurately. Secondly, we want to emphasize again that MVPS is generally solved using a sequence of involved steps. Hence, the main motivation of this thesis is to utilize the modern approach for the classical MVPS problem and explore how far we can go with it (with a framework that is as simple as possible). We show that we can get state-of-the-art reconstruction results with much simpler frameworks by leveraging uncertainty modeling and continuous volumetric rendering approaches. All in all, this thesis provides new approaches to MVPS, and we believe that working on such ideas can help us come up with a better and simpler way to recover 3D from MVPS images.

Our experimental analysis of the MVPS problem shows that mindful integration of surface estimates from different modalities can further improve reconstruction quality. Therefore, we believe that better reconstructions can be achieved by employing more sophisticated acquisition devices. An exciting future research direction is to utilize polarization cameras [219–221] and multispectral images [222, 223] in MVPS setup. Moreover, our proposed approaches consider only the calibrated setting, and it would be interesting to explore the applications without calibration data.

Finally, in Chapter 4, we introduced a differentiable neural architecture search-based framework for the uncalibrated photometric stereo problem. Our approach provides promising results for the application of neural architecture search on low-level vision problems. By considering the task-specific constraints of the uncalibrated photometric stereo problem, we obtained a lightweight architecture that performs comparable to and better than the state-of-the-art hand-crafted architectures. Since making neural networks lighter is important for better deployment, we believe that our work will facilitate the application of deep photometric stereo in real-world settings.

APPENDIX

A.1 UNCALIBRATED NEURAL INVERSE RENDERING

Here, we present qualitative results on all of the categories of our dataset. Figure A.1 to Figure A.6 compares the output normal maps of our method with other baselines. Note that our implementation of Nayar *et al.* [68] uses Woodham’s classical photometric stereo [11] to calculate the pseudo surface and updates the normals with the interreflection modeling for 15 iterations. Even though the Nayar *et al.* [68] interreflection algorithm is not theoretically guaranteed to converge for all surfaces, it gives a stable response on our dataset. We initialized Nayar’s algorithm using the same predicted light sources of our method for a fair comparison.

The results show that our method achieves the best results overall, both qualitatively and quantitatively. We observed that other deep learning networks [43, 54] may fail to remove the surface ambiguity in challenging subjects. This is because these networks require supervised training with ground-truth normals, and their performance depends on the content of the training dataset. On the other hand, the results show that Nayar *et al.* [68] performs much better on challenging concave shapes. However, it cannot model specularities and cast shadows. On the other hand, our method can model these non-Lambertian effects with the reflectance mapping, and therefore, it performs better than Nayar *et al.* in all the tested categories.

Lastly, we provide the reflectance map obtained using our method on the proposed dataset. Figure A.7 and Figure A.8 show the reflectance map obtained using our method on the synthetic and real sequences respectively.

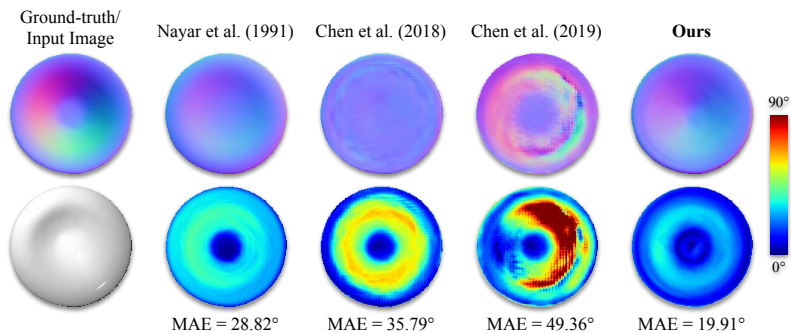


FIGURE A.1: Qualitative comparison on the **Vase** scene. Here, it is obvious that previous deep learning based methods fail to handle the concavity of the subject. In contrast, our method works reasonably well showing the competence of our modeling procedure.

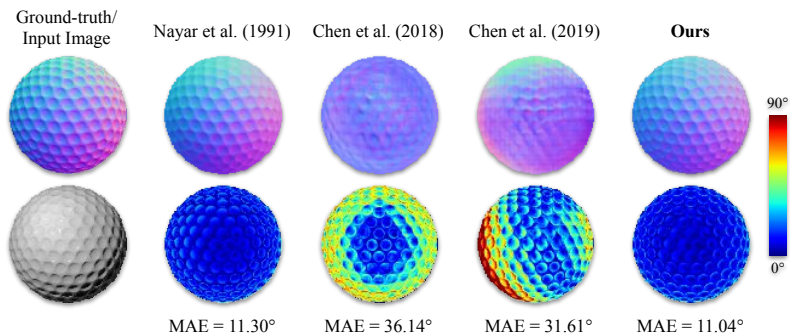


FIGURE A.2: Qualitative comparison on the **Golf-ball** scene. Although deep learning based methods perform well smooth objects, they cannot handle fine structures and indentations.

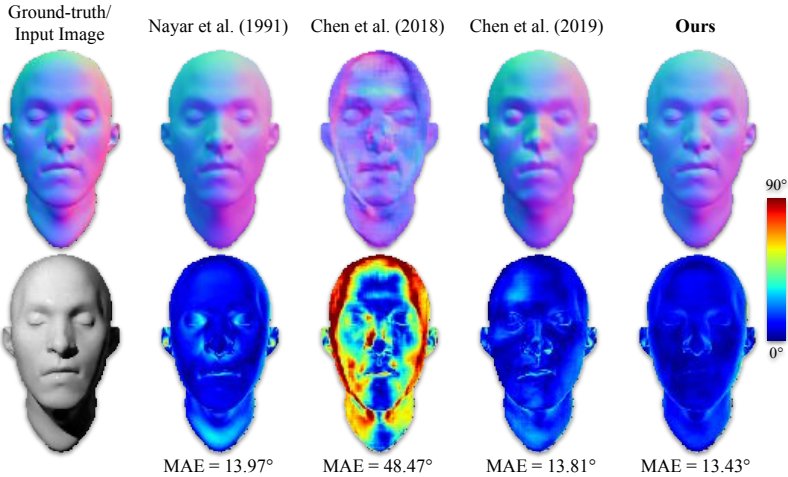


FIGURE A.3: Qualitative comparison on the **Face** scene. Although Nayar *et al.* [68] models interreflections, it cannot handle cast shadows. Therefore, it performs poorly on regions surrounding the eyes and the nose where cast shadows are effective. Here, we also observe that Chen *et al.* [43] cannot estimate accurately for higher slant angles.

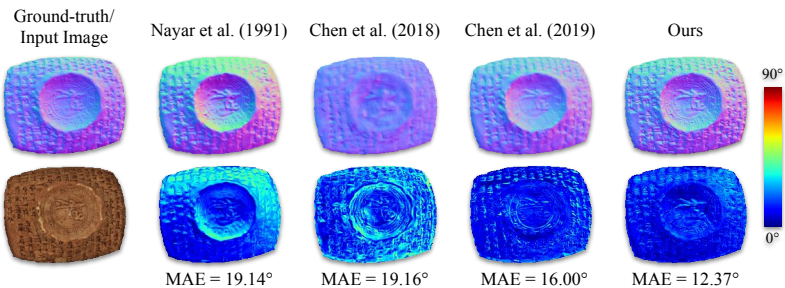


FIGURE A.4: Qualitative comparison on the **Tablet1** scene. This subject has a complicated geometry involving cuneiform and reliefs. Apart from these fine structures, the object can be treated as a composite surface which has a large concavity in the middle part.

