

Imperial College of Science, Technology and Medicine
Department of Computing

Practical SVBRDF and Shape Acquisition using Deep Dual Imaging

Chongrui Fan

Submitted in part fulfilment of the requirements for
the degree of Doctor of Philosophy in Computing and
the Diploma of Imperial College, October 2024

Abstract

The appearance of objects can be described by their geometric shapes and spatially varying optical reflectance maps. Traditional methods typically require specialized and complex hardware to recover geometric shapes and underlying physical models while controlling lighting conditions. Although the results are usually accurate, the costly equipment and complex capture process no longer meet the rapidly growing demands of digital content creation. With the advancement of deep learning, many rapid and straightforward methods have emerged, reducing the required number of images to a minimum—sometimes only one. However, these outstanding studies still lag in result quality, especially for structurally complex 3D objects. In this thesis, we focus on using the simplest possible capture devices, such as Off-the-shelf smartphones, and eliminate restrictions on lighting conditions as much as possible. Based on deep neural networks, our methods use two images to predict high-quality shapes and spatially varying reflectance of 3D objects and human faces, finding a sweet spot between quality and capture costs, significantly reducing the user’s capture expenses.

We utilize a smartphone’s multi-lens imaging technology to acquire high-quality shapes and spatially varying reflectance of 3D objects. Our method captures two images simultaneously using the zoom and the wide-angle lens of a smartphone, under both natural illumination and phone flash, working as efficiently as single-shot methods and achieving superior results. Furthermore, specifically for the most popular and intricate 3D object—the human face—we propose a deep learning-based imaging method that predicts the shape and SVBRDF of a subject’s face directly from two images taken from each side of the face, avoiding the need for a pre-learned morphable model and facilitating ease of use. Compared to single-image facial reconstruction methods, our approach excels in rendering faces at extreme angles and provides texture maps that are directly usable in most rendering systems.

Acknowledgements

First and foremost, I sincerely thank my advisor, Professor Abhijeet Ghosh. His guidance went beyond academics and he offered exceptional support throughout my PhD, especially when I first arrived in the UK and struggled to adapt during the pandemic. It's fair to say that without Professor Ghosh, I wouldn't be where I am today. There is no doubt that he has been the most influential person to me over these four years. I also must take this opportunity to express my deep gratitude to Yiming Lin for his meticulous guidance on my work, involving himself in every academic detail and method—an invaluable mentor to have by my side. Thank you both for your constructive comments and valuable contributions throughout the development of several papers and this thesis.

Likewise, I need to thank all the seniors and juniors in the Realistic Graphics group, including Valentin Deschaintre, Yuliya Gitlina, Alexandros Lattas, Ekin Öztürk, Xiaohui Li, Emilie Nogue, Arvin Lin, and Yiming Luo. They are not only great colleagues but also very close friends. A special thanks to Arvin Lin for often inviting me to sports activities and for the casual chats in our free time, not to mention his contributions to our research. I am lucky to have met all of you, and I must have built up quite some good karma in past lives to have the privilege of working with you these years.

Besides, my life in the UK over these four years—perhaps three, if we subtract the lockdowns—has been made richer by the following friends: Li Liu, Liang Liang, Siyi Qiu, Kang Gao, Xiaotong Ji, Siying Liu, Guang Yang, Yanran Wang, and Shiyu Bian. I'm sure that when I recall my time studying in the UK in the future, your names will be inextricably linked with those memories. I cannot list all names here, but thank you, everyone.

Dedication

Dedicated to my family and friends

‘Il n’y a qu’un héros au monde : c’est celui qui voit le monde tel qu’il est et qui l’aime’

“There is only one heroism in the world: to see the world as it is, and to love it.”

Romain Rolland

Statement of originality:

I, Chongrui Fan, declare that this thesis titled, “Practical SVBRDF and Shape Acquisition using Deep Dual Imaging” and the work presented in it are my own and any third-party material has been appropriately referenced.

© Chongrui Fan 2024

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution 4.0 International License (CC BY).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Contents

Abstract	i
Acknowledgements	iii
1 Introduction	1
1.1 Motivation and Objectives	1
1.2 Scope of this Thesis	4
1.3 Thesis Overview	5
2 Background	7
2.1 Bidirectional Reflectance Distribution Function	8
2.1.1 Basics and Empirical BRDF Models	10
2.1.2 Physically-based BRDF Models	12
2.1.3 Rendering Equation and Normal Map	18
2.2 Previous Acquisition Methods	22
2.2.1 Classic Acquisition Methods	22
2.2.2 Simplified Acquisition Methods	24
2.2.3 Polarisation and Polarisation Acquisition Methods	28

2.3	Deep Learning	33
2.3.1	Convolutional Neural Networks	33
2.3.2	Stereo Matching and Optical Flow	38
2.3.3	Diffusion Models	40
3	Deep Shape and SVBRDF Estimation using Smartphone Multi-lens Imaging	45
3.1	Introduction	48
3.2	Method	49
3.2.1	Overview	49
3.2.2	Explore on Polarisation	51
3.2.3	Depth Estimation	55
3.2.4	SVBRDF Estimation	57
3.2.5	Training and Datasets	58
3.2.6	Surface Detail Enhancement	59
3.3	Results	61
3.3.1	Optical Flow	61
3.3.2	Depth and SVBRDF	62
3.3.3	Qualitative Comparison	67
3.3.4	Quantitative Comparison	69
3.4	Limitation	71
3.5	Conclusion	73

4 Two-shot Shape and SVBRDF Reconstruction of Human Faces with Albedo-Conditioned Diffusion	75
4.1 Introduction	77
4.2 Method	79
4.2.1 Overview	79
4.2.2 Diffuse and Specular Estimation	80
4.2.3 Shape and SVBRDF Generation	81
4.2.4 Model Fine-tuning	82
4.2.5 Data Generation	86
4.3 Results	90
4.3.1 Test Dataset Evaluation	90
4.3.2 Evaluation on Public Datasets and Field Experiment	92
4.3.3 Ablation Studies	99
4.4 Limitations	104
4.5 Conclusion	105
5 Conclusion	107
Bibliography	109
A Appendix	127
A.1 Mobilephone Image Pre-processing	127
A.2 Rendering	132
A.2.1 Basic Elements of Rendering	133
A.2.2 Basic Rendering Process	137

A.3	Deep Learning	141
A.3.1	Train a Network	142
A.3.2	Prepare Training Data	144
A.3.3	Deep Learning Frameworks	146
A.3.4	Diffusion Model Principles	148
A.4	Abbreviations	150

List of Tables

3.1	Quantitative comparison on synthetic data using L1 differences. Numerically we perform significantly better compared to Li et al. [LXR*18] and Boss et al. [BJK*20] on both flash data and natural illumination data.	70
4.1	Root mean square error (RMSE) of the SVBRDF maps with our model trained with 1, 2, and 3 input images. Significant improvement can be observed when using 2 images compared to 1 image, while the improvement with 3 images over 2 images is negligible.	99
4.2	Root mean square error of the generated SVBRDF maps with and without render loss. The introduction of render loss improves all of the generated SVBRDF maps.	102
4.3	Root mean square error of the generated SVBRDF maps with and without render loss. The introduction of render loss improves all of the generated SVBRDF maps.	103

List of Figures

2.1	Schematic representation of a typical Bidirectional Reflectance Distribution Function (BRDF)	8
2.2	(a) A stone footpath photo and (b) its corresponding normal map. The normal map encodes the surface detail of the stone, allowing for realistic shading effects in rendering.	20
2.3	Visualization of the three different colour channels from the normal map. The red channel represents the x-axis detail, the green channel represents the y-axis detail, and the blue channel represents the z-axis detail. These channels encode the directional information necessary for realistic shading effects.	20
2.4	Rendered result using the normal map from Figure 2.2 applied to a plane mesh.	21
2.5	Experimental setup for measuring BRDF proposed by [BEWW*08]. This setup illustrates the arrangement of sensors around the sample to capture reflected light from multiple angles.	22
2.6	The degree of polarisation of diffused polarised and specularly polarised light as the angle of incidence changes. The degree of polarisation is a measure of the extent to which light is polarised, indicating the proportion of light waves oscillating in a particular direction.	29

2.7	(a) On the left is an ideal normalized Stokes map for a sphere under frontal flash illumination: RGB colour coding for Stokes vectors, $R(s_0)$ is set to 0.5, $G(s_1)$ and $B(s_2)$ are normalised and mapped to 0 – 1 range for visualisation. (b) On the right, we illustrate the signal captured in practice with a measured Stokes map of a rubber ball with embossed text under flash illumination.	32
2.8	Network architecture diagram of the VGG net[SZ15].	34
2.9	Network architecture diagram of the U-net [RFB15]. The architecture consists of a contracting path (encoder) that captures context and an expansive path (decoder) that enables precise localization. The skip connections between the corresponding layers in the encoder and decoder help to preserve spatial information.	35
2.10	Li et al.'s [LSC18] network for SVBRDF estimation consists of an encoder, three decoder blocks with skip links to retrieve SVBRDF components, a rendering layer, and a material classifier, followed by a Diffuse Color Refinement (DCRF) for further improvement. Based on U-Net, the network modifies the decoder section to suit its specific target better.	36
2.11	The network architecture of [DLG21] follows a general U-Net structure, dividing the decoder into three distinct branches, each handling a set of related output maps: normals and depth, diffuse reflectance, and roughness with specular reflectance. Additionally, res-blocks were introduced in the skip connections between the encoder and the various decoder branches, allowing the network to adjust the features passed to each decoder branch.	37
3.1	The estimated Bidirectional Reflectance Distribution Function (BRDF) and depth maps (c – g) of a wooden elephant are obtained from two input images (a, b) captured using a smartphone with a multi-lens imaging system. A probe is provided in the corners of input images to indicate the lighting conditions during the capture (not being used as an input). This enables realistic relighting rendering (h) under the Grace Cathedral lighting environment.	47

3.2	Pipeline of our method for estimating shape and SVBRDF for real-world 3D objects. Our pipeline takes two images (a, b) respectively taken by the zoom and wide-angle lenses of a smartphone. We use a retrained RAFT [TD20] to estimate the optical flow (c). We then feed (a, c) to a depth estimation network to obtain the normalised depth (d), and forward (d) together with (a) to the final SVBRDF estimation network to get the remaining SVBRDF maps (e-h). With the proposed depth-based rendering loss, we can qualitatively tell the mesh (j) converted from the estimated depth is smoother and more accurate than the one in (i) obtained from the depth estimated using a network trained without the depth-based rendering loss. At the same time, any details that might be lost at this stage can be partially restored during the subsequent enhancement step. . .	50
3.3	A schematic diagram of the capture setup in the Deschaintre et al. [DLG21] . .	52
3.4	Images obtained by three different lenses of OPPO Find X3 Pro	53
3.5	A diagram illustrating an image captured twice by rotating the camera to obtain photos at different polarisation angles. The direction of the optical axis of the polarising filter placed in front of the lens is indicated by blue arrows in the diagram.	54
3.6	A simple camera model: displacement of 3D points becomes ambiguous for the depth in disparity calculated from two cameras with different focal lengths. . .	55
3.7	An example of surface detail enhancement of normal map of a sponge ball. (b): the normal map directly get from network. (c): enhanced normal map.	60
3.8	Comparison of optical flow quality: 3.8b estimated optical flow from native RAFT [TD20]. 3.8c estimated optical flow from retrained RAFT using our data. The content in the white square is enlarged in the lower right corner to highlight the difference	61

- 3.9 Comparison of depth quality: (a) Input image, (b) Our result, (c) Ground Truth (GT), (d) results form Unimatch [XZ20], (e) results form CRNnet [LWX*22], and (f) results form RAFT-Stereo [LTD21]. Compared to existing methods (d), (e), and (f), which exhibit noticeable artifacts and deviations from the GT in (c), our result (b) reducing major errors observed in other methods and improved alignment with the overall structure of the GT. 62
- 3.10 We tested our method on various real-world 3D objects under both natural lighting conditions (first three rows) and flashlights (last four rows). Our estimated shape and reflectance maps generate high-quality rendering results under both environmental illumination (Grace cathedral), and flash illumination. 63
- 3.11 Relighting comparisons of acquired 3D objects under novel flash illumination conditions that were not employed during the measurement process. This figure demonstrates the ability of our method to render 3D objects under unseen lighting conditions. The sponge ball was acquired under frontal flash illumination, while the toy elephant was acquired under natural illumination. Photos taken by a smartphone camera under the novel flash angles are provided as ground truth for visual comparison. Specifically, (a_1, a_2) are photos captured under new lighting conditions, representing the ground truth (GT) for comparison. (b_1, b_2) are centred and masked versions of the photos. (c_1, c_2) are relighting render results generated using our method, simulating novel flash illumination conditions. For the sponge ball, the relighting renderings correspond to a flash positioned at 45 degrees top-left (c_1) and 45 degrees bottom-right (c_2). For the toy elephant, the relighting is performed for a flash positioned from the right (c_1) and directly above (c_2). This figure demonstrates the effectiveness of our method in reproducing plausible 3D relighting effects under new illumination conditions. . . 65

- 3.12 We compare our estimated results (a, c, e, g) to g.t. (b, d, f, h) on various synthetic 3D objects not seen during training, under both natural lighting conditions (a, c) and flashlights (e, g). Our estimated maps are accurate in comparison to g.t. maps, and renderings are close to g.t. renderings. We focus on qualitative comparisons through rendered images, which better demonstrate how the predicted maps perform in practice. Rendering results provide an intuitive way to evaluate the consistency of both shape and reflectance properties compared to the ground truth, directly reflecting the visual impact of the network’s predictions. 66
- 3.13 We compare to the flash-based method of Li et al. [LXR*18] and the two-shots method of Boss et al. [BJK*20] on a synthetic example. Our estimated maps are closer to g.t. maps than results of [LXR*18, BJK*20], and our renderings are also a better match to the g.t. The method proposed by Li et al.[LXR*18] does not provide specular outputs; therefore, we did not include this comparison. . . . 67
- 3.14 We compare some real objects to the flash-based method of Li et al. [LXR*18]. Overall, our approach has a much more correct global shape and does not suffer from low-frequency bias and our rendering results are qualitatively better as our shape estimation is superior. Specifically, for the ball case, Li et al.’s results limited the 3D fidelity of the estimated normals. The normals resemble a flattened disc rather than accurately reconstructing the ball’s curvature, which compresses the numerical range and exaggerates the appearance of finer details. In contrast, our method preserves the ball’s 3D geometry, resulting in a more faithful representation of the true surface shape. Again the method proposed by Li et al.[LXR*18] does not provide specular outputs; therefore, we did not include this comparison. 68
- 3.15 A failure case for Boss et al.’s method [BJK*20] under strong natural illumination. Under strong natural lighting, it becomes difficult to distinguish between input images captured with (a), and without (b) flash. The absence of flash cues in the input results in large errors in estimated surface normal and depth. . . . 69
- 3.16 Some partial failure cases. Since our network has been trained on dominantly dielectric materials, it doesn’t work that well on other types of scenes/subjects such as a face (top row), or a dominantly metallic object (bottom row). 71

4.1	Overall pipeline of our proposed method which requires two photographs of each side of the face as input. Our method first predicts the image-space diffuse and specular albedo with a light-weight estimation network, then uses the diffuse and specular albedo images, as well as the original photographs, as guidance to train a denoising diffusion network that predicts the shape and SVBRDF maps in UV-space. Our results enable 3D rendering of the given subject under any desirable environment maps.	79
4.2	Graphical illustration of our simple diffuse and specular estimation network. . .	80
4.3	Graphical illustration of our shape and SVBRDF generation diffusion denoising network. The guidance images are concatenated to the noisy shape and SVBRDF maps. The timestep is a value between zero and 1 and is passed into all blocks with time embedding except the final output layer.	81
4.4	An example showing the locations of key points found by landmarks detection. The figure shows the unwrapped position map as a grayscale image, and the key points found are represented by red dots. (a): the key points found by applying the landmark detection on one ground truth map. (b): the key points detected on the generated position map in the later-stage of training.	84
4.5	An illustration of the capture setup used by the RealFaceDB and (b-e) is one reference sample from the RealFaceDB.	87
4.6	UV unwrapping of a base mesh in the RealFaceDB dataset. (a): Basemesh visualization. (b): UV unwrapped image after remapping positions of vertices with texture coordinates. (c): Dense UV position map obtained with linear interpolation of (b).	88
4.7	Estimated diffuse and specular albedo (right two columns) given a well lit face image (left most column). The input images have been rendered under the St. Peters environment (a), and uniform white illumination (b), respectively. . . .	90

4.8	Comparison of the results generated by the network and ground truth. The first row shows the ground truth shape and SVBRDF maps, while the second row presents our generated results. From left to right: diffuse normal map, specular normal map, diffuse albedo map, specular albedo map, position map. Through visual inspection, it can be observed that our network generates texture maps that are qualitatively consistent with the ground truth maps.	91
4.9	The reconstructed mesh (a) and the relighting results under multiple novel lighting conditions (b, c, d), compared to the ground truth (GT). The relighting results are illuminated under Park (b), Pisa (c), and Grace Cathedral (d).	92
4.10	Generated results from two uniformly-lit photographs. The first row displays the model's two input images. Rows 2 to 5 show the generated UV space SVBRDF maps. The remaining mesh and rendering results will be shown in the next figure, Fig 4.11	93
4.11	Rest results from two uniformly-lit photographs. The first row displays the model's two input images. while row 2 is the reconstructed mesh. The mesh has been rotated to a certain angle to demonstrate the effectiveness of our 3D restoration. The last 2 rows display the relighting results based on the outputs above. The left side indicates the lighting conditions.	94
4.12	Comparison of rendered results between our method and headNeRF [HPX*22]. Top row shows the actual photographs that serve as ground truth. The second and third row shows our rendered results and results rendered by headNeRF respectively. Our method better preserves the identity and is more robust when rendering at extreme angles. While headNeRF requires only one input image, their work struggles to reconstruct details that are not present at the frontal image.	95
4.13	Comparison of our generated SVBRDF maps with AvatarMe++ [LMP*21]. We compare the diffuse albedo maps (a), specular albedo maps (b), and the specular normal maps (c). Our generated maps are less noisy and contains more details on the sides.	96

4.14	Additional results of subjects captured outdoor. The first row displays the raw images captured. The second low shows the model’s two input images. Rows 3 to 6 show the generated UV space SVBRDF maps. The last 2 rows display the relighting results based on the outputs above. The left side indicates the lighting conditions as same as Fig 4.10.	97
4.15	Additional results of subjects captured outdoor. The first row displays the raw images captured. The second low shows the model’s two input images. Rows 3 to 6 show the generated UV space SVBRDF maps. The last 2 rows display the relighting results based on the outputs above. The left side indicates the lighting conditions as same as Fig 4.10.	98
4.16	Comparison of albedo maps obtained using different number of input images. The map generated map using only the frontal image as input fails to accurately reconstruct the beard on the side cheek due to the lack of information in frontal only image.	100
4.17	Comparison of albedo maps obtained with diffuse an specular estimation network or directly predict the UV space maps. In terms of details, the result on the far right exhibits less noise and better consistency compared to the image in the middle.	100
4.18	Comparison of the generated diffuse albedo map with and without first estimating the diffuse and specular maps in image space. (a): generated diffuse using only the captured photographs. (b): generated diffuse using the captured photographs and the estimated diffuse and specular maps in image space. It can be observed that by first estimating the diffuse and specular albedo, the generated diffuse albedo map is more robust to lighting variations in the captured photographs.	101
4.19	Reconstructed mesh with position maps obtained before fine-tuning (a) and after the fine-tuning stage (b). The improvement is obtained with the introduction of landmark loss and smoothness loss in the fine-tuning stage.	103

A.1	Images were taken with three different lenses of the mobile phone, and the results after distortion correction. (a) telephoto lens. (b) wide-angle lens (c) ultra-wide-angle lens.	128
A.2	One example using five photos with different exposure times to synthesise HDR images	130
A.3	One example for a green fitness ball photos before and after white balance processing (a): Original image; (b): After white balance processing	131
A.4	A simple scene with an object and several polygons, demonstrating the difference between directional light and spotlight illumination.	135
A.5	Two examples of light maps. (a) Garden EV map representing an outdoor environment, and (b) Thatch Chapel EV map representing an indoor environment. Both maps were captured by artists and sourced from the free, open-source visual sharing website, Polyhaven.	136
A.6	Key stages in the OpenGL rendering pipeline, illustrating the flow from vertex data to rasterization. Image sourced from the open-source tutorial LearnOpenGL.	138
A.7	Coordinate transformations in the vertex shader, illustrating transformations from local space to screen space through model, world, view, and projection matrices. Image sourced from the open-source tutorial LearnOpenGL.	139
A.8	A simplified diagram of a single-layer perceptron neural network. The diagram illustrates the process of combining multiple inputs x_1, x_2, x_3 with their respective weights w_1, w_2, w_3 , adding a bias term b , and applying an activation function f_{act} to produce the output y_{out}	141
A.9	Decomposition of a noisy image into different noise components and their respective reconstructions [HBC23]. The figure illustrates that the decomposition of a noisy image can be arbitrary.	149
A.10	An infinitely subdivided separation operation will eventually converge to a fixed path where the randomness is averaged out [HBC23].	149

Chapter 1

Introduction

1.1 Motivation and Objectives

Computer graphics rendering has permeated various aspects of people’s digital lives. Applications such as gaming and film have taken a significant role in everyday entertainment, generating tremendous economic benefits. For example, Black Myth: Wukong, a game released in August 2024, recently broke the record for the highest number of concurrent players on the Steam platform and has grossed over 442.2 million British pounds. In these types of high-quality games, which take years to develop, game developers and artists no longer focus solely on gameplay and mechanics. Instead, they strive to deliver a comprehensive entertainment experience through music, art, and special effects.

In particular, AAA games today have moved beyond simple, repetitive geometric backgrounds like buildings or forests. Instead, development teams conduct on-site surveys, scan real-world structures and statues, and use these elements as the foundation for game design. Many architectural elements in Black Myth: Wukong are inspired by existing ancient Chinese buildings, including temples, pagodas, palaces, castles, residences, and gardens. Interactive objects in the game, such as large bells, incense burners, scrolls, and fans, are no longer freely designed by artists and programmers using polygons. Instead, real-life objects that fit the game’s setting are captured and faithfully recreated in 3D. This approach significantly enhances player immersion,

allowing them to feel truly absorbed in the game’s fantastical environments.

Beyond the setting, many MMORPGs (Massively Multiplayer Online Role-Playing Games) and large role-playing games offer increasingly rich character customization options for players. Many players are passionate about this feature, spending over an hour in the ‘character creation’ phase to craft their virtual persona before the game starts. This character-creation process can significantly impact the following tens or even hundreds of hours of gameplay. To enhance and streamline the user experience in this aspect, some historical MMORPGs, such as Justice Online (Nishuihan), allow players to upload photos to generate a 3D virtual avatar (which can, of course, be adjusted to look more handsome or beautiful as desired). Once introduced, this feature of creating a virtual digital double received widespread praise, even attracting more players than traditional elements like the game’s graphics, main storyline, or unique gameplay features. At the same time, with the rapid development of internet technology, an increasing number of online applications and social networking platforms are offering users the ability to create virtual avatars, allowing them to use their 3D models to chat with friends from around the world in virtual chat rooms.

All of these exciting graphics applications rely on accurate models of material or facial reflectance and shape, as our perception of objects is a complex function of their geometry, intrinsic reflectance properties, and the lighting conditions under which we observe them. It is easy to imagine that computer rendering of an object using accurately measured reflectance data would be the most realistic. With this intent in mind, measurement-based methods have started attracting researchers’ interest, and since the 1990s, numerous methods for acquiring reflectance data have been specifically designed. Although these methods are undoubtedly expensive, bulky, and complicated to operate, they have paved the way for realistic rendering in graphics applications.

Subsequently, research in this area has primarily focused on simplifying the acquisition process, sometimes sacrificing a bit of precision for the sake of significantly simplifying the setup, which is clearly more appealing to non-expert users. The reason for this trend is not hard to understand, as borrowing historical artefacts for scanning in a lab is obviously an enormous

undertaking. Moreover, for individual users, accessing such specialised scanning environments is often unrealistic. Simplifying this laborious capture process and reducing the cost of expensive equipment would help unlock the full creative potential of 3D capture in the hands of individual artists, rather than limiting it to large film and game studios.

Over the long journey of research and development, material model acquisition methods have gradually shifted towards image-based or photo-based approaches, with state-of-the-art techniques able to produce relatively good results from only a few captures. Although existing methods are quite advanced, our goal is to continue exploring lightweight implementations, pushing the boundaries of current technological capabilities even further. This desire serves as the main motivation for this thesis, in which we demonstrate various in-the-field and lightweight methods capable of modelling everyday objects and faces both indoors and outdoors.

1.2 Scope of this Thesis

In computer graphics, Practical SVBRDF (Spatially Varying Bidirectional Reflectance Distribution Function) and Shape Acquisition generally refer to capturing the material reflectance properties and geometric shape of real-world objects for digital reconstruction and rendering. To address the challenges, this thesis contributes to the field in the following ways:

First, we propose a practical and efficient methodology for capturing SVBRDF and shape data for everyday 3D objects using commercially available smartphones without introducing additional sensors. By leveraging the subtle baseline and viewing angle differences provided by two smartphone cameras, we extract optical flow fields to estimate fine-scale depth. The estimated depth is further used to guide SVBRDF estimation, producing superior results compared to existing single-shot methods. Additionally, this two-view approach demonstrates enhanced robustness under varying lighting conditions.

Second, we introduce a novel two-shot deep learning method for human appearance capture. This method predicts the subject’s facial shape and SVBRDF directly from two images captured from opposite sides of the face. It offers a user-friendly solution that eliminates the need for pre-trained deformable models, making the capture process more accessible and efficient.

Moreover, the proposed methods ensure high compatibility with mainstream rendering systems. The texture maps generated by our approach can be directly used to drive popular rendering engines, providing practical utility for digital content creation, and bridging the gap between deep learning-based acquisition and rendering technology.

In summary, this thesis pushes the frontier of practical SVBRDF and shape acquisition by presenting innovative methodologies. The proposed contributions advance the field by improving efficiency, robustness, and rendering compatibility.

1.3 Thesis Overview

The structure of the remainder of this thesis is as follows.

We begin with the background knowledge, introducing key concepts related to our main research objective, SVBRDF. This includes both empirical models and physically-based models, along with an introduction to the rendering equation based on these models. In Section 2.2, we review existing techniques for SVBRDF and shape acquisition, covering approaches from early measurement methods to modern estimation techniques with limited information. This section also includes fundamental mathematical principles of polarisation signals and techniques utilising polarisation, which were a major focus of our early explorations. At the end of the background section, we discuss the deep learning methods primarily used in this thesis. We highlight network architectures and related developments applied to SVBRDF and shape acquisition. Additionally, this part provides an overview of two other types of deep learning networks discussed in Chapters 3 and 4: stereo matching and optical flow, as well as the development of diffusion generative models.

Starting from Chapter 3, we introduce our new practical method for acquiring SVBRDF and shapes. Chapter 3 begins with appearance capture for everyday objects. We utilize smartphones, which have now become widely available in daily life, as the capture tool. By leveraging the multi-camera capabilities of smartphones, we efficiently capture SVBRDF and shape data. The method works well under both smartphone flash and daily outdoor and indoor lighting conditions, producing better results than previous single-angle image methods. In this chapter, we demonstrate the robustness of our method in various ways, including the rendering results of the captured materials.