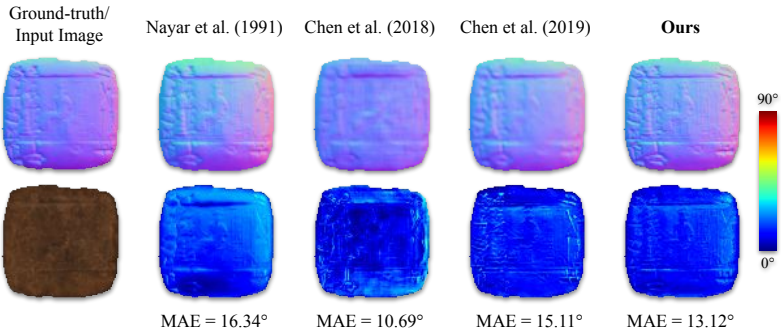


FIGURE A.5: Qualitative comparison on the **Tabletz** scene. Similar to *Tablet1*, this subject also contains reliefs and cuneiform scripts. Since the overall geometry is approximately flat, all methods perform comparable on this category.

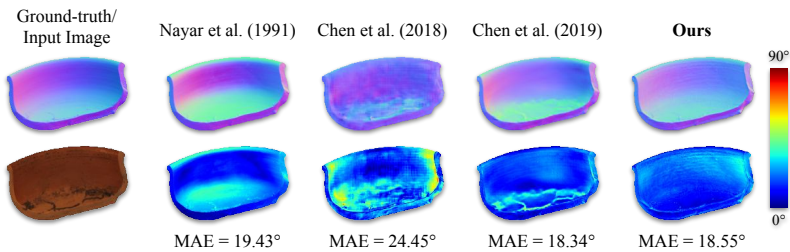


FIGURE A.6: Qualitative comparison on the **Broken Pot** scene.

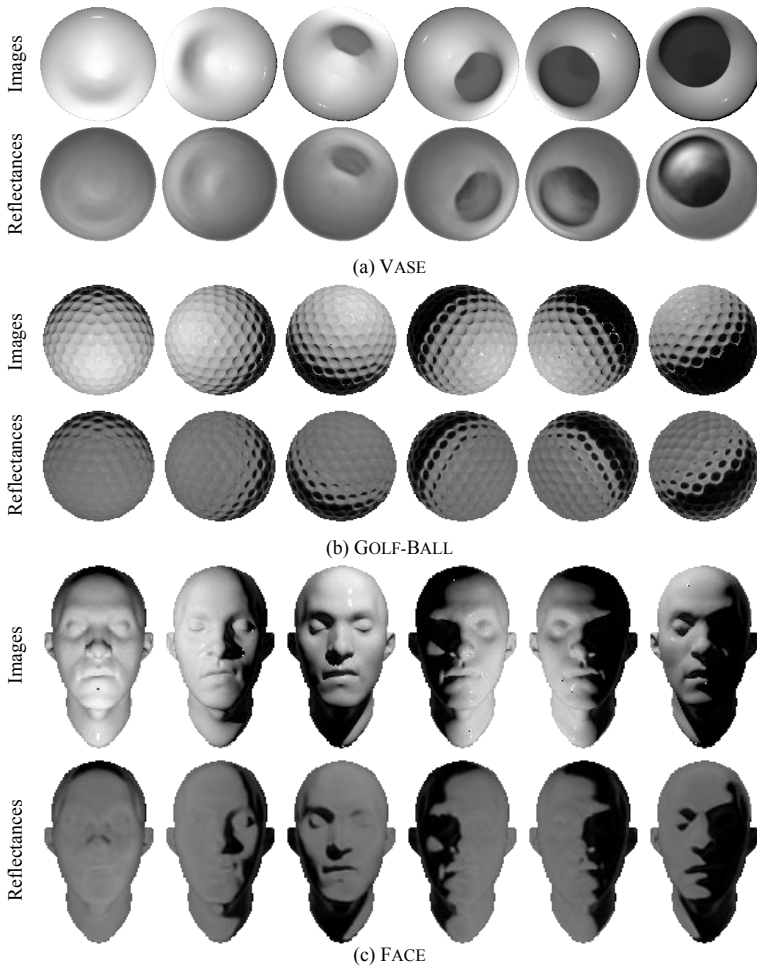


FIGURE A.7: Reflectance maps obtained with our method from **Vase**, **Golf-ball** and **Face** categories.

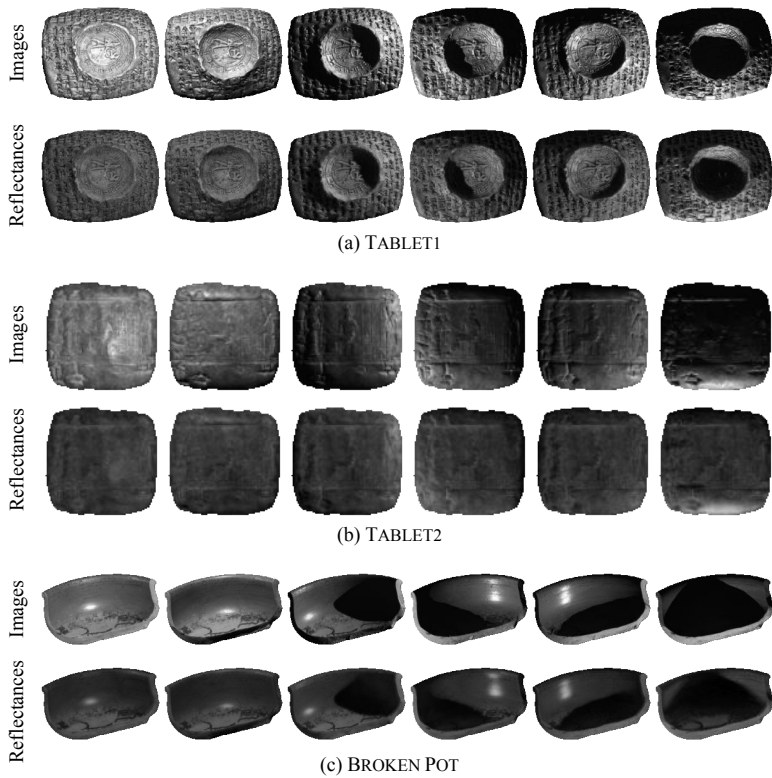


FIGURE A.8: Reflectance maps obtained with our method from **Tablet1**, **Tablet2** and **Broken Pot** categories.

A.2 UNCERTAINTY-AWARE DEEP MVPS

Extending the qualitative analysis in Section §3.5, we demonstrate the quality of the recovered meshes for DiLiGenT-MV objects. Fig.A.9-Fig.A.12 show the colored Wireframe model comparison of the object surface recovered using our approach, B-MVPS [91] and GT. The visualizations show that the distribution of the geometric primitives of B-MVPS [91] is irregular and unevenly distributed. Similarly, Fig.A.13-Fig.A.17 show the quality of the meshes compared to NeRF [88], R-MVPS [108], and B-MVPS [91]. Overall, it can be observed that our method provides surfaces which are superior in quality, regular, hence more useful for geometry processing applications.

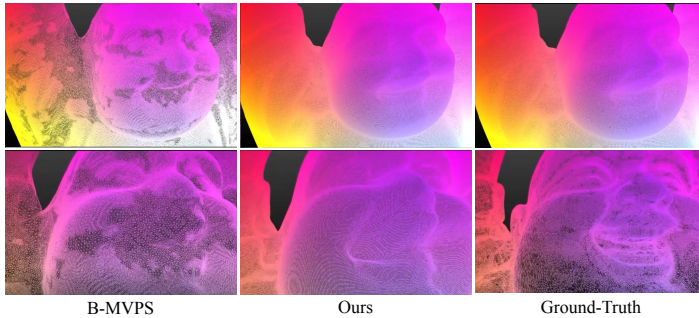


FIGURE A.9: Colored Wireframe qualitative comparison with B-MVPS [91] on BUDDHA.