In Chapter 4, we shift our focus to humans, specifically capturing the appearance of human heads. Inspired by the emerging generative models, our method captures the SVBRDF and base shape of a human head using just two input images, providing a quick and easy capture process. This method significantly reduces the limitations imposed by lighting conditions. We generate the subject’s 3D geometry by directly learning 2D positional maps, thereby avoiding

the need for pre-learned deformable models. Our method has advantages in rendering extreme facial angles and producing texture maps that can be used directly in most rendering systems.

Finally, we summarise the contributions of this work and present the conclusions in Chapter 5. Following the references, the appendix provides high-level background content related to the topics discussed in the thesis, aimed at facilitating a better understanding of the work accomplished. The appendix also includes data preprocessing procedures and a table of abbreviations.

This thesis is based on the following publications:

- Deep Shape and SVBRDF Estimation using Smartphone Multi-lens Imaging. C. Fan, Y. Lin, A. Ghosh. *Computer Graphics Forum*, 42(7), e14972.
- Two-shot Shape and SVBRDF Reconstruction of Human Faces with Albedo-Conditioned Diffusion. C. Fan, Y. Lin, A. Lin, A. Ghosh. (Under submission to Eurographics 2025 at the time of thesis submission)

Chapter 2

Background

Describing the appearance of a 3D object essentially involves characterizing its reflectance and shape under illumination. In this chapter, we begin with background knowledge, provide an overview of current research in computer graphics, and highlight some notable achievements in the field.

We will start by introducing the concept of the BRDF, which describes how materials interact with light. We review different forms of BRDF, including diffuse and specular reflection, as well as empirical and physically-based models. We then briefly discuss the rendering equation driven by these models. This section also addresses the representation of material shape, including normals and depth. After covering these fundamental concepts, a general understanding of how those remarkable visual effects are achieved will be obtained. Following this, we introduce prior work on material model acquisition, ranging from traditional laboratory setups to lightweight capture methods. We then shift our focus to deep learning methods, discussing basic concepts of commonly used framework architectures and the advantages they provide. Naturally, the above content is tied to several important prior works in this field, which have had significant influence.

Discussions of background knowledge or mathematical formulas can sometimes be tedious and overwhelming. Therefore, throughout Chapter 2, we include numerous examples to make the text more engaging, with the aim of enhancing the overall narrative.

2.1 Bidirectional Reflectance Distribution Function

Most of the objects we see in our daily lives do not emit light themselves. We are able to see them because they reflect light from sources such as the sun, lamps, or other types of illumination. Describing the interaction of electromagnetic radiation with the surface of objects is evidently a complex issue. In 1965, Nicodemus [Nic65] was the first to mathematically model this interaction, introducing the Bidirectional Reflectance Distribution Function (BRDF), which is defined as follows:

$$f_r(\omega_i, \omega_o) = \frac{dL_o(\omega_o)}{dE_i(\omega_i)} = \frac{dL_o(\omega_o)}{L_i(\omega_i) \cos \theta_i d\omega_i} \quad (2.1)$$

Where $\omega_i = (\theta_i, \phi_i)$ and $\omega_o = (\theta_o, \phi_o)$ represent the incoming and outgoing light directions, respectively, in spherical coordinates. The term $dE_i(\omega_i)$ denotes the irradiance, which is the incident radiant flux per unit area along ω_i , while $dL_o(\omega_o)$ represents the reflected radiance or the outgoing radiant flux per unit area along ω_o .

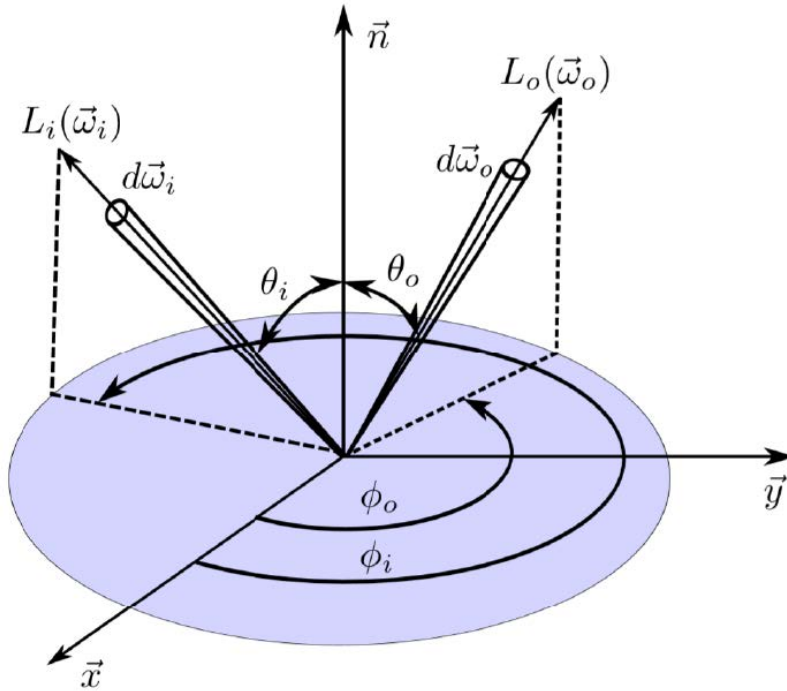


Figure 2.1: Schematic representation of a typical Bidirectional Reflectance Distribution Function (BRDF)

A physically plausible BRDF must satisfy several key properties:

- **Non-negativity:** The BRDF must always yield non-negative values for any pair of directions $(\vec{\omega}_i, \vec{\omega}_o)$.
- **Reciprocity:** The BRDF must be symmetric under the exchange of incident and outgoing directions, which is expressed as:

$$\forall(\vec{\omega}_i, \vec{\omega}_o) \in (\Omega^+)^2, f_r(\vec{\omega}_i, \vec{\omega}_o) = f_r(\vec{\omega}_o, \vec{\omega}_i) \quad (2.2)$$

- **Energy conservation:** The total amount of reflected energy cannot exceed the incoming energy at any surface point. This principle is mathematically expressed as:

$$\forall \vec{\omega}_i \in \Omega^+, \int_{\Omega^+} f_r(\vec{\omega}_i, \vec{\omega}_o)(\vec{n} \cdot \vec{\omega}_o) d\vec{\omega}_o \leq 1 \quad (2.3)$$

These properties form the foundation for physically based BRDF models and ensure their physical plausibility in rendering simulations.

For homogeneous materials, the BRDF is a function of the wavelength and other four variables, specifically the incoming light direction $\omega_i = (\theta_i, \phi_i)$ and the outgoing light direction $\omega_o = (\theta_o, \phi_o)$. For materials with spatial variation, i.e., non-homogeneous materials where the BRDF changes with surface position x , this function expands to the Spatially Varying Bidirectional Reflectance Distribution Function (SVBRDF). In this thesis, our focus is on specific 3D targets, so the results will predominantly be represented as SVBRDFs with 2D spatial distribution. For convenience, there may be some interchangeability between the terms BRDF and SVBRDF in this chapter. This is noted here to avoid any potential confusion.

There are two main components of reflectance: diffuse reflection and specular reflection. The optical appearance of an object that we observe results from the combined contributions of these two components. In the following sections, we begin with the basics by introducing the fundamental concepts of both types of reflection and their empirical models. Subsequently, more accurate, physically-based models are discussed.

2.1.1 Basics and Empirical BRDF Models

Diffuse reflection is caused by the interaction between light and the interior of the material's surface, and it is independent of the viewing direction. More specifically, after a point receives illumination, the outgoing radiance remains constant across all directions in the upper hemisphere. This classical and simplest BRDF model is known as the Lambertian model [Lam92]:

$$f_r(\omega_o, \omega_i) = \frac{\rho_d}{\pi} \quad (2.4)$$

where ρ_d refers to what is commonly called the diffuse albedo. When rendering real-world objects, it is typically represented using a 2D map, with each pixel corresponding to the diffuse albedo value at a specific position on the object's surface. This is also a crucial component of the SVBRDF map.

The Lambertian model is an empirical BRDF model, which is both classical and widely used in computer graphics rendering. However, real-world objects are rarely ideal Lambertian surfaces. The rough surfaces of real objects tend to appear brighter when the viewing direction is close to the illumination direction [PH10].

In reality, purely diffuse materials are almost nonexistent. In addition to diffuse reflection, materials also exhibit specular reflection, which is more complex and highly dependent on the viewing angle. Several empirical models attempt to describe specular behaviour in simple forms, among which the simplest is the Phong model [Pho75]. The model is mathematically defined as:

$$f_{r,\text{spec}}(\vec{\omega}_o, \vec{\omega}_i) = \rho_s (\vec{\omega}_r \cdot \vec{\omega}_o)^s \quad (2.5)$$

ρ_s represents the specular albedo, which corresponds to the specular colour and intensity at a given surface point. The parameter s controls the shininess of the surface: higher s values result in smoother material appearances with a smaller specular lobe, whereas lower s values lead to

rougher surfaces with broader lobes. Specular reflection in the Phong model is considered view-dependent, with the strongest reflected radiance observed in the reflected light direction ω_r , calculated as:

$$\vec{\omega}_r = 2(\vec{\omega}_i \cdot \vec{n})\vec{n} - \vec{\omega}_i \quad (2.6)$$

The fall-off of observed radiance away from ω_r is described by the specular lobe, expressed in the Phong BRDF as the cosine term $\omega_r \cdot \omega_o$. Similar to diffuse albedo, if a material exhibits an SVBRDF, its specular albedo is represented by a specular albedo map.

Building upon this, the Blinn model [Bli77] introduced some adjustments to the specular lobe:

$$f_{r,\text{spec}}(\vec{\omega}_o, \vec{\omega}_i) = \frac{\rho_s (\vec{n} \cdot \vec{\omega}_h)^s}{\vec{n} \cdot \vec{\omega}_i} \quad (2.7)$$

In the Blinn-Phong model, the specular lobe is described by a cosine function of the angle between the normal vector $\vec{n}$ and the half-vector $\vec{\omega}_h$. The half-vector $\vec{\omega}_h$ is the unit vector that points halfway between the outgoing direction $\vec{\omega}_o$ and the incoming direction $\vec{\omega}_i$.

$$\vec{\omega}_h = \frac{\vec{\omega}_o + \vec{\omega}_i}{\|\vec{\omega}_o + \vec{\omega}_i\|} \quad (2.8)$$

According to Blinn's model, specular intensity reaches its maximum value when the surface normal aligns with the half-vector and decreases as the half-vector deviates from the surface normal.

A fundamental principle in physically based rendering is energy conservation. However, simpler BRDF models, such as the Lambertian and Phong models, do not always adhere to this constraint. For example, the Phong model can produce excessively high specular reflectance values for certain combinations of parameters, leading to energy non-conservation. This occurs because the model lacks proper normalization to account for the reflected energy. While this issue may not be visually apparent in many rendering scenarios, it becomes critical in physically

accurate simulations or when integrating with global illumination algorithms.

Furthermore, this simple and easy-to-use model does not perform well for anisotropic materials. To address this limitation, the Ward BRDF model [War92] introduces modifications as follows:

$$f r_{\text{spec}}(\vec{\omega}_o, \vec{\omega}_i) = \frac{\rho_s}{4\pi\alpha_u\alpha_v\sqrt{(\vec{\omega}_i \cdot \vec{n})(\vec{\omega}_o \cdot \vec{n})}} \exp\left(-\tan^2(\delta)\left(\frac{\cos^2(\phi_h)}{\alpha_u^2} + \frac{\sin^2(\phi_h)}{\alpha_v^2}\right)\right) \quad (2.9)$$

Here, α_u and α_v are parameters controlling the anisotropy along the tangent $\vec{t}$ and bitangent $\vec{b}$ directions of the surface. The tangent $\vec{t}$ aligns with the u -direction of the texture coordinates, while the bitangent $\vec{b}$ aligns with the v -direction. Together with the normal vector $\vec{n}$, they form an orthonormal basis known as the local tangent space. This tangent space allows the reflection properties to be described relative to the surface orientation, making it particularly suitable for modelling anisotropic materials. The angles δ and ϕ_h represent the elevation and azimuth angles, respectively, of the half-vector in this tangent space. The Ward model uses an elliptical Gaussian distribution to describe the specular lobe and satisfies the principle of reciprocity.

It can be observed that, although models have become increasingly accurate in describing complex reflection phenomena, their growing complexity has moved away from the simplicity and intuitive nature of earlier empirical models. This trade-off between physical accuracy and computational simplicity reflects the evolving priorities in rendering techniques. Consequently, in recent years, particularly with the continuous improvement in hardware computational power, physically-based BRDF models have gained increasing popularity.

2.1.2 Physically-based BRDF Models

Although empirical BRDF models are fast to compute, they are not based on actual physical modelling. Consequently, renderings using empirical models often differ from real-world observations. While this difference is generally acceptable, physically-based BRDF models not only maintain physical correctness (e.g., energy conservation, reciprocity) but also deliver photore-

alistic rendering results for the end user. As a result, physically-based BRDF models based on microfacet theory are more widely used today. This theory models the surface as a collection of mirror-like microfacets. Each microfacet has an associated slope, and the overall distribution of these slopes defines the surface's roughness. The first microfacet BRDF was proposed by Torrance and Sparrow [TS67] and was later adapted to computer graphics by Cook and Torrance [CT82], which defines the Cook-Torrance BRDF as:

$$f_s(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \eta)G(\vec{\omega}_i, \vec{\omega}_o)D(\vec{\omega}_h, \sigma)}{\pi|\vec{n} \cdot \vec{\omega}_i||\vec{n} \cdot \vec{\omega}_o|} \quad (2.10)$$

Where $F(\vec{\omega}_h, \eta)$ is the Fresnel term, dependent on the halfway vector $\vec{\omega}_h$ and the index of refraction η , $G(\vec{\omega}_i, \vec{\omega}_o)$ describes the attenuation of light due to shadowing and masking between neighbouring micro facets, and $D(\vec{\omega}_h, \sigma)$ is the distribution function that describes the statistical orientation of the microfacets with respect to the halfway vector $\vec{\omega}_h$ and the roughness parameter σ . We provide a detailed introduction to the components F , G , and D below.

Fresnel Equations

The F term in the Equation 2.10 is commonly referred to as the Fresnel equation. This concept is well-known and important in physical optics. When light travels from one medium with a refractive index of n_1 to another medium with a refractive index of n_2 at the interface between the two (usually referred to as the boundary), both reflection and refraction of light may occur simultaneously. The Fresnel equations describe how different components of the light wave are refracted and reflected, as well as the phase shift that occurs upon reflection. These equations successfully account for the polarisation effects produced when light, as a typical electromagnetic wave, is reflected by an object. Polarisation refers to the property of specifying the geometric direction of transverse wave oscillations, particularly the direction of oscillation of the electric field in light waves. If the light is linearly polarised, its electric field oscillates in a single direction.

Using the Fresnel equations, we can quantitatively represent the affected light as:

$$R_s = \left| \frac{n_1 \cos \theta_i - n_2 \sqrt{1 - \left(\frac{n_1}{n_2} \sin \theta_i\right)^2}}{n_1 \cos \theta_i + n_2 \sqrt{1 - \left(\frac{n_1}{n_2} \sin \theta_i\right)^2}} \right|^2 \quad (2.11)$$

$$R_p = \left| \frac{n_1 \sqrt{1 - \left(\frac{n_1}{n_2} \sin \theta_i\right)^2} - n_2 \cos \theta_i}{n_1 \sqrt{1 - \left(\frac{n_1}{n_2} \sin \theta_i\right)^2} + n_2 \cos \theta_i} \right|^2 \quad (2.12)$$

Transmittance is given by:

$$T_s = 1 - R_s \quad (2.13)$$

$$T_p = 1 - R_p \quad (2.14)$$

The coefficients R_s and R_p represent the fractions of incident light power that are reflected at the interface for s -polarised and p -polarised light, respectively. The transmission coefficients T_s and T_p are given by $1 - R_s$ and $1 - R_p$, assuming no absorption or scattering occurs at the interface. The angles θ_i is the incident angles

Since any polarisation state can be decomposed into two orthogonal linearly polarised waves, the electric field components perpendicular and parallel to the plane of incidence (s - and p -polarisation) are sufficient to describe the light's behaviour:

- The electric field component of the polarised incident light is perpendicular to the plane formed by the incident and reflected light. This state of incident light is called **s-polarisation**, derived from the German word *senkrecht* (meaning "perpendicular").
- The electric field component of the polarised incident light is parallel to the plane formed by the incident and reflected light. This state of incident light is called **p-polarisation**, derived from the German word *parallel* (meaning "parallel").

Although the equations may seem complex, in practical applications, the refractive index is often fixed for a given object, so users do not need to pay much attention to it. Instead, the

polarisation state of the light is primarily influenced by the direction of the incident light, the direction of the outgoing light, and the surface normal of the object. This means that when the direction of illumination (from a controlled light source) and the direction of outgoing light (from a specific viewing angle) are known, a relationship can be established between the intensity of the collected light and the normal direction of the surface. This relationship forms the underlying principle of many polarisation-based surface measurement methods.

At the outset of our exploration of practical methods, we attempted to follow in the footsteps of previous successful approaches, aiming to effectively utilise polarisation information to enhance our results. Unfortunately, our attempt did not achieve the expected outcome, and we will discuss further details in Chapter 3.

If we wish to render images under polarisation (i.e., to simulate a photograph taken with a camera equipped with a polarising filter), we need to use the full Fresnel equations to separate the R_s and R_p components. However, if this is not required, in most cases, we can use the simplified unpolarised Fresnel reflection coefficient:

$$F(\theta_i) = \frac{R_p(\theta_i) + R_s(\theta_i)}{2} \quad (2.15)$$

Cook and Torrance simplified the Fresnel reflection coefficient for unpolarised light to:

$$F(\theta_h) = \left(\frac{1 + (g - \cos \theta_h)^2}{2(g + \cos \theta_h)^2} \right) \left(1 + \left(\frac{\cos \theta_h (g + \cos \theta_h) - 1}{\cos \theta_h (g - \cos \theta_h) + 1} \right)^2 \right) \quad (2.16)$$

where θ_h is the angle between the incoming direction $\vec{\omega}_i$ and the halfway vector $\vec{\omega}_h$, and $g = \sqrt{n_t^2 - 1 + \cos^2 \theta_h}$.

Building on this, Schlick [Sch94] proposed an even simpler calculation method:

$$F(\theta_h) = F(0^\circ) + (1 - F(0^\circ))(1 - \cos \theta_h)^5 \quad (2.17)$$

Reflectance at normal incidence is determined from the indices of refraction:

$$F(0^\circ) = \left(\frac{n_i - n_t}{n_i + n_t} \right)^2 \quad (2.18)$$

where n_i is the index of refraction of the incident medium and n_t is that of the transmitting medium. The reflectance at normal incidence, $F(0^\circ)$, is determined solely by the refractive indices of the two media, providing a material-dependent baseline for reflectance calculations.

This approximation significantly reduces the computational cost while maintaining reasonable accuracy, making it particularly useful for real-time rendering applications. After introducing Schlick's method for approximating the Fresnel term, we first establish how the reflection coefficient is influenced by material properties and the angle of incidence, which is crucial in rendering.

Microfacet Distribution and Geometry

The microfacet distribution function $D(\vec{\omega}_i, \vec{\omega}_o, \vec{n}, \sigma)$ is a bell-shaped function that gives the differential area of microfacets with the surface normal $\vec{n}$. The geometry function $G(\vec{\omega}_i, \vec{\omega}_o, \vec{n}, \sigma)$ accounts for occlusion and shadowing between neighbouring micro facets, and it is dependent on the distribution function. Cook and Torrance originally proposed a Beckmann distribution function with their BRDF model:

$$D(\vec{\omega}_i, \vec{\omega}_o, \vec{n}, \sigma) = \frac{1}{\pi \sigma^2 (\vec{\omega}_h \cdot \vec{n})^4} \exp \left(\frac{(\vec{\omega}_h \cdot \vec{n})^2 - 1}{(\vec{\omega}_h \cdot \vec{n})^2 \sigma^2} \right) \quad (2.19)$$

Where σ is the roughness parameter, defined as the Root Mean Squared (RMS) slope of the microfacets. The Beckmann distribution describes the statistical distribution of microfacet normals and is often compared to a Gaussian-like shape due to its bell-like appearance. The geometry function associated with the Beckmann distribution is:

$$G(\vec{\omega}_i, \vec{\omega}_o, \vec{n}) = \min \left\{ 1, \frac{2(\vec{\omega}_h \cdot \vec{n})(\vec{\omega}_o \cdot \vec{n})}{(\vec{\omega}_h \cdot \vec{\omega}_o)}, \frac{2(\vec{\omega}_h \cdot \vec{n})(\vec{\omega}_i \cdot \vec{n})}{(\vec{\omega}_h \cdot \vec{\omega}_o)} \right\} \quad (2.20)$$

Equation 2.20 is non-differentiable and contains first derivative discontinuities, which are not seen in real surfaces. This poses challenges in accurately modelling real-world materials. As noted by Walter et al. [WMLT07], the Smith shadowing-masking term is more appropriate for describing the geometry function, as it avoids these issues while providing a more physically plausible representation. The Smith shadowing-masking term is defined as:

$$G(\vec{\omega}_i, \vec{\omega}_o, \vec{n}) \approx G_1(\vec{\omega}_i, \vec{\omega}_h)G_1(\vec{\omega}_o, \vec{\omega}_h) \quad (2.21)$$

Where G_1 is the single-sided shadowing term, defined for a single direction. The approximation splits the geometry function into independent components for the incoming and outgoing directions, improving its applicability in physically based rendering and optimization tasks. Where G_1 can be derived directly from the distribution function.

Another widely used microfacet distribution function is the GGX distribution [WMLT07]. This function is commonly applied in physically-based rendering engines due to its high efficiency and differentiability. The G_1 term of GGX is defined as:

$$G_1(\vec{\omega}_i, \vec{\omega}_o) = \frac{2}{1 + \sqrt{1 + \sigma^2 (1 - (\vec{n} \cdot \vec{\omega}_h)^2) / (\vec{n} \cdot \vec{\omega}_h)^2}} \quad (2.22)$$

The distribution function $D(\vec{\omega}_i, \vec{\omega}_o, \vec{n}, \sigma)$ is defined as:

$$D(\vec{\omega}_h, \vec{n}, \sigma) = \frac{\sigma^2}{\pi(\vec{n} \cdot \vec{\omega}_h)^4} \left(\frac{\sigma^2 + 1 - (\vec{n} \cdot \vec{\omega}_h)^2}{(\vec{n} \cdot \vec{\omega}_h)^2} \right)^2 \quad (2.23)$$

After obtaining the required F , G , and D terms, we can compute the BRDF based on the microfacet model. Models based on microfacet theory are more commonly used today. In the paper [WMLT07], the π term in the denominator of Equation 2.10 is replaced by the number 4.

This is a common approximation, and in this thesis, we also follow this approximation, using Equation 2.24 in the subsequent sections.

$$f_s(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \eta)G(\vec{\omega}_i, \vec{\omega}_o)D(\vec{\omega}_h, \sigma)}{4|\vec{n} \cdot \vec{\omega}_i||\vec{n} \cdot \vec{\omega}_o|} \quad (2.24)$$

The microfacet model is primarily used to simulate specular reflection, assuming that each microfacet represents an infinitely small mirror where perfect reflection occurs. However, diffuse reflection, as observed on real-world objects, originates from subsurface scattering and other phenomena where light penetrates the surface and interacts within the material. These processes cannot be directly captured by the microfacet model.

To achieve better visual results in rendering, we incorporate a separate diffuse reflection model. One widely used model for diffuse reflection is the Oren-Nayar BRDF[ON94], which approximates the effects of surface roughness on the reflected light without explicitly relying on microfacet definitions. The formula for the Oren-Nayar approximation is as follows:

$$f_d(\omega_i, \omega_o) = \frac{\rho_d}{\pi} (A + B \cdot \max\{0, \cos(\phi_i - \phi_o)\} \cdot \sin \alpha \cdot \tan \beta) \quad (2.25)$$

where $A = 1 - 0.5 \frac{\sigma^2}{\sigma^2 + 0.33}$, $B = 0.45 \frac{\sigma^2}{\sigma^2 + 0.09}$, $\alpha = \max\{\theta_i, \theta_o\}$, $\beta = \min\{\theta_i, \theta_o\}$, and σ is the surface's roughness.

By integrating these two components, we can achieve a physically plausible yet computationally efficient representation of material reflectance, which is essential for realistic rendering in computer graphics.

2.1.3 Rendering Equation and Normal Map

The Rendering Equation [Kaj86] is the core mathematical model used to describe the interaction between light and object surfaces in the 3D rendering process. It is not an isolated step but serves as a computational framework typically used during the fragment shading stage to

calculate the lighting effects for each pixel. After understanding the concept of BRDF, we will provide additional explanations for this section at the end.

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_o) L_i(\mathbf{x}, \omega_i) (\omega_i \cdot \mathbf{n}) d\omega_i \quad (2.26)$$

- $L_o(\mathbf{x}, \omega_o)$: Radiance leaving point $\mathbf{x}$ in the outgoing direction ω_o , i.e., the observed light in that direction.
- $L_e(\mathbf{x}, \omega_o)$: Emitted radiance from point $\mathbf{x}$ in direction ω_o , such as light emitted from a light source or other emissive objects.
- $\int_{\Omega}$: Integral over the unit hemisphere Ω , representing all possible incoming light directions.
- $f_r(\mathbf{x}, \omega_i, \omega_o)$: Bidirectional Reflectance Distribution Function (BRDF), describing the reflection of light from direction ω_i to ω_o at point $\mathbf{x}$.
- $L_i(\mathbf{x}, \omega_i)$: Incoming radiance at point $\mathbf{x}$ from direction ω_i , i.e., the light arriving from that direction.
- $\omega_i \cdot \mathbf{n}$: Dot product between the incoming direction ω_i and the surface normal $\mathbf{n}$ at point $\mathbf{x}$, representing the angle between them and affecting the intensity of the incoming light.
- $d\omega_i$: Differential solid angle element, used for the integral.

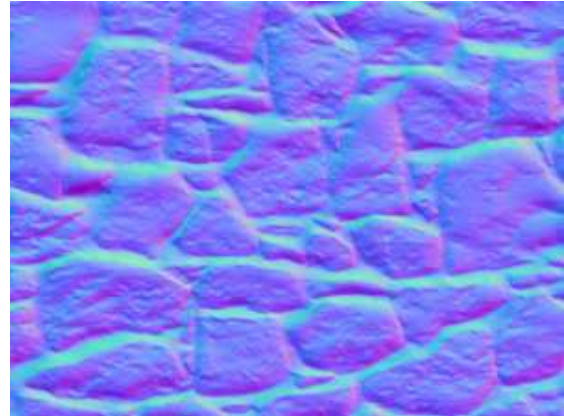
This equation takes into account the complex behaviours of light, including reflection, refraction, and diffusion, and describes how the outgoing light at any point on a surface is a combination of emitted light and incoming light after reflection. For our purposes, since the object itself does not emit light, the first term can be omitted. In the fragment shading stage, the colour and lighting of each pixel are calculated based on the second half of the equation, using the BRDF model discussed above and the specified light sources introduced in Section A.2.1, and surface normal information (which can be provided via a Normal Map).