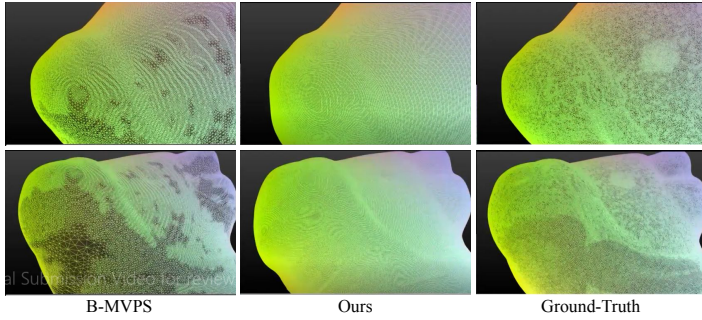


FIGURE A.10: Colored Wireframe qualitative comparison with B-MVPS [91] on COW.

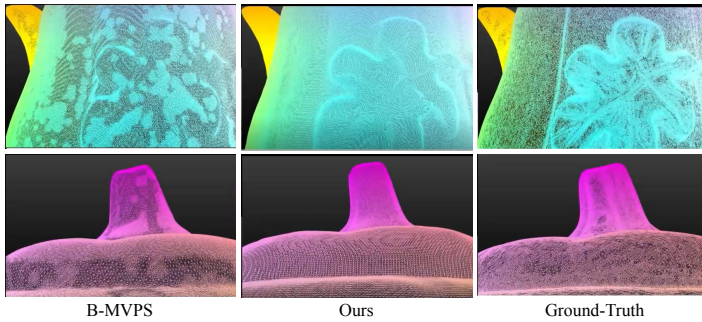


FIGURE A.11: Colored Wireframe qualitative comparison with B-MVPS [91] on POT2.

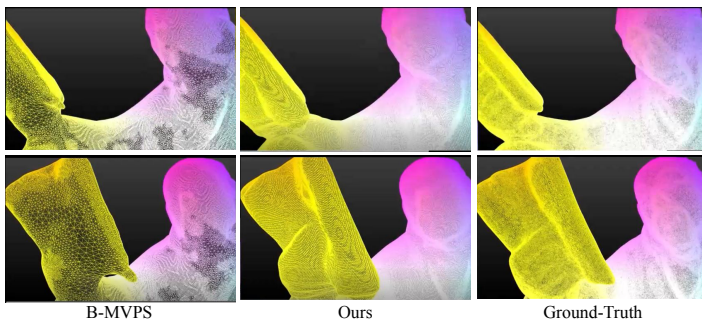


FIGURE A.12: Colored Wireframe qualitative comparison with B-MVPS [91] on READING.

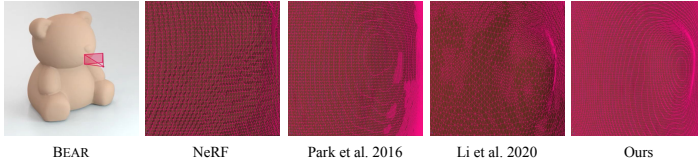


FIGURE A.13: Qualitative mesh comparison on BEAR.

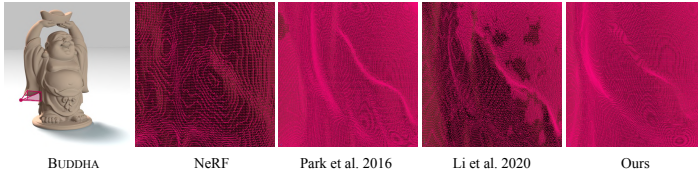


FIGURE A.14: Qualitative mesh comparison on BUDDHA.

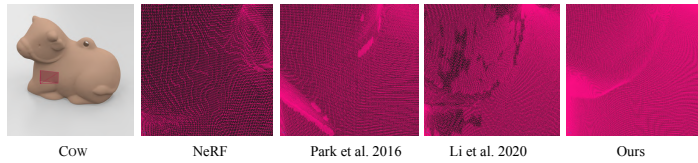


FIGURE A.15: Qualitative mesh comparison on COW.



FIGURE A.16: Qualitative mesh comparison on POT2.

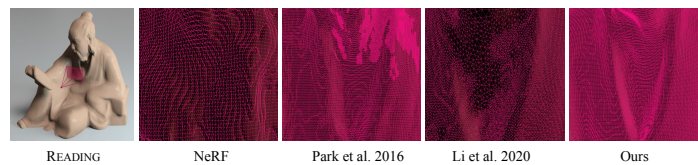


FIGURE A.17: Qualitative mesh comparison on READING.

A.3 UNCERTAINTY-AWARE VOLUME RENDERING FOR MVPS

(a) Image Rendering. In Section §3.6, we used the suitable 3D reconstruction metric to show the high-quality 3D reconstruction achieved using our approach. Nevertheless, we further analyzed our method’s performance using popular image rendering metrics showing our method’s ability to provide favorable surface 3D geometry as a function of volume density. Indeed, volume rendering is inherently conditioned on the implicit geometry, and therefore, obtaining high-quality rendering is an indicator for learning accurate representation of the object. Fig.A.18 shows the rendered images obtained from our approach 3D reconstruction results and their corresponding PSNR scores for quantitative evaluation. The results show that our approach not only reconstructs accurate 3D geometry of the object but also provides high-quality image rendering.

(b) High-Frequency Details. Here, we extend the qualitative comparison provided in Section §3.6 of the thesis by demonstrating the high-frequency details of our 3D reconstructions. Fig.A.19 visually compares the 3D reconstructions obtained by our method and other MVPS methods, focusing on the ear of Buddha object. Clearly, our method recovers fine details that are missing in other methods and provides outstanding 3D reconstructions.











	BEAR	BUDDHA	COW	POT2	READING
Ground-truth Images					
Rendered Images					
PSNR	30.91	30.04	31.61	32.20	29.76

FIGURE A.18: Image rendering quality achieved by using our method’s 3D reconstruction on DiLiGenT-MV dataset [91]. PSNR metric value quantify the rendered image quality (higher is better).

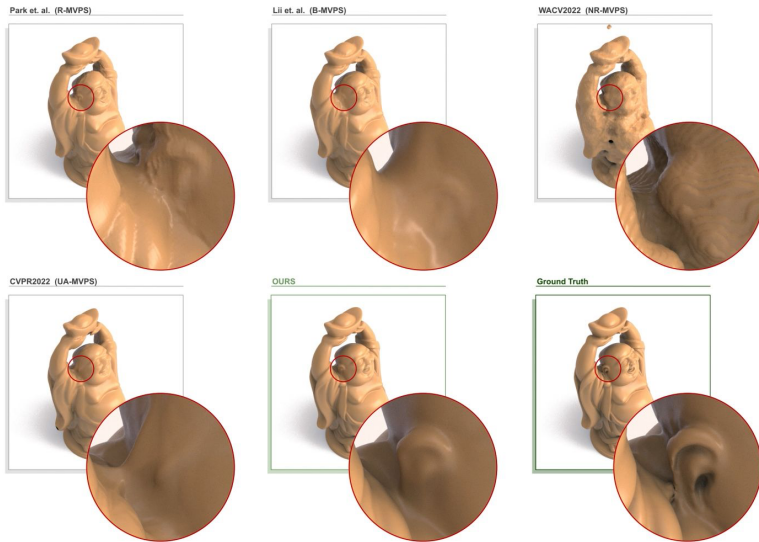


FIGURE A.19: Visual comparison of MVPS reconstructions on Buddha category, demonstrating the benefit of our approach in recovering high-frequency details.