The Normal Map, similar in form to the diffuse map and specular map, is a 2D map corresponding to each pixel. Normally, each surface of a 3D object has a normal, which determines

the direction of light reflection on the surface. A Normal Map uses a texture to store the offset information of these normals, providing pseudo-normals for each pixel in the rendering equation. This directly affects the calculation of incoming and outgoing light in the rendering equation without the need to increase the number of geometric polygons on the object.



(a) A stone footpath image



(b) normal map

Figure 2.2: (a) A stone footpath photo and (b) its corresponding normal map. The normal map encodes the surface detail of the stone, allowing for realistic shading effects in rendering.

The Normal Map is typically encoded using RGB channels, where the R (red channel) represents the variation of the normal along the x-axis, the G (green channel) represents the variation along the y-axis, and the B (blue channel) represents the variation along the z-axis.

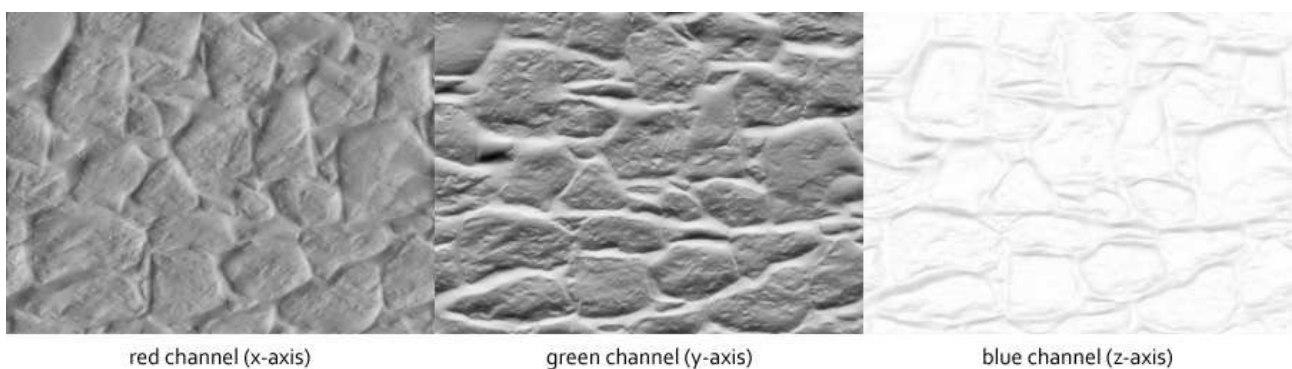


Figure 2.3: Visualization of the three different colour channels from the normal map. The red channel represents the x-axis detail, the green channel represents the y-axis detail, and the blue channel represents the z-axis detail. These channels encode the directional information necessary for realistic shading effects.

In the image, the green channel shows that brighter areas (values closer to 1.0) define regions where the normal points upward, while darker areas (values closer to 0.0) define regions where

the normal points downward. Most normal maps appear bluish because the Z-axis (blue channel) usually points towards us, with relatively high values.

For example, in the case of a classic stone path, we can set its mesh (basic shape) as a simple plane and use the normal vectors from the Normal Map when calculating the rendering equation, resulting in high-quality lighting effects.



Figure 2.4: Rendered result using the normal map from Figure 2.2 applied to a plane mesh.

At this point, we have introduced the BRDF and normal maps required to drive the rendering equation. The goal of this thesis is to acquire the shape and material maps necessary for 3D objects through image-based methods, including specular reflectance and diffuse reflectance maps involved in the BRDF model etc., as well as the normal maps introduced in this section. These maps can further drive realistic rendering. A brief introduction to how rendering engine tools convert these maps into rendered images is provided in the Appendix. In Chapters 3 and Chapters 4, the results will be presented primarily in the form of various material maps and rendering outcomes. It is hoped that this chapter helps clarify our research objectives when encountering related topics. In the next section, we will discuss the innovative contributions made by previous researchers to achieve similar goals.

2.2 Previous Acquisition Methods

2.2.1 Classic Acquisition Methods

The traditional and classic approach to material model acquisition (BRDF and shape) aims to fully restore every parameter in the rendering equation. While this approach is certainly the most accurate, it is also the most inconvenient. The most common and traditional way to measure BRDF is by using a small-angle light source and a corresponding radiometer to measure scattered radiance. The classic device for such measurements is the gonioreflectometer (angle reflectometer). Sing et al. [Foo97] were among the first to apply the gonioreflectometer to computer graphics research. Due to the design of such devices, only one radiance value can be measured at a time, making the process extremely time-consuming. These devices are also typically large and expensive. This setup, without further explanation, is clearly unsuitable for most research and commercial needs. To speed up BRDF measurements and reduce equipment costs, Ben-Ezra et al. [BEWW*08] designed a system using Light Emitting Diodes (LEDs) for BRDF acquisition without requiring cameras or mirrors. Their method is theoretically capable of high-speed, high-density BRDF acquisition.

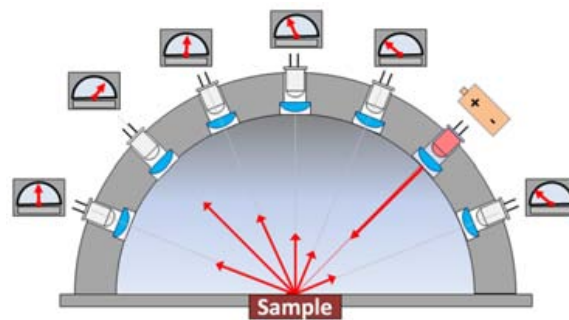


Figure 2.5: Experimental setup for measuring BRDF proposed by [BEWW*08]. This setup illustrates the arrangement of sensors around the sample to capture reflected light from multiple angles.

With the advancement and maturation of imaging technology, researchers have found that a single photograph can capture light interactions from multiple incident and outgoing directions simultaneously (at least for BRDFs). This significantly reduces the number of photographs required for dense sampling. [War92] proposed the first image-based BRDF measurement

method. Their method uses a so-called imaging gonioreflectometer, which employs a curved mirror and a Charge-Coupled Device (CCD) to capture multiple reflected rays in parallel. Even without using a curved mirror, similar results can be achieved by using convex material samples. Marschner et al. [MWL*99] provided an example of this by using a digital camera and a stable light source to capture the isotropic BRDF of curved material samples (such as cylinders or spheres). By adding a 3D scanner, their method could measure a broader class of near-convex curved objects. Based on a similar idea, Matusik et al. [MPBM03] proposed a setup for measuring the BRDF of uniform spheres. They later used this method to construct the MERL database. However, even for uniform and isotropic materials, these methods still require a very high sampling density. In particular, exhaustive image sampling from all viewing angles and lighting directions is essential for capturing the object’s characteristics.

Subsequent research has attempted to further simplify the BRDF acquisition process by carefully designing light sources to provide multiple incident directions simultaneously, thereby reducing the number of required images compared to methods using a single light source. For example, Gardner et al. [GTHD03] built a mechanical gantry with a linear light source. Their method captures spatially varying reflectance maps of planar surfaces by recording a video from a fixed viewpoint while the linear light source scans across the surface. Chen et al. [CDP*14] later proposed a similar design for capturing anisotropic BRDFs.

In addition to using linear light sources, researchers also proposed methods utilising area light sources, which provide broader controlled illumination coverage, further simplifying the acquisition process. A typical example of such an acquisition method is Ghosh et al. [GHAO10], who directly measured BRDFs in a basis representation by projecting a series of basis functions from a spherical set of directions. Another notable method was proposed by Ghosh et al. [GCP*09], where they used a Liquid Crystal Display (LCD) to project spherical gradient illumination patterns, recovering the specular roughness and anisotropic properties of anisotropic materials.

2.2.2 Simplified Acquisition Methods

Complete material model measurements require controlled lighting, dense sampling, extensive knowledge of reflectometry, and meticulous handling of the measurement process, which poses a significant challenge for non-experts. Additionally, these measurements typically demand highly precise descriptions of material reflectance and shape properties. For many computer graphics applications, there is often a preference for simplifying the acquisition process by slightly compromising the accuracy of BRDF and shape, which is a trade-off for convenience. Based on this simplified approach, researchers have proposed various material model estimation methods that reduce reliance on complex acquisition equipment, making it easier for end users to obtain material data.

We classify these methods as simplified acquisition methods, and they have shown remarkable technical progress. The traditional BRDF estimation problem is usually complex due to its high dimensionality, but the introduction of analytic BRDF models has greatly simplified this complexity. For isotropic materials, commonly used analytic models require only three parameters: diffuse albedo, specular albedo, and specular roughness, allowing for fairly accurate estimates without extensive sampling. These models have lower dimensionality and can simplify the acquisition process by inferring geometry or lighting information, as demonstrated in previous work [KRFB06, AAL16, AWL15, DAD*18, RPG14].

There are also methods that employ non-parametric models [RZ10, BM15], which play an important role in reducing the dependence on collected data. Non-parametric approaches allow for more flexibility by relying on statistical inferences rather than fixed parametric descriptions, making them particularly useful for materials that exhibit complex, irregular properties that cannot be easily captured by simple parametric models.

Reducing the number of images

Some research within this domain focuses on reducing the number of images required for BRDF estimation. Wang et al. [KRFB06] proposed an image-based material editing method that

employs image processing techniques, such as colour and brightness threshold recognition, to separate lighting effects from material properties within an image. This work demonstrated the possibility of simplifying material capture processes to enable efficient material modelling. In 2015, Aittala et al. [AAL16] introduced a simplified material modelling method based on neural texture synthesis, showcasing the feasibility of capturing material properties with a small number of input images. By leveraging neural networks to learn the texture characteristics of materials, this method achieved efficient material capture and modelling. Subsequently, in 2018, Deschaintre et al. [DAD*18] proposed a deep learning-based method for single-image SVBRDF estimation, which maintained high model accuracy while reducing the number of required input images.

Despite these significant advances in reducing the number of images required for BRDF estimation, certain limitations remain, particularly regarding lighting conditions. For example, Wang et al. [KRFB06] faced challenges in accurately separating lighting effects from material properties under uncontrolled lighting environments. Similarly, while the methods of Aittala et al. [AAL16] demonstrated efficient material modelling, their experiments assumed that the measured objects were planar. Overall, the practical applicability of these methods in real-world scenarios requires further investigation, especially under non-standard lighting conditions or for objects with complex geometries.

Reducing light limitation

Other works have focused on reducing the constraints imposed by lighting conditions. Under unknown lighting conditions, Romeiro and Zickler [RZ10] used Bayesian inference to derive BRDFs through a linear combination of reflectance basis functions. The advantage of their method is that it only requires an HDR image of a material sample and does not need controlled lighting. However, this method is only applicable to samples with smooth surfaces and known geometry, and the final rendering results do not meet expectations. Similarly, unknown geometry also poses challenges for material capture, as it is difficult to distinguish whether the reflection is caused by geometry or material properties.

In extreme cases where both lighting and geometry are unknown, Barron et al. [BM15] proposed a framework for the joint recovery of reflectance, lighting, and geometry by making assumptions about surface smoothness, reflectance sparsity, and other priors. However, their assumptions limit the applicability of this method, particularly when dealing with rough specular or diffuse materials. Additionally, their method only estimates diffuse albedo, not the full BRDF. Finally, Xia et al. [XDPT16] estimated surface reflectance and geometry from a video of a rotating object, but this method requires a complex image capture process, making it less suitable for simplified acquisition needs.

In summary, while significant progress has been made in simplifying material acquisition, challenges remain in terms of computational resources and scene applicability. Future research will need to further balance the trade-off between model accuracy and acquisition convenience.

Using commodity hardware

In particular, there is a category of BRDF acquisition methods focused on compact and portable capture techniques employing commodity devices, which are more closely related to our work.

Wu et al. [WZ15] have proposed an integrated system for hand-held acquisition of shape and reflectance of objects with a Kinect sensor. Aittala et al. [AWL15] have proposed a two-shot method for the acquisition of stationary materials using a mobile phone. They employ a pair of flash-no-flash observations of the sample coupled with statistical analysis to extract reflectance maps. Riviere et al. [RPG14] proposed two mobile acquisition setups for the acquisition of more general spatially varying planar surfaces. Free-form acquisition with flash illumination has also been employed for acquiring SVBRDFs of planar surfaces [HSL*17], and non-planar 3D objects [NLGK18].

Recent methods have demonstrated state-of-the-art results employing specialised LED panel/area-source [MKZ*21] for such free-form acquisition. Xu et al. [XLZ*23] propose a method using structured light consisting of an LED array and an LCD mask coupled with a single-lens reflex camera to obtain shape and reflectivity from a single view. Wu et al. [WWZ16], Park et

al. [PNS18] and Ha et al. [HBNK20] showed a series of progress in collecting geometry and SVBRDF, surface light field and shape in motion using RGB-D sensors.

However, the above family of approaches either relies on a large number of measurements and/or strong priors, such as self-repetitive materials or planar geometry or uses additional dedicated devices or sensors to obtain depth and SVBRDF.

2.2.3 Polarisation and Polarisation Acquisition Methods

Polarisation of light waves or electromagnetic waves refers to the direction of the oscillation of their electric field. For most light sources (such as sunlight and fluorescent lamps), the oscillation direction of the emitted light is random, resulting in unpolarised light. By using optical components such as a linear polariser, light oscillating in directions perpendicular to the filter will be blocked, allowing us to obtain a beam of light oscillating in only one fixed direction. Since the endpoint of the oscillation vector traces a straight line, this light is also called linearly polarised light. Additionally, polarised light can include beams where the endpoint of the oscillation vector forms a circular or elliptical shape.

When light makes contact with an object, polarisation occurs due to the different oscillation directions of reflected and refracted light, as described by Fresnel equations in the BRDF chapter. The polarisation effect that occurs on the surface of the object is called specular polarisation. At a specific angle, known as the Brewster angle, the reflected light becomes purely linearly polarised. The degree of polarisation (DOP) is used to describe the strength of polarisation. In most cases, specular reflection accounts for the majority of polarised light, but to some extent, diffuse reflection can also exhibit polarisation. The strength and distribution of diffuse polarisation are very different from those of specular polarisation. Figure 2.6 is an excellent example demonstrating this difference.

Stokes Parameters and Mueller Calculus can describe the change in polarisation state caused by a beam of light shining on an object and reflecting off. The Stokes parameters form a four-dimensional vector known as the Stokes vector: $S = (S_0, S_1, S_2, S_3)$, which was introduced by George Gabriel Stokes in 1852 [Sto52]. Here, S_0 represents the total intensity of light, S_1 and S_2 represent the intensity of 0° and $+45^\circ$ polarisation, respectively, and S_3 represents the intensity of right circular polarisation. Using these four vector values, we can simply represent the current polarisation state of the light.

The change in polarisation state after light reflection can be calculated using Mueller Calculus [Mue43]. A Stokes vector is transformed into a new Stokes vector when a polarised ray

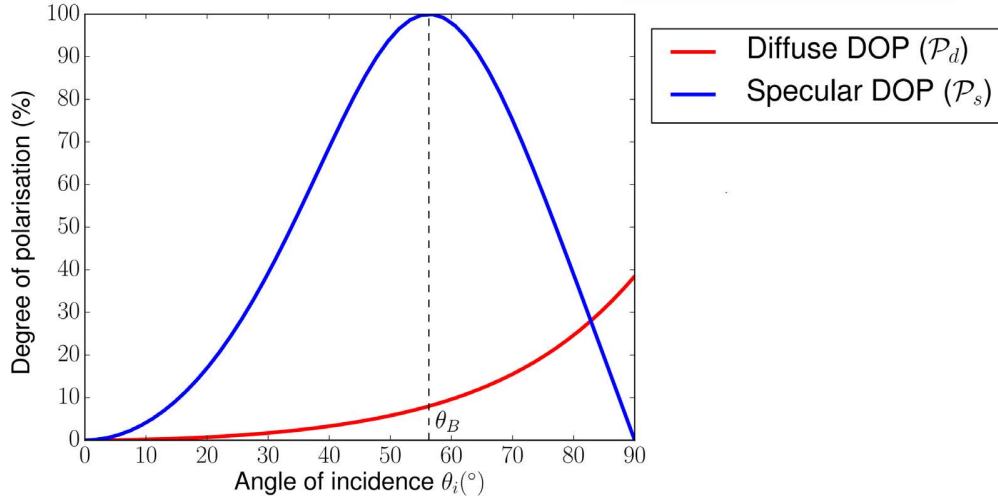


Figure 2.6: The degree of polarisation of diffused polarised and specularly polarised light as the angle of incidence changes. The degree of polarisation is a measure of the extent to which light is polarised, indicating the proportion of light waves oscillating in a particular direction.

interacts with a surface during light transport. In Mueller calculus, this transformation can be described by linear operators as:

$$s' = C(\phi)D(\delta; n)R(\theta; n)C(-\phi)s \quad (2.27)$$

In the formula, C is the Mueller rotation matrix, ϕ is the angle between the plane of incidence and the x-axis, D represents the retardation Mueller matrix, n is the surface normal, and R is the Mueller reflection matrix.

The Mueller rotation matrix represents a rotation around the z -axis by the angle ϕ :

$$C(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2\phi) & \sin(2\phi) & 0 \\ 0 & -\sin(2\phi) & \cos(2\phi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.28)$$

The retardation matrix $D(\delta; n)$ represents a phase retardation, where δ is the phase shift and n is the surface normal:

$$D(\delta; n) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\delta) & 0 & \sin(\delta) \\ 0 & 0 & 1 & 0 \\ 0 & -\sin(\delta) & 0 & \cos(\delta) \end{pmatrix} \quad (2.29)$$

The Mueller reflection matrix describes the change in polarisation upon reflection, where θ is the angle of incidence:

$$R(\theta; n) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & R_{\parallel} - R_{\perp} & 0 & 0 \\ 0 & 0 & R_{\perp} & 0 \\ 0 & 0 & 0 & R_{\parallel} + R_{\perp} \end{pmatrix} \quad (2.30)$$

where $R_{\parallel}$ and $R_{\perp}$ are the reflection coefficients for parallel and perpendicular polarisation components.

By expressing the optical element as a matrix, the change in polarisation state expressed by Stokes Parameters can be calculated and represented using matrix multiplication. When calculating the reflection change, the Mueller matrix appears as the product of a series of matrices. Although the structure is complex, it provides us with a powerful tool to analyse the changes in polarisation before and after reflection. Polarisation-based methods primarily leverage the principle of using this formula, attempting to detect the target by utilising the state changes that light, as an electromagnetic wave, undergoes when interacting with objects.

Polarisation techniques are widely applied in the fields of computer graphics and vision, including material classification, shape estimation, reflectance separation, and reflectometry. Early research, such as Wolff's work [Wol90], used the phase of Transmitted Radiance Sinusoid (TRS) for multi-view shape estimation, with surface normals determined using two views. Miyazaki et al. [MKI03] further combined this with reflectance modelling. Atkinson and Hancock [AH04] were among the first teams to study diffuse polarisation in shape estimation, although most materials exhibit a mixture of specular and diffuse polarisation. Ghosh et al. [GCP*10] pro-

posed a method for accurately recovering reflectance by combining circularly polarised light with Stokes parameters. Ma et al. proposed using polarised spherical gradient patterns to compute specular and diffuse reflectance maps [MHP*07], while Ghosh et al. further extended this work to multiview face capture [GFT*11].

Other notable contributions include the work of Guarnera et al. [GPDG12], who studied how to infer object shapes using circular polarisation, and Kadambi et al. [KTSR15], who proposed a technique to enhance shape estimation by combining polarisation cues with depth sensor data. Smith et al. [SRT16] addressed the azimuth ambiguity caused by linear polarisation using an optimisation approach. Taamazyan et al. [TKR16] successfully separated diffuse and specular reflection and recovered 3D shape by combining viewpoint and polarisation data. In terms of reflectance estimation, Riviere et al. [RRFG17] proposed a method for capturing outdoor planar material reflectance using polarisation images, while Baek et al. [BJTK18] estimated the shape and reflectance of convex objects using active lighting and a known initial polarisation state.

Among these works, it is worth mentioning Deschaintre et al.’s research [DLG21]. Their new method demonstrated that combining polarisation information with deep learning networks can achieve better results than using images captured with flash illumination alone. Given that our work has been deeply influenced by this study, a more detailed explanation of its methods is provided here.

The highlight of this successful approach lies in the introduction of the Stokes map as an abstraction of the polarisation signal, which is input into the network along with the raw image, thereby enhancing the input for deep learning. The primary advantage is that the polarisation state of light serves as a powerful cue for surface normals, and the Stokes map, described by the Stokes parameters, is strongly correlated with the surface normals of an object.

To obtain the Stokes map, it is sufficient to calculate it from three polarised photos. By observing the object at three different polarisation angles (0° , 45° , and 90° , corresponding to I_H , I_{45} , and I_V), we obtain S_0 , S_1 , and S_2 and can be converted into RGB representation. The Stokes map can be used to visualise the distribution of polarised light on the object’s surface.

After normalisation, the Stokes map of a perfect sphere is shown in Figure 2.7. Since this method does not involve circular polarisation, S_3 is set aside and not discussed.

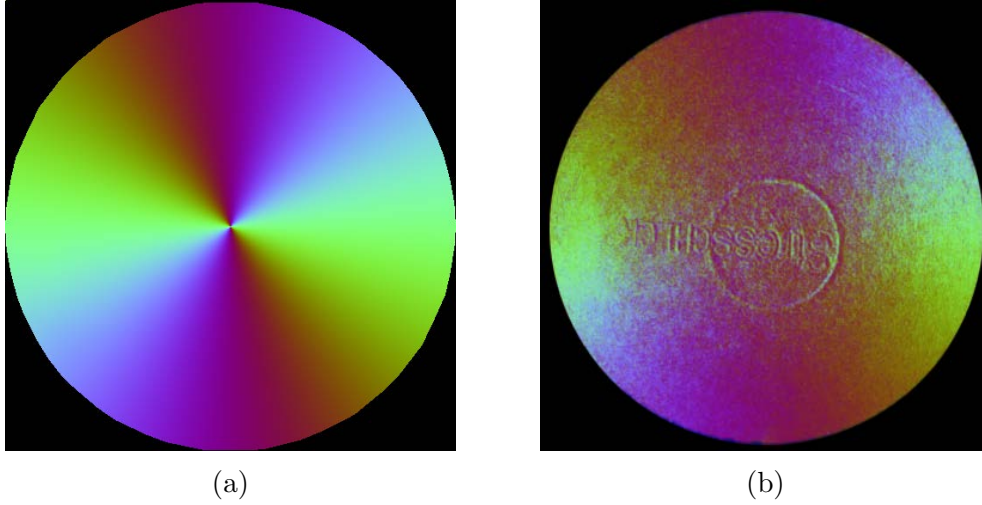


Figure 2.7: (a) On the left is an ideal normalized Stokes map for a sphere under frontal flash illumination: RGB colour coding for Stokes vectors, R (s_0) is set to 0.5, G (s_1) and B (s_2) are normalised and mapped to 0 – 1 range for visualisation. (b) On the right, we illustrate the signal captured in practice with a measured Stokes map of a rubber ball with embossed text under flash illumination.

By computing multiple images of an object into Stokes maps and providing them alongside the original photographs to the network, Deschaintre et al. [DLG21] successfully achieved superior results compared to other single-view capture methods. This offers us an approach to incorporate polarisation information into image inputs through multiple captures. The detailed network structure of this work will be introduced in the Deep Learning Section.

2.3 Deep Learning

2.3.1 Convolutional Neural Networks

The utilisation of deep learning tools has become a primary approach for appearance extraction tasks and even for the broader field of computer graphics research. In this section, we will introduce classic architectures and gradually narrow the focus to target appearance, shape, and SVBRDF acquisition. To aid in a better understanding of this section, the appendix provides a brief explanation of the basic principles of deep learning and how to train a model.

The problem we are addressing mainly relies on images as input, and for image-related applications, we frequently use Convolutional Neural Networks (CNNs). CNNs were first proposed by LeCun et al. in 1998 [LBBH98]. CNNs effectively reduce the number of connections between neurons, enabling neural networks to handle the massive input data associated with images (for example, even a small 64x64 image with three channels yields a feature vector dimension of 12,288). A classic CNN consists of convolutional layers that apply kernels to the image matrix using a sliding window to extract high-level features, pooling layers that reduce the spatial size of the convolutional features, and fully connected layers that perform nonlinear tasks using the extracted features.

Since LeCun's LeNet in 1998, which was used to solve the visual task of handwritten digit recognition, many new CNN architectures have been developed. In 2012, Krizhevsky et al. [KSH17] proposed AlexNet, which consisted of five convolutional layers and three fully connected layers. AlexNet significantly increased the number of layers compared to LeNet and won the 2012 ImageNet competition with an overwhelming margin of 10.9 percentage points ahead of the second-place model. In 2014, Oxford University's Visual Geometry Group (VGG) proposed VGG-Nets [SZ15], which adopted a pre-training approach: first training a smaller network and gradually deepening it once this smaller network stabilised. By this point, CNNs had already become quite deep.

In 2015, He Kaiming introduced ResNet [HZRS16], which made a significant innovation in

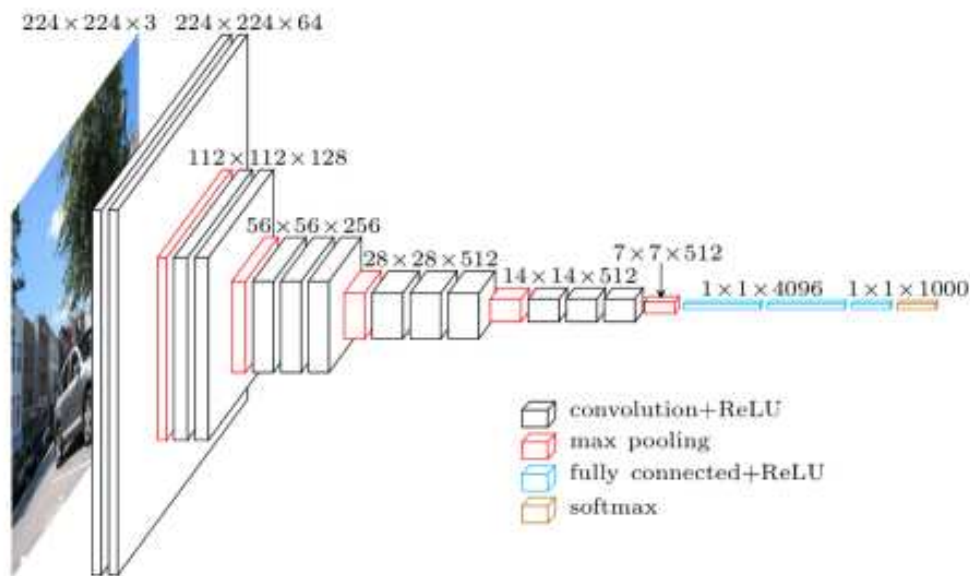


Figure 2.8: Network architecture diagram of the VGG net[SZ15].

network structure by incorporating skip connections between adjacent layers. This allowed for better information flow and mitigated the issue where deeper parts of the network might fail to learn effectively. Building on this, DenseNet [HLVDMW17] adopted the principles of ResNet by densely deploying skip connections between all previous layers, enabling each layer to receive input from all preceding layers and resulting in powerful gradient flow.