BIBLIOGRAPHY

1. Furukawa, Y. & Hernández, C. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision* **9**, 1 (2015).
2. Bronstein, A. M., Bronstein, M. M. & Kimmel, R. *Numerical geometry of non-rigid shapes* (Springer Science & Business Media, 2008).
3. Moons, T., Van Gool, L. & Vergauwen, M. *3D reconstruction from multiple images: principles* (Now Publishers Inc, 2009).
4. Szeliski, R. *Computer vision: algorithms and applications* (Springer Science & Business Media, 2010).
5. Zollhöfer, M., Stotko, P., Görlitz, A., Theobalt, C., Nießner, M., Klein, R. & Kolb, A. State of the Art on 3D Reconstruction with RGB-D Cameras in *Computer graphics forum* **37** (2018), 625.
6. Schönberger, J. L. & Frahm, J.-M. *Structure-from-Motion Revisited* in *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
7. Hartley, R. & Zisserman, A. *Multiple view geometry in computer vision* (Cambridge university press, 2003).
8. Pentland, A. P. A new sense for depth of field. *IEEE transactions on pattern analysis and machine intelligence*, 523 (1987).
9. Blostein, D. & Ahuja, N. Shape from texture: Integrating texture-element extraction and surface estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**, 1233 (1989).
10. Horn, B. K. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view (1970).
11. Woodham, R. J. Photometric method for determining surface orientation from multiple images. *Optical engineering* **19**, 191139 (1980).
12. Chandraker, M. K., Kahl, F. & Kriegman, D. J. *Reflections on the generalized bas-relief ambiguity* in 2005 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* **1** (2005), 788.
13. Ikehata, S., Wipf, D., Matsushita, Y. & Aizawa, K. *Robust photometric stereo using sparse regression* in 2012 *IEEE Conference on Computer Vision and Pattern Recognition* (2012), 318.
14. Ikehata, S. *CNN-PS: CNN-based photometric stereo for general non-convex surfaces* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), 3.
15. Taniai, T. & Maehara, T. *Neural Inverse Rendering for General Reflectance Photometric Stereo* in *International Conference on Machine Learning (ICML)* (2018), 4857.
16. Chen, G., Waechter, M., Shi, B., Wong, K.-Y. K. & Matsushita, Y. *What is Learned in Deep Uncalibrated Photometric Stereo?* in *European Conference on Computer Vision* (2020).
17. Chen, G., Han, K., Shi, B., Matsushita, Y. & Wong, K.-Y. K. *Deep photometric stereo for non-Lambertian surfaces*. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

18. Wu, L., Ganesh, A., Shi, B., Matsushita, Y., Wang, Y. & Ma, Y. *Robust photometric stereo via low-rank matrix completion and recovery in Asian Conference on Computer Vision* (2010), 703.
19. Ikehata, S., Wipf, D., Matsushita, Y. & Aizawa, K. Photometric stereo using sparse Bayesian regression for general diffuse surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**, 1816 (2014).
20. Oh, T.-H., Kim, H., Tai, Y.-W., Bazin, J.-C. & So Kweon, I. *Partial sum minimization of singular values in RPCA for low-level vision in Proceedings of the IEEE international conference on computer vision* (2013), 145.
21. Mukaigawa, Y., Ishii, Y. & Shakunaga, T. Analysis of photometric factors based on photometric linearization. *JOSA A* **24**, 3326 (2007).
22. Wu, T.-P. & Tang, C.-K. Photometric stereo via expectation maximization. *IEEE transactions on pattern analysis and machine intelligence* **32**, 546 (2009).
23. Georghiades, A. S. *Incorporating the torrance and sparrow model of reflectance in uncalibrated photometric stereo in Computer Vision, IEEE International Conference on* **3** (2003), 816.
24. Chung, H.-S. & Jia, J. *Efficient photometric stereo on glossy surfaces with wide specular lobes in 2008 IEEE Conference on Computer Vision and Pattern Recognition* (2008), 1.
25. Goldman, D. B., Curless, B., Hertzmann, A. & Seitz, S. M. Shape and spatially-varying brdfs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**, 1060 (2009).
26. Belhumeur, P. N., Kriegman, D. J. & Yuille, A. L. The bas-relief ambiguity. *International journal of computer vision* **35**, 33 (1999).
27. Alldrin, N. G., Mallick, S. P. & Kriegman, D. J. *Resolving the generalized bas-relief ambiguity by entropy minimization in 2007 IEEE conference on computer vision and pattern recognition* (2007), 1.
28. Shi, B., Matsushita, Y., Wei, Y., Xu, C. & Tan, P. *Self-calibrating photometric stereo in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2010), 1118.
29. Papadimitri, T. & Favaro, P. A closed-form, consistent and robust solution to uncalibrated photometric stereo via local diffuse reflectance maxima. *International journal of computer vision* **107**, 139 (2014).
30. Papadimitri, T. & Favaro, P. *A new perspective on uncalibrated photometric stereo in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), 1474.
31. Drbohlav, O. & Chaniler, M. *Can two specular pixels calibrate photometric stereo? in Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 2* (2005), 1850.
32. Sengupta, S., Zhou, H., Forkel, W., Basri, R., Goldstein, T. & Jacobs, D. Solving uncalibrated photometric stereo using fewer images by jointly optimizing low-rank matrix completion and integrability. *Journal of Mathematical Imaging and Vision* **60**, 563 (2018).
33. Tan, P., Mallick, S. P., Quan, L., Kriegman, D. J. & Zickler, T. *Isotropy, reciprocity and the generalized bas-relief ambiguity in 2007 IEEE Conference on Computer Vision and Pattern Recognition* (2007), 1.

34. Wu, Z. & Tan, P. *Calibrating photometric stereo by holistic reflectance symmetry analysis in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), 1498.
35. Lu, F., Chen, X., Sato, I. & Sato, Y. SymPS: BRDF symmetry guided photometric stereo for shape and light source estimation. *IEEE transactions on pattern analysis and machine intelligence* **40**, 221 (2017).
36. Basri, R., Jacobs, D. & Kemelmacher, I. Photometric stereo with general, unknown lighting. *International Journal of computer vision* **72**, 239 (2007).
37. Simakov, D., Frolova, D. & Basri, R. *Dense Shape Reconstruction of a Moving Object under Arbitrary, Unknown Lighting*. in *ICCV* **3** (2003), 1202.
38. Chen, C.-P. & Chen, C.-S. *The 4-source photometric stereo under general unknown lighting in European Conference on Computer Vision* (2006), 72.
39. Shen, L. & Tan, P. *Photometric stereo and weather estimation using internet images in 2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), 1850.
40. Haefner, B., Ye, Z., Gao, M., Wu, T., Quéau, Y. & Cremers, D. *Variational uncalibrated photometric stereo under general lighting in Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), 8539.
41. Brahimi, M., Quéau, Y., Haefner, B. & Cremers, D. *On the well-posedness of uncalibrated photometric stereo under general lighting in Advances in Photometric 3D-Reconstruction* (Springer, 2020), 147.
42. Santo, H., Samejima, M., Sugano, Y., Shi, B. & Matsushita, Y. *Deep photometric stereo network in Proceedings of the IEEE International Conference on Computer Vision Workshops* (2017), 501.
43. Chen, G., Han, K. & Wong, K.-Y. K. *PS-FCN: A flexible learning framework for photometric stereo in Proceedings of the European Conference on Computer Vision (ECCV)* (2018), 3.
44. Li, J. & Li, H. *Neural Reflectance for Shape Recovery with Shadow Handling in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 16221.
45. Liu, H., Yan, Y., Song, K. & Yu, H. *SPS-Net: Self-attention photometric stereo network. IEEE Transactions on Instrumentation and Measurement* **70**, 1 (2020).
46. Ikehata, S. *Ps-transformer: Learning sparse photometric stereo network using self-attention mechanism. arXiv preprint arXiv:2211.11386* (2022).
47. Ju, Y., Shi, B., Jian, M., Qi, L., Dong, J. & Lam, K.-M. *NormAttention-PSN: A High-frequency Region Enhanced Photometric Stereo Network with Normalized Attention. International Journal of Computer Vision* **130**, 3014 (2022).
48. Yao, Z., Li, K., Fu, Y., Hu, H. & Shi, B. *Gps-net: Graph-based photometric stereo network. Advances in Neural Information Processing Systems* **33**, 10306 (2020).
49. Logothetis, F., Budvytis, I., Mecca, R. & Cipolla, R. *Px-net: Simple and efficient pixel-wise training of photometric stereo networks in Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 12757.
50. Logothetis, F., Budvytis, I., Mecca, R. & Cipolla, R. *A CNN Based Approach for the Near-Field Photometric Stereo Problem. arXiv preprint arXiv:2009.05792* (2020).
51. Hiroaki, S., Waechter, M. & Matsushita, Y. *Deep near-light photometric stereo for spatially varying reflectances in European Conference on Computer Vision* (2020).

52. Zheng, Q., Jia, Y., Shi, B., Jiang, X., Duan, L.-Y. & Kot, A. C. *SPLINE-Net: Sparse photometric stereo through lighting interpolation and normal estimation networks* in *Proceedings of the IEEE International Conference on Computer Vision* (2019), 8549.
53. Li, J., Robles-Kelly, A., You, S. & Matsushita, Y. *Learning to minify photometric stereo* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), 7568.
54. Chen, G., Han, K., Shi, B., Matsushita, Y. & Wong, K.-Y. K. *Self-calibrating deep photometric stereo networks* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), 8739.
55. Ikehata, S. *Universal Photometric Stereo Network using Global Lighting Contexts* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 12591.
56. Hernandez, C., Vogiatzis, G. & Cipolla, R. *Multiview photometric stereo*. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**, 548 (2008).
57. Nehab, D., Rusinkiewicz, S., Davis, J. & Ramamoorthi, R. *Efficiently Combining Positions and Normals for Precise 3D Geometry*. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2005)* **24**, 536 (2005).
58. Kaya, B., Kumar, S., Oliveira, C., Ferrari, V. & Van Gool, L. *Uncertainty-Aware Deep Multi-View Photometric Stereo* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 12601.
59. Chatterjee, A. & Govindu, V. M. *Efficient and robust large-scale rotation averaging* in *Proceedings of the IEEE International Conference on Computer Vision* (2013), 521.
60. Sandström, E., Oswald, M. R., Kumar, S., Weder, S., Yu, F., Sminchisescu, C. & Van Gool, L. *Learning Online Multi-Sensor Depth Fusion*. *arXiv preprint arXiv:2204.03353* (2022).
61. Sarno, F., Kumar, S., Kaya, B., Huang, Z., Ferrari, V. & Van Gool, L. *Neural Architecture Search for Efficient Uncalibrated Deep Photometric Stereo* in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2022), 361.
62. Park, J., Sinha, S. N., Matsushita, Y., Tai, Y.-W. & So Kweon, I. *Multiview photometric stereo using planar mesh parameterization* in *Proceedings of the IEEE International Conference on Computer Vision* (2013), 1161.
63. Li, Z., Gogia, P. C. & Kaess, M. *Dense surface reconstruction from monocular vision and LiDAR* in *2019 International Conference on Robotics and Automation (ICRA)* (2019), 6905.
64. Quéau, Y., Wu, T., Lauze, F., Durou, J.-D. & Cremers, D. *A non-convex variational approach to photometric stereo under inaccurate lighting* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), 99.
65. Alldrin, N., Zickler, T. & Kriegman, D. *Photometric stereo with non-parametric and spatially-varying reflectance* in *2008 IEEE Conference on Computer Vision and Pattern Recognition* (2008), 1.
66. Higo, T., Matsushita, Y. & Ikeuchi, K. *Consensus photometric stereo* in *2010 IEEE computer society conference on computer vision and pattern recognition* (2010), 1157.
67. Logothetis, F., Budvytis, I., Mecca, R. & Cipolla, R. *PX-NET: Simple, Efficient Pixel-Wise Training of Photometric Stereo Networks*. *arXiv preprint arXiv:2008.04933* (2020).
68. Nayar, S. K., Ikeuchi, K. & Kanade, T. *Shape from interreflections*. *International Journal of Computer Vision* **6**, 173 (1991).

69. Xu, B., Wang, N., Chen, T. & Li, M. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853* (2015).
70. Crane, K. *Conformal Geometry Processing* PhD thesis (Caltech, 2013).
71. Ikehata, S. & Aizawa, K. Photometric stereo using constrained bivariate regression for general isotropic surfaces in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), 2179.
72. Shi, B., Tan, P., Matsushita, Y. & Ikeuchi, K. Bi-polynomial modeling of low-frequency reflectances. *IEEE transactions on pattern analysis and machine intelligence* **36**, 1078 (2013).
73. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).
74. Antensteiner, D., Štolc, S. & Pock, T. A review of depth and normal fusion algorithms. *Sensors* **18**, 431 (2018).
75. Boyd, S., Parikh, N., Chu, E., Peleato, B. & Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* **3**, 1 (2011).
76. Lin, Z., Chen, M. & Ma, Y. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055* (2010).
77. Hale, E. T., Yin, W. & Zhang, Y. Fixed-point continuation for l_1 -minimization: Methodology and convergence. *SIAM Journal on Optimization* **19**, 1107 (2008).
78. Shi, B., Wu, Z., Mo, Z., Duan, D., Yeung, S.-K. & Tan, P. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), 3707.
79. Johnson, M. K. & Adelson, E. H. Shape estimation in natural illumination in *CVPR 2011* (2011), 2553.
80. Wiles, O. & Zisserman, A. Silnet: Single-and multi-view reconstruction by learning from silhouettes. *arXiv preprint arXiv:1711.07888* (2017).
81. Matusik, W. *A data-driven reflectance model* PhD thesis (Massachusetts Institute of Technology, 2003).
82. Lu, F., Matsushita, Y., Sato, I., Okabe, T. & Sato, Y. Uncalibrated photometric stereo for unknown isotropic reflectances in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), 1490.
83. Blinn, J. F. Models of light reflection for computer synthesized pictures in *Proceedings of the 4th annual conference on Computer graphics and interactive techniques* (1977), 192.
84. Menini, D., Kumar, S., Oswald, M. R., Sandstrom, E., Sminchisescu, C. & Van Gool, L. A Real-Time Online Learning Framework for Joint 3D Reconstruction and Semantic Segmentation of Indoor Scenes. *arXiv preprint arXiv:2108.05246* (2021).
85. Kaya, B., Kumar, S., Oliveira, C., Ferrari, V. & Van Gool, L. Uncalibrated Neural Inverse Rendering for Photometric Stereo of General Surfaces in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), 3804.
86. Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S. & Fitzgibbon, A. Kinectfusion: Real-time dense surface mapping and tracking in 2011 10th IEEE international symposium on mixed and augmented reality (2011), 127.