U-Net Architecture

The classic CNN primarily focuses on image classification tasks, such as those we previously mentioned, where the network is provided with a photo and asked to distinguish whether it is a cat or a dog. However, for tasks where the output is also an image, such as the common image segmentation task in the medical field, U-Net was proposed by Ronneberger et al. in 2015 [RFB15]. As its name suggests, U-Net resembles two symmetrical convolutional neural networks, as shown in Figure 2.9. The network structure employs a series of convolutional layers that reduce the spatial dimensions of the image while increasing its depth (number of channels), forming the encoder. The obtained latent feature vectors are then passed through symmetrical deconvolution layers to return to the original data dimensions (this part is generally referred to as the decoder). A notable feature of U-Net is the use of skip connections that link the feature

maps from the encoder to the corresponding layers in the decoder. This helps the decoder better utilise the positional and contextual information from the original input, thereby improving the accuracy of the output.

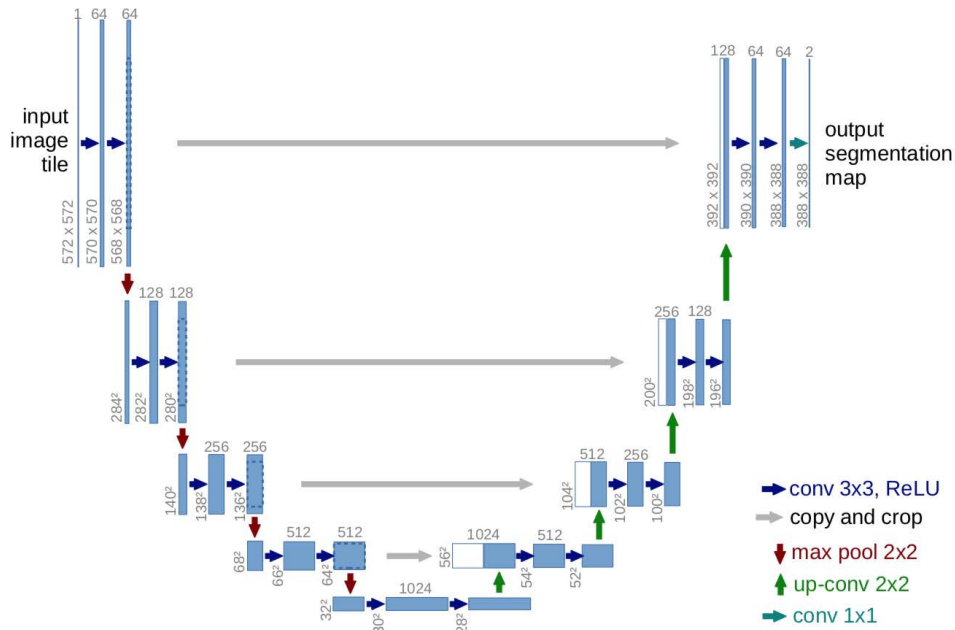


Figure 2.9: Network architecture diagram of the U-net [RFB15]. The architecture consists of a contracting path (encoder) that captures context and an expansive path (decoder) that enables precise localization. The skip connections between the corresponding layers in the encoder and decoder help to preserve spatial information.

Several methods have recently been proposed for surface reflectance and shape estimation from sparse measurements. In 2017, Li et al. [LDPT17] proposed a Self-Augmented Convolutional Neural Network to model physically plausible spatially varying surface reflectance functions (SVBRDF) from a single photograph of a planar sample captured under unknown natural lighting. Meka et al. [MMZ*18] further expanded the application of deep learning in material estimation by developing a real-time material estimation system, LIME, capable of quickly inferring reflectance and normal information from a single RGB-D image, providing an effective solution for real-time applications. That same year, Li et al. [LSC18] addressed the limitations of mobile devices by proposing a method to estimate SVBRDF and normals from a single image captured using a smartphone. This work combined flash and ambient lighting and improved the network's robustness and accuracy through physics-inspired modules. Deep learning-based SVBRDF extraction methods greatly simplify the image capture process, delegating the task

of generating reflectance maps to the network through carefully designed architectures.

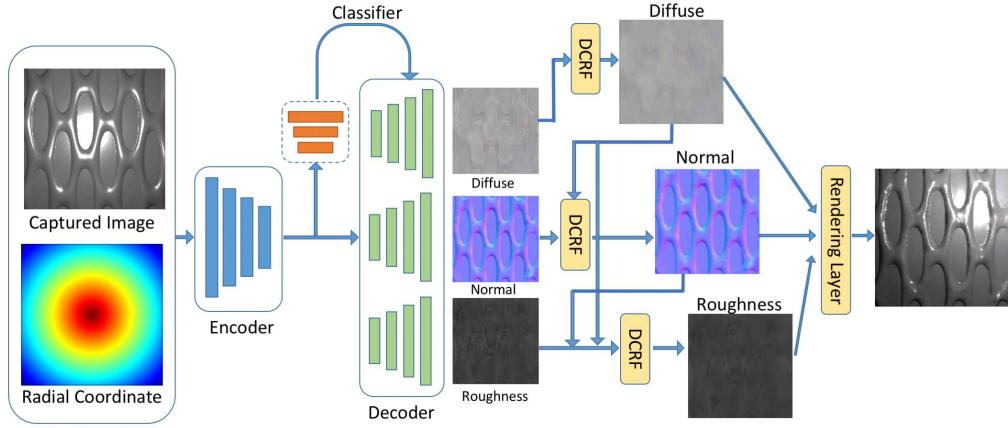


Figure 2.10: Li et al.’s [LSC18] network for SVBRDF estimation consists of an encoder, three decoder blocks with skip links to retrieve SVBRDF components, a rendering layer, and a material classifier, followed by a Diffuse Color Refinement (DCRF) for further improvement. Based on U-Net, the network modifies the decoder section to suit its specific target better.

Beyond these works, researchers have tried to introduce additional information to improve the generalisation capability for complex materials and geometries as well. In 2019, Deschaintre et al. [DAD*19] incorporated multi-view information from a variable number of uncalibrated and unordered images to enhance the reconstruction of complex appearances. Their method used multiple U-Nets to process each input individually, pooling the resulting feature maps into a fixed-size representation of the material. In the same year, Gao et al. [GLD*19] proposed a unified deep inverse rendering framework capable of recovering high-resolution SVBRDF properties from an arbitrary number of input images, making it broadly applicable to high-resolution images and complex materials. Boss et al. [BJK*20] introduced an efficient method requiring only two images to estimate SVBRDF and shape properties for complex objects. Their model utilised observations under different lighting conditions, maintaining high accuracy while reducing the input data requirements.

One particularly noteworthy contribution is introducing polarisation information into deep learning inputs. Deschaintre et al. [DLG21] demonstrated high-quality estimation of 3D object shapes (surface normals and depth) and SVBRDF under frontal flash illumination using single-view polarised imaging. Their acquisition process involved capturing an object under flash illumination with three polarisation filter orientations: 0° , 45° , and 90° . Using a DSLR camera

mounted on a tripod and a linear polarising filter rotated manually, they calculated Stokes maps and Normalised Color Maps from the captured images. These maps served as additional inputs to the deep learning network. This work effectively shows that even though acquiring both 3D shape (surface normals and depth) and spatially varying reflectance from a single view is a highly ill-posed problem, deep learning combined with additional information can significantly improve the results.

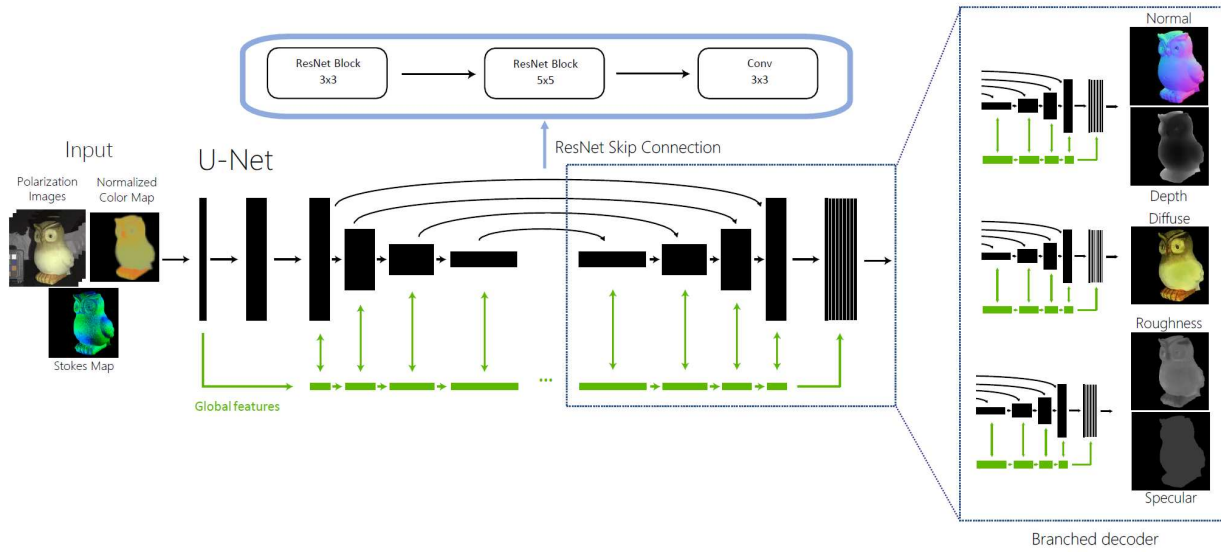


Figure 2.11: The network architecture of [DLG21] follows a general U-Net structure, dividing the decoder into three distinct branches, each handling a set of related output maps: normals and depth, diffuse reflectance, and roughness with specular reflectance. Additionally, res-blocks were introduced in the skip connections between the encoder and the various decoder branches, allowing the network to adjust the features passed to each decoder branch.

This multi-branch network design enables shared input and separated outputs. The encoder shares input features (e.g., global surface features), which are then divided into multiple branches during the decoding stage. This is particularly suitable for SVBRDF prediction tasks, where SVBRDF is composed of multiple reflectance components with distinct physical properties. Separating these components allows the network to model each one more accurately without interference from others. Furthermore, the res-blocks in the skip connections enable feature adjustment tailored to specific branches. For instance, the diffuse branch can focus on low-frequency global features, while the specular branch can prioritise fine high-frequency details.

The aforementioned works demonstrate the applicability, advantages, and feasibility of U-Net

structures for SVBRDF extraction tasks. U-Net’s down-sampling path captures global contextual information, while its skip connections and up-sampling path retain high-resolution local features. The robustness of U-Net’s base architecture, coupled with its adaptability to incorporate further advancements in deep learning networks, ensures its continued significance in the field of SVBRDF extraction tasks.

2.3.2 Stereo Matching and Optical Flow

When utilising deep learning networks to process images, a natural approach is to incorporate multiple inputs to provide the network with more information, thereby improving the quality of the final results. However, in some cases, the use of multiple inputs (in our cases, images) introduces a new variable: the images may not be pixel-aligned, which could complicate the network’s processing and hinder its effectiveness.

Beyond this merging information task, aligning the corresponding parts of multiple images is itself a significant research topic in computer vision, particularly in the field of dense stereo matching. Establishing explicit pixel correspondences between images not only facilitates subsequent network operations or other processing methods but also enables an effective understanding of the geometric relationships between images. This is particularly beneficial for 3D-related tasks, where precise spatial understanding is crucial.

In the context of stereo matching, various methods have been proposed over the years [PSR*17, KFR*18, YMHR19, GYY*19, THZ*21, XZ20, LTD21, LWX*22]. Each successive work has demonstrated improvements over its predecessors on standard stereo datasets, such as KITTI [MHG15], ETH3D [SSG*17], and Middlebury [SHK*14]. These advancements highlight the importance of accurately aligning images to improve performance in stereo-matching tasks, further underscoring its role as a key challenge in computer vision.

One line of methods [PSR*17, THZ*21, XZ20] is based on 2D convolution. Pang et al.[PSR*17] used multiscale residual learning, while AANet[XZ20] proposed a novel aggregation method using sparse points and multi-scale interaction. Furthermore, Tankovich et al.[THZ*21] intro-

duced a differentiable 2D geometric propagation and warping mechanism to infer the disparity. Another line of methods [GY*19] calculates a cost volume between two input images and filters the cost volume through 3D convolutions, which requires high computation and memory costs. RAFT-Stereo [LTD21] adapted the iterative refinement strategy from the optical flow network RAFT [TD20] to design a network for stereo matching. Similarly, Li et al. [LWX*22] proposed using Recurrent Update Modules with Adaptive Group Correlation, which produces a much smaller cost volume than RAFT-Stereo [LTD21]. Furthermore, Xu et al. [XZC*23] employed a transformer network and formulated optical flow, rectified stereo matching, and stereo depth estimation as a unified dense correspondence matching problem.

However, all these methods aim at solving disparity for scenes with significant camera baselines and identical optics, whereas we focus on estimating the object shape with a small baseline and differing optics, as found in smartphone multi-lens setups. This presents a challenge, and Optical Flow becomes a crucial tool in addressing it.

Optical Flow refers to the estimation of pixel movements between consecutive frames in an image sequence, aiming to determine the corresponding position of each pixel in the next frame and thereby derive a motion vector field. It is commonly used in dynamic scenarios such as video analysis and object tracking. Optical Flow and Stereo Matching share certain computational similarities, particularly in finding matching points between images. Both require identifying the corresponding positions of pixels in another image, often relying on similar cost functions based on colour or texture similarity.