87. Wolff, L. B. Polarization vision: a new sensory approach to image understanding. *Image and Vision computing* **15**, 81 (1997).
88. Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R. & Ng, R. *Nerf: Representing scenes as neural radiance fields for view synthesis* in *European conference on computer vision* (2020), 405.
89. Kumar, S., Dai, Y. & Li, H. Superpixel soup: Monocular dense 3d reconstruction of a complex dynamic scene. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).
90. Kumar, S., Dai, Y. & Li, H. *Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames* in *Proceedings of the IEEE International Conference on Computer Vision* (2017), 4649.
91. Li, M., Zhou, Z., Wu, Z., Shi, B., Diao, C. & Tan, P. Multi-View Photometric Stereo: A Robust Solution and Benchmark Dataset for Spatially Varying Isotropic Materials. *IEEE Transactions on Image Processing* **29**, 4159 (2020).
92. Wu, C. *et al.* VisualSfM: A visual structure from motion system (2011).
93. Furukawa, Y. & Ponce, J. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence* **32**, 1362 (2009).
94. Horn, B. K. & Brooks, M. J. The variational approach to shape from shading. *Computer Vision, Graphics, and Image Processing* **33**, 174 (1986).
95. Xie, W., Wang, M., Wei, M., Jiang, J. & Qin, J. *Surface reconstruction from normals: A robust dgp-based discontinuity preservation approach* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), 5328.
96. Quéau, Y., Durou, J.-D. & Aujol, J.-F. Variational methods for normal integration. *Journal of Mathematical Imaging and Vision* **60**, 609 (2018).
97. Schonberger, J. L. & Frahm, J.-M. *Structure-from-motion revisited* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), 4104.
98. Galliani, S., Lasinger, K. & Schindler, K. *Massively parallel multiview stereopsis by surface normal diffusion* in *Proceedings of the IEEE International Conference on Computer Vision* (2015), 873.
99. Mostafa, M.-H., Yamany, S. M. & Farag, A. A. *Integrating shape from shading and range data using neural networks* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* **2** (1999), 15.
100. Lange, H. *Advances in the cooperation of shape from shading and stereo vision* in *Second International Conference on 3-D Digital Imaging and Modeling* (Cat. No. PR0062) (1999), 46.
101. Chatterjee, A. & Madhav Govindu, V. *Photometric refinement of depth maps for multi-albedo objects* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), 933.
102. Geng, J. Structured-light 3D surface imaging: a tutorial. *Advances in Optics and Photonics* **3**, 128 (2011).
103. Zhang, L., Curless, B. & Seitz, S. M. *Rapid shape acquisition using color structured light and multi-pass dynamic programming* in *Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission* (2002), 24.

104. Zhang, L., Curless, B. & Seitz, S. M. *Spacetime stereo: Shape recovery for dynamic scenes in 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* 2 (2003), II.
105. Davis, J., Ramamoorthi, R. & Rusinkiewicz, S. *Spacetime stereo: A unifying framework for depth from triangulation in 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* 2 (2003), II.
106. Ren, J., Jian, Z., Wang, X., Mingjun, R., Zhu, L. & Jiang, X. Complex Surface Reconstruction Based on Fusion of Surface Normals and Sparse Depth Measurement. *IEEE Transactions on Instrumentation and Measurement* 70, 1 (2021).
107. Bylow, E., Maier, R., Kahl, F. & Olsson, C. *Combining depth fusion and photometric stereo for fine-detailed 3d models in Scandinavian Conference on Image Analysis* (2019), 261.
108. Park, J., Sinha, S. N., Matsushita, Y., Tai, Y.-W. & Kweon, I. S. Robust multiview photometric stereo using planar mesh parameterization. *IEEE transactions on pattern analysis and machine intelligence* 39, 1591 (2016).
109. Yu, L.-F., Yeung, S.-K., Tai, Y.-W. & Lin, S. *Shading-based shape refinement of RGB-D images in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), 1415.
110. Kutulakos, K. N. & Seitz, S. M. A theory of shape by space carving. *International journal of computer vision* 38, 199 (2000).
111. Bleyer, M., Rhemann, C. & Rother, C. *PatchMatch Stereo-Stereo Matching with Slanted Support Windows.* in *Bmvc* 11 (2011), 1.
112. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F. & Tan, P. *Cascade cost volume for high-resolution multi-view stereo and stereo matching in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 2495.
113. Ji, M., Gall, J., Zheng, H., Liu, Y. & Fang, L. *Surfacenet: An end-to-end 3d neural network for multiview stereopsis in Proceedings of the IEEE International Conference on Computer Vision* (2017), 2307.
114. Kar, A., Häne, C. & Malik, J. Learning a multi-view stereo machine. *arXiv preprint arXiv:1708.05375* (2017).
115. Seitz, S. M. & Dyer, C. R. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision* 35, 151 (1999).
116. Faugeras, O. & Keriven, R. *Variational principles, surface evolution, PDE's, level set methods and the stereo problem* (IEEE, 2002).
117. Vogiatzis, G., Torr, P. H. & Cipolla, R. *Multi-view stereo via volumetric graph-cuts in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* 2 (2005), 391.
118. Barnes, C., Shechtman, E., Finkelstein, A. & Goldman, D. B. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* 28, 24 (2009).
119. Locher, A., Perdoch, M. & Van Gool, L. *Progressive prioritized multi-view stereo in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), 3244.
120. Goesele, M., Curless, B. & Seitz, S. M. *Multi-view stereo revisited in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* 2 (2006), 2402.

121. Strecha, C., Fransens, R. & Van Gool, L. *Combined depth and outlier estimation in multi-view stereo in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* 2 (2006), 2394.
122. Campbell, N. D., Vogiatzis, G., Hernández, C. & Cipolla, R. *Using multiple hypotheses to improve depth-maps for multi-view stereo in European Conference on Computer Vision* (2008), 766.
123. Schönberger, J. L., Zheng, E., Frahm, J.-M. & Pollefeys, M. *Pixelwise view selection for unstructured multi-view stereo in European Conference on Computer Vision* (2016), 501.
124. Xu, Q. & Tao, W. *Multi-scale geometric consistency guided multi-view stereo in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), 5483.
125. Lhuillier, M. & Quan, L. *A quasi-dense approach to surface reconstruction from uncalibrated images. IEEE transactions on pattern analysis and machine intelligence* 27, 418 (2005).
126. Zhang, R., Zhu, S., Fang, T. & Quan, L. *Distributed Very Large Scale Bundle Adjustment by Global Camera Consensus in Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2017).
127. Zhu, S., Zhang, R., Zhou, L., Shen, T., Fang, T., Tan, P. & Quan, L. *Very large-scale global sfm by distributed motion averaging in Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), 4568.
128. Zbontar, J., LeCun, Y., *et al.* *Stereo matching by training a convolutional neural network to compare image patches. J. Mach. Learn. Res.* 17, 2287 (2016).
129. Hartmann, W., Galliani, S., Havlena, M., Van Gool, L. & Schindler, K. *Learned multi-patch similarity in Proceedings of the IEEE International Conference on Computer Vision* (2017), 1586.
130. Chen, R., Han, S., Xu, J. & Su, H. *Point-based multi-view stereo network in Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), 1538.
131. Hou, Y., Kannala, J. & Solin, A. *Multi-view stereo by temporal nonparametric fusion in Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), 2651.
132. Xue, Y., Chen, J., Wan, W., Huang, Y., Yu, C., Li, T. & Bao, J. *Mvsrnf: Learning multi-view stereo with conditional random fields in Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), 4312.
133. Im, S., Jeon, H.-G., Lin, S. & Kweon, I. S. *DPSNet: End-to-end Deep Plane Sweep Stereo in International Conference on Learning Representations* (2018).
134. Luo, K., Guan, T., Ju, L., Huang, H. & Luo, Y. *P-mvsnet: Learning patch-wise matching confidence aggregation for multi-view stereo in Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), 10452.
135. Yao, Y., Luo, Z., Li, S., Fang, T. & Quan, L. *Mvsnet: Depth inference for unstructured multi-view stereo in Proceedings of the European Conference on Computer Vision (ECCV)* (2018), 767.
136. Xu, Q. & Tao, W. *Learning inverse depth regression for multi-view stereo with correlation cost volume in Proceedings of the AAAI Conference on Artificial Intelligence* 34 (2020), 12508.
137. Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T. & Quan, L. *Recurrent mvsnet for high-resolution multi-view stereo depth inference in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), 5525.

138. Cheng, S., Xu, Z., Zhu, S., Li, Z., Li, L. E., Ramamoorthi, R. & Su, H. *Deep stereo using adaptive thin volume representation with uncertainty awareness in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 2524.
139. Xu, Q. & Tao, W. PVSNet: Pixelwise Visibility-Aware Multi-View Stereo Network. *arXiv preprint arXiv:2007.07714* (2020).
140. Wang, F., Galliani, S., Vogel, C., Speciale, P. & Pollefeys, M. PatchmatchNet: Learned Multi-View Patchmatch Stereo. *arXiv preprint arXiv:2012.01411* (2020).
141. Yu, Z. & Gao, S. *Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 1949.
142. Yu, A., Ye, V., Tancik, M. & Kanazawa, A. pixelNeRF: Neural Radiance Fields from One or Few Images. *arXiv preprint arXiv:2012.02190* (2020).
143. Yariv, L., Gu, J., Kasten, Y. & Lipman, Y. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems* **34** (2021).
144. Zhang, K., Luan, F., Li, Z. & Snavely, N. IRON: Inverse Rendering by Optimizing Neural SDFs and Materials from Photometric Images in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 5565.
145. Martin-Brualla, R., Radwan, N., Sajjadi, M. S., Barron, J. T., Dosovitskiy, A. & Duckworth, D. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *arXiv preprint arXiv:2008.02268* (2020).
146. Neff, T., Stadlbauer, P., Parger, M., Kurz, A., Chaitanya, C. R. A., Kaplanyan, A. & Steinberger, M. DONeRF: Towards Real-Time Rendering of Neural Radiance Fields using Depth Oracle Networks. *arXiv preprint arXiv:2103.03231* (2021).
147. Pumarola, A., Corona, E., Pons-Moll, G. & Moreno-Noguer, F. D-NeRF: Neural Radiance Fields for Dynamic Scenes. *arXiv preprint arXiv:2011.13961* (2020).
148. Srinivasan, P. P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B. & Barron, J. T. NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis. *arXiv preprint arXiv:2012.03927* (2020).
149. Yu, A., Li, R., Tancik, M., Li, H., Ng, R. & Kanazawa, A. PlenOctrees for Real-time Rendering of Neural Radiance Fields. *arXiv preprint arXiv:2103.14024* (2021).
150. Rematas, K., Martin-Brualla, R. & Ferrari, V. *Sharf: Shape-conditioned Radiance Fields from a Single View in ICML* (2021).
151. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J. & Su, H. MVSNeRF: Fast Generalizable Radiance Field Reconstruction from Multi-View Stereo. *arXiv preprint arXiv:2103.15595* (2021).
152. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B. & Lipman, Y. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. *Advances in Neural Information Processing Systems* **33** (2020).
153. Logothetis, F., Mecca, R. & Cipolla, R. *A differential volumetric approach to multi-view photometric stereo in Proceedings of the IEEE International Conference on Computer Vision* (2019), 1052.
154. Wan, E. A. & Nelson, A. T. Dual extended Kalman filter methods. *Kalman filtering and neural networks* **123** (2001).

155. Kazhdan, M., Bolitho, M. & Hoppe, H. *Poisson surface reconstruction* in *Proceedings of the fourth Eurographics symposium on Geometry processing* 7 (2006).
156. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
157. Cheng, Z., Li, H., Asano, Y., Zheng, Y. & Sato, I. *Multi-view 3D Reconstruction of a Texture-less Smooth Surface of Unknown Generic Reflectance* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), 16226.
158. Cheng, Z., Li, H., Hartley, R., Zheng, Y. & Sato, I. One Ring to Rule Them All: a simple solution to multi-view 3D-Reconstruction of shapes with unknown BRDF via a small Recurrent ResNet. *arXiv preprint arXiv:2104.05014* (2021).
159. Yang, W., Chen, G., Chen, C., Chen, Z. & Wong, K.-Y. K. *Ps-nerf: Neural inverse rendering for multi-view photometric stereo* in *European Conference on Computer Vision* (2022), 266.
160. Maier, R., Kim, K., Cremers, D., Kautz, J. & Nießner, M. *Intrinsic3d: High-quality 3D reconstruction by joint appearance and geometry optimization with spatially-varying lighting* in *Proceedings of the IEEE international conference on computer vision* (2017), 3114.
161. Zollhöfer, M., Dai, A., Innmann, M., Wu, C., Stamminger, M., Theobalt, C. & Nießner, M. Shading-based refinement on volumetric signed distance functions. *ACM Transactions on Graphics (TOG)* 34, 1 (2015).
162. Liang, Z., Xu, C., Hu, J., Li, Y. & Meng, Z. Better together: shading cues and multi-view stereo for reconstruction depth optimization. *IEEE Access* 8, 112348 (2020).
163. Shi, B., Mo, Z., Wu, Z., Duan, D., Yeung, S. & Tan, P. A Benchmark Dataset and Evaluation for Non-Lambertian and Uncalibrated Photometric Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 271 (2019).
164. Levoy, M. Display of surfaces from volume data. *IEEE Computer graphics and Applications* 8, 29 (1988).
165. Jönsson, D., Sundén, E., Ynnerman, A. & Ropinski, T. *A survey of volumetric illumination techniques for interactive volume rendering* in *Computer Graphics Forum* 33 (2014), 27.
166. Pfister, H. Hardware-Accelerated. *Visualization Handbook* 2 (2005).
167. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y. & Courville, A. *On the spectral bias of neural networks* in *International Conference on Machine Learning* (2019), 5301.
168. Kajiya, J. T. *The rendering equation* in *Proceedings of the 13th annual conference on Computer graphics and interactive techniques* (1986), 143.
169. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. *Densely connected convolutional networks* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 4700.
170. Kajiya, J. T. & Von Herzen, B. P. Ray tracing volume densities. *ACM SIGGRAPH computer graphics* 18, 165 (1984).
171. Max, N. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 1, 99 (1995).
172. Lorensen, W. E. & Cline, H. E. Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph computer graphics* 21, 163 (1987).

173. Aanaes, H., Jensen, R. R., Vogiatzis, G., Tola, E. & Dahl, A. B. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision* **120**, 153 (2016).
174. Alldrin, N. G. & Kriegman, D. J. *Toward reconstructing surfaces with arbitrary isotropic reflectance: A stratified photometric stereo approach* in *2007 IEEE 11th International Conference on Computer Vision* (2007), 1.
175. Zhang, R., Isola, P., Efros, A. A., Shechtman, E. & Wang, O. *The unreasonable effectiveness of deep features as a perceptual metric* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), 586.
176. Wang, F., Galliani, S., Vogel, C., Speciale, P. & Pollefeys, M. *PatchmatchNet: Learned Multi-View Patchmatch Stereo* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), 14194.
177. Crandall, M. G. & Lions, P.-L. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American mathematical society* **277**, 1 (1983).
178. MacKay, D. J. A practical Bayesian framework for backpropagation networks. *Neural computation* **4**, 448 (1992).
179. Neal, R. M. *Bayesian learning for neural networks* (Springer Science & Business Media, 2012).
180. Polson, N. G. & Sokolov, V. Deep learning: A Bayesian perspective. *Bayesian Analysis* **12**, 1275 (2017).
181. Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., *et al.* A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342* (2021).
182. Gal, Y. & Ghahramani, Z. *Dropout as a bayesian approximation: Representing model uncertainty in deep learning* in *international conference on machine learning* (2016), 1050.
183. Gal, Y. & Ghahramani, Z. Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158* (2015).
184. Huang, P.-H., Matzen, K., Kopf, J., Ahuja, N. & Huang, J.-B. *Deepmos: Learning multi-view stereopsis* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 2821.
185. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B. & Belongie, S. *Feature pyramid networks for object detection* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 2117.
186. Xu, H. & Zhang, J. *Aanet: Adaptive aggregation network for efficient stereo matching* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 1959.
187. Guo, X., Yang, K., Yang, W., Wang, X. & Li, H. *Group-wise correlation stereo network* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), 3273.
188. Hui, T.-W., Loy, C. C. & Tang, X. *Depth map super-resolution by deep multi-scale guidance* in *European conference on computer vision* (2016), 353.
189. Curless, B. & Levoy, M. *A volumetric method for building complex models from range images* in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), 303.

190. Dai, A., Ruizhongtai Qi, C. & Nießner, M. *Shape completion using 3d-encoder-predictor cnns and shape synthesis in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), 5868.
191. Jiang, Y., Ji, D., Han, Z. & Zwicker, M. *Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 1251.
192. Michalkiewicz, M., Pontes, J. K., Jack, D., Baktashmotlagh, M. & Eriksson, A. Deep level sets: Implicit surface representations for 3d shape inference. *arXiv preprint arXiv:1901.06802* (2019).
193. Horn, B. *Shape Form Shading* (MIT Press, 1989).
194. Gropp, A., Yariv, L., Haim, N., Atzmon, M. & Lipman, Y. in *Proceedings of Machine Learning and Systems 2020* 3569 (2020).
195. Kaya, B., Kumar, S., Sarno, F., Ferrari, V. & Van Gool, L. *Neural Radiance Fields Approach to Deep Multi-View Photometric Stereo in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2022), 1965.
196. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. & Lerer, A. Automatic differentiation in pytorch (2017).
197. Knapitsch, A., Park, J., Zhou, Q.-Y. & Koltun, V. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)* **36**, 1 (2017).
198. Bi, S., Xu, Z., Sunkavalli, K., Kriegman, D. & Ramamoorthi, R. *Deep 3D Capture: Geometry and Reflectance from Sparse Multi-View Images in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 5960.
199. He, K., Zhang, X., Ren, S. & Sun, J. *Identity Mappings in Deep Residual Networks* 2016.
200. Tan, M., Pang, R. & Le, Q. V. *Efficientdet: Scalable and efficient object detection in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), 10781.
201. Xu, H., Yao, L., Zhang, W., Liang, X. & Li, Z. *Auto-fpn: Automatic network architecture adaptation for object detection beyond classification in Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), 6649.
202. Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., Tian, Y., Vajda, P., Jia, Y. & Keutzer, K. *Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), 10734.
203. Wu, Y., Huang, Z., Kumar, S., Sukthanker, R. S., Timofte, R. & Van Gool, L. Trilevel Neural Architecture Search for Efficient Single Image Super-Resolution. *arXiv preprint arXiv:2101.06658* (2021).
204. Sukthanker, R. S., Huang, Z., Kumar, S., Endsjo, E. G., Wu, Y. & Gool, L. V. *Neural Architecture Search of SPD Manifold Networks* 2020.
205. Liu, C., Chen, L.-C., Schroff, F., Adam, H., Hua, W., Yuille, A. L. & Fei-Fei, L. *Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation in Proceedings of the IEEE conference on computer vision and pattern recognition* (2019), 82.
206. Real, E., Aggarwal, A., Huang, Y. & Le, Q. V. *Regularized Evolution for Image Classifier Architecture Search* 2019.
207. Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y. L., Tan, J., Le, Q. & Kurakin, A. Large-scale evolution of image classifiers. *arXiv preprint arXiv:1703.01041* (2017).

208. Zoph, B. & Le, Q. V. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578* (2016).
209. Zoph, B., Vasudevan, V., Shlens, J. & Le, Q. V. *Learning transferable architectures for scalable image recognition in Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (2018), 8697.
210. Liu, H., Simonyan, K. & Yang, Y. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055* (2018).
211. Liu, C., Chen, L.-C., Schroff, F., Adam, H., Hua, W., Yuille, A. & Fei-Fei, L. *Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation* 2019.
212. Chu, X., Zhou, T., Zhang, B. & Li, J. *Fair DARTS: Eliminating Unfair Advantages in Differentiable Architecture Search* 2020.
213. Fu, Y., Chen, W., Wang, H., Li, H., Lin, Y. & Wang, Z. Autogan-distiller: Searching to compress generative adversarial networks. *ICML* (2020).
214. Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L.-J., Fei-Fei, L., Yuille, A., Huang, J. & Murphy, K. *Progressive Neural Architecture Search* 2017.
215. Kandasamy, K., Neiswanger, W., Schneider, J., Póczos, B. & Xing, E. *Neural Architecture Search with Bayesian Optimisation and Optimal Transport* 2019.
216. Pham, H., Guan, M. Y., Zoph, B., Le, Q. V. & Dean, J. *Efficient Neural Architecture Search via Parameter Sharing* 2018.
217. Johnson, M. K. & Adelson, E. H. *Shape Estimation in Natural Illumination in (IEEE Computer Society, USA, 2011), 2553.*
218. Lu, F., Sato, I. & Sato, Y. *Uncalibrated photometric stereo based on elevation angle recovery from BRDF symmetry of isotropic materials in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), 168.
219. Miyazaki, D., Tan, R. T., Hara, K. & Ikeuchi, K. *Polarization-based inverse rendering from a single view in Computer Vision, IEEE International Conference on* **3** (2003), 982.
220. Atkinson, G. A. & Hancock, E. R. Shape estimation using polarization and shading from two views. *IEEE transactions on pattern analysis and machine intelligence* **29**, 2001 (2007).
221. Rahmann, S. & Canterakis, N. *Reconstruction of specular surfaces using polarization imaging in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* **1** (2001), 1.
222. Guo, H., Okura, F., Shi, B., Funatomi, T., Mukaigawa, Y. & Matsushita, Y. *Multispectral Photometric Stereo for Spatially-Varying Spectral Reflectances: A well posed problem?* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), 963.
223. Kontsevich, L. L., Petrov, A. & Vergelskaya, I. Reconstruction of shape from shading in color images. *JOSA A* **11**, 1047 (1994).

CURRICULUM VITAE

PERSONAL DATA

Name	Berk Kaya
Date of Birth	January 1, 1996
Place of Birth	Ankara, Turkey
Citizen of	Turkey

EDUCATION

2019 – 2023	ETH Zürich, Switzerland Doctoral Studies in Computer Vision Lab
2017 – 2019	ETH Zürich, Switzerland MSc in Electrical Engineering and Information Technology
2013 – 2017	Middle East Technical University, Turkey BSc in Electrical and Electronics Engineering

EMPLOYMENT

June– August 2022	Research Intern Apple Zürich, Switzerland
July– August 2016	Engineering Intern TÜBİTAK Space Technologies Research Institute Ankara, Turkey
July– August 2015	Engineering Intern Turkish Aerospace Industries - TAI Ankara, Turkey