The core idea behind Optical Flow algorithms is the Optical Flow consistency constraint, which assumes that the pixel intensity of objects remains constant between consecutive image frames. This implies that the brightness of a pixel does not change as it moves from one location to another. The concept of Optical Flow was first introduced by Horn and Schunck in 1981 [HS81] when they proposed the first Optical Flow estimation algorithm based on a global smoothness constraint. This foundational work, along with the Lucas-Kanade algorithm of the same year [LK81], laid the groundwork for the estimation of optical flows by combining local and global optimization methods.

With the advent of deep learning, CNN-based Optical Flow estimation methods have significantly advanced the field. In 2015, FlowNet [DFI*15] was introduced as the first end-to-end deep learning model for Optical Flow estimation, capable of directly producing Optical Flow fields from two input images. This groundbreaking approach was further enhanced by FlowNet 2.0 [IMS*17] in 2017, which used a stacked architecture to achieve improved accuracy and generalization. In 2018, PWC-Net [SYLK18] became a milestone in Optical Flow research, introducing a pyramid structure with warping and refinement of cost volume to significantly improve computational efficiency and estimation accuracy. This evolution continued with the introduction of RAFT [TD20] in 2020, which used a grid-based correlation matrix and iterative updates to achieve state-of-the-art performance, providing dense pixel correlations and robust Optical Flow fields.

In addition to algorithmic advancements, the development of datasets and benchmarks has played a crucial role in improving Optical Flow estimation. The MPI-Sintel dataset [BWSB12] and the KITTI dataset [GLU12], both released in 2012, have become standard benchmarks, offering increasingly complex and realistic scenarios that challenge and enhance algorithm performance.

Although Optical Flow is not inherently designed to align pixels between two images of the same object, its ability to track object motion within the same background makes it highly adaptable. This capability allows Optical Flow to align images effectively and provides valuable cues for tasks such as depth estimation.

2.3.3 Diffusion Models

In recent years, there has been significant progress in the field of generative models, with Diffusion Models emerging as key players. Diffusion Models were first proposed by Sohl-Dickstein et al. in 2015 [SDWMG15] and represent a class of generative models based on Markov chains, which progressively add noise to data and then learn to reverse the process, generating data from noise. This process of 'adding noise' and 'denoising' resembles the physical diffusion process, hence the name Diffusion Models.

This generative approach, based on a step-by-step reverse process, has demonstrated extremely high quality when generating data such as images and audio. The generative process of Diffusion Models can be compared to other generative models such as Generative Adversarial Networks (GANs) [GPAM*14] and Variational Autoencoders (VAEs) [Kin13]. Unlike GANs, Diffusion Models do not rely on adversarial training, making them more stable and less prone to mode collapse. It is difficult to succinctly explain how diffusion models magically achieve generation from nothing in just two sentences. For more detailed content, please refer to the relevant sections in the appendix.

Although the theoretical foundations of diffusion models were proposed in the early days, practical applications faced challenges such as high computational complexity and slow sampling speeds, preventing their widespread adoption. However, since 2020, the rise of diffusion models can be described as a "superstar" story in the AI field. This is due to several technological advancements that significantly addressed these issues. In 2020, Ho et al. introduced the Denoising Diffusion Probabilistic Model (DDPM) [HJA20], which demonstrated exceptional performance in image generation tasks and is considered one of the most promising generative models today. Subsequently, in 2021, Nichol and Dhariwal proposed the improved DDPM++ model [ND21], further enhancing generation quality and efficiency.

In 2021, diffusion models surpassed GANs in generation quality for the first time. The research by Dhariwal and Nichol [DN21] demonstrated that diffusion models outperform GANs on evaluation metrics such as Inception Score (IS) and Fréchet Inception Distance (FID). This breakthrough prompted both academia and industry to re-evaluate diffusion models, leading to their rapid adoption in practical applications. Over the past two years (2022 and 2023), diffusion models have become one of the most prominent topics in deep learning research, experiencing rapid development. Stable Diffusion utilises Latent Diffusion Models (LDM) to significantly reduce computational demands by compressing the process into a latent space, achieving efficient and high-resolution text-to-image generation [RBL*22]. In super-resolution tasks, Saharia et al. [SHC*22] employed diffusion models to perform image super-resolution, restoring high-resolution images through gradual denoising while demonstrating superior performance in detail preservation and global consistency. At the same time, diffusion models

themselves have attracted researchers to further explore new possibilities, such as replacing Gaussian noise with blue noise [HSV*24] and other innovations.

However, it is important to note that diffusion models primarily operate in 2D space, as they were originally designed based on 2D image data. In 2D image generation, data is represented in the form of pixel grids. In contrast, in 3D generation, data representation becomes much more complex, and directly applying diffusion models to point clouds or voxels incurs very high computational costs.

Researchers have proposed numerous innovative ideas to address 3D tasks. Ben Poole et al. [PJBM22] introduced a method that leverages pre-trained 2D text-to-image diffusion models to guide the generation of 3D objects. By optimising the parameters of Neural Radiance Fields (NeRF) to match the multi-view 2D images rendered by NeRF with the outputs of diffusion models, they achieved the generation of high-quality 3D models from textual descriptions. Similarly, methods like [CGC*23] utilise mutually orthogonal triplanes to represent the triplane features of individual scenes, combined with a shared NeRF decoder and a triplane diffusion prior. This approach enables unconditional generation, single-view and multi-view reconstruction, demonstrating its versatility and effectiveness in 3D generation tasks. However, the 3D methods represented by these two examples have not yet fully addressed the direct generation of SVBRDFs or the creation of high-quality maps. As previously emphasised, diffusion models primarily handle 2D images or implicit geometric fields and have limited capability for modelling multidimensional physical properties. Generating SVBRDF maps requires maintaining physical consistency between different channels (e.g., reflectance, normal maps), which exceeds the current methods' capacity for texture generation.

Nevertheless, we have observed that a class of Conditional Diffusion Models offers a viable solution for SVBRDF tasks. By introducing conditional information (e.g., albedo maps, normal maps, or geometric models), conditional diffusion models can incorporate physical constraints during the generation process, enabling multi-channel collaborative generation. For example, DreamFusion [PJBM22] uses text conditions to generate multi-view consistent 3D scenes, while Palette achieves high-quality inpainting and colourisation tasks through conditional images [].

These studies demonstrate the feasibility of extending conditional diffusion models to complex texture generation tasks, such as SVBRDF maps, and highlight their potential in multi-input and complex generation tasks. This insight inspired the experiments and explorations in our work presented in Chapter 4.

Chapter 3

Deep Shape and SVBRDF Estimation using Smartphone Multi-lens Imaging

Prologue

Starting from this section, we will introduce the specific work we have completed. First, we focus on the appearance acquisition of everyday objects. Currently, image-based and deep-learning methods have made significant progress due to their ease of use and reduced hardware requirements compared to traditional methods. The appearance extraction of objects in daily life has gradually moved out of laboratories and into the hands of users, greatly facilitating individual artists in arranging realistic objects in their works. With this approach in mind, we explored how to use commercially available devices to quickly and easily achieve this goal for users, and found that the multi-camera imaging systems of modern smartphones offered significant benefits.

We present a method based on deep neural networks, utilising the multi-camera imaging capabilities of smartphones to acquire high-quality shapes and SVBRDF of 3D objects. Our method captures two images simultaneously using a smartphone’s zoom lens and wide-angle lens under natural lighting or the phone’s flash. By leveraging the subtle baseline and viewpoint differences provided by the two lenses, we extract the optical flow field to estimate fine-scale depth. The estimated depth is further used to guide the SVBRDF estimation, yielding superior results compared to existing single-shot methods. In this chapter, we also discuss the deployment of polarisation measurements on smartphones, which was an early research direction of our project.

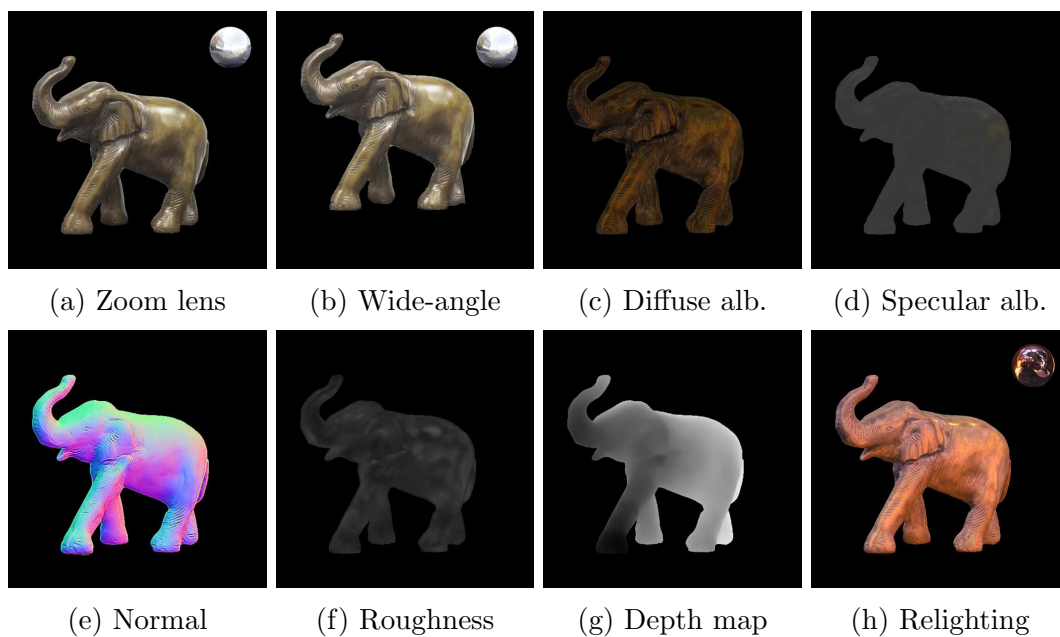


Figure 3.1: The estimated Bidirectional Reflectance Distribution Function (BRDF) and depth maps (c – g) of a wooden elephant are obtained from two input images (a, b) captured using a smartphone with a multi-lens imaging system. A probe is provided in the corners of input images to indicate the lighting conditions during the capture (not being used as an input). This enables realistic relighting rendering (h) under the Grace Cathedral lighting environment.

3.1 Introduction

Image-based shape and spatially varying reflectance estimation for 3D objects and materials have received significant research attention in recent years. The approach has shifted from traditional dense measurement methods towards making acquisition more practical, focusing on employing a minimal number of photographs while improving acquisition quality. Recently, deep learning-based methods that require minimal acquisition from a single viewpoint have gained popularity due to their ease of use [LXR*18, BJK*20, DLG21].

In this work, we propose a deep neural network-based method that estimates high-quality shape and spatially varying reflectance (SVBRDF) of 3D objects using smartphone multi-lens imaging. Our method requires only two input images with subtle viewpoint and perspective shifts, as captured by cameras on a smartphone. Modern multi-lens smartphones can acquire images with their two (or more) cameras simultaneously, making our method as effortless for acquisition as a single-shot method. Unlike previous work that utilises shading cues from flash illumination [LXR*18, BJK*20] or combines flash and polarisation cues [DLG21], our method relies on stereo information extracted from the two input images. This lifts the constraint on lighting conditions, enabling the method to work under either natural illumination or phone flash conditions. However, the multi-lens setup on the back of a smartphone typically has lenses only 2-3 centimetres apart, resulting in near sub-pixel disparity from corresponding features in the two images. Acquiring an accurate depth map with such subtle viewpoint and perspective shifts, alongside different optics, is extremely challenging, and even state-of-the-art deep-learning-based methods for stereo matching [XZ20, LTD21, LWX*22] fail to deliver a reasonable result. Instead, we utilise RAFT [TD20] to estimate a fine-scale optical flow field, using this optical flow field as the shape cue for a network trained with a depth-based rendering loss to estimate the depth map of 3D objects. The depth-based rendering loss enables surface awareness for the network during training, producing more accurate, continuous, and visually superior results in inference. We further use the estimated depth to guide SVBRDF estimation, and we train a network as proposed in [DLG21]. We also demonstrate superior results in both shape and reflectance compared to existing single-shot methods.

3.2 Method

3.2.1 Overview

As mentioned in the Polarisation section 2.2.3, the starting point of this project was inspired by the idea that combining polarisation signals with deep learning can achieve better SVBRDF estimation results compared to simply inputting a single image into a deep network. We conducted numerous experiments using smartphones to explore practical, convenient, and effective methods for polarisation capture. This discussion will be included in the first part of the Chapter, followed by the details of the relatively more optimal solution.

Our method aims to estimate both 3D shapes, in terms of depth and normals, and spatially varying reflectance for real-world objects from two input images taken under either natural or flash lighting conditions, using smartphone multi-lens systems. To address this highly ill-posed problem, we propose a two-stage method that first estimates depth from the subtle camera baseline and perspective differences resulting from the distinct focal lengths of the two images acquired by the smartphone’s zoom lens and wide-angle lens, respectively. This estimated depth is then used to guide the SVBRDF estimation in the second stage.

Inspired by the recent stereo matching method [LTD21], we also leverage the RAFT network proposed by Teed et al. [TD20] to establish dense pixel correlations and estimate the flow field, which serves as the cue for our depth estimation network. We retrained the RAFT network using our data to achieve the best quality results.

Our depth estimation network adopts the same architecture as proposed by Deschaintre et al. [DLG21], incorporating a novel depth-based rendering loss. Unlike the depth rendering loss recently proposed by Chang et al. [CBZ*22], which uses a NeRF-style volumetric depth rendering, we solely utilise the predicted depth as the input for rendering and subsequently transform this predicted depth into surface normals. Following the traditional Cook-Torrance model and rendering equation, computes the loss between the rendering driven by ground truth maps and the rendering results from the network output.

Figure 3.2 demonstrates our two-stage pipeline.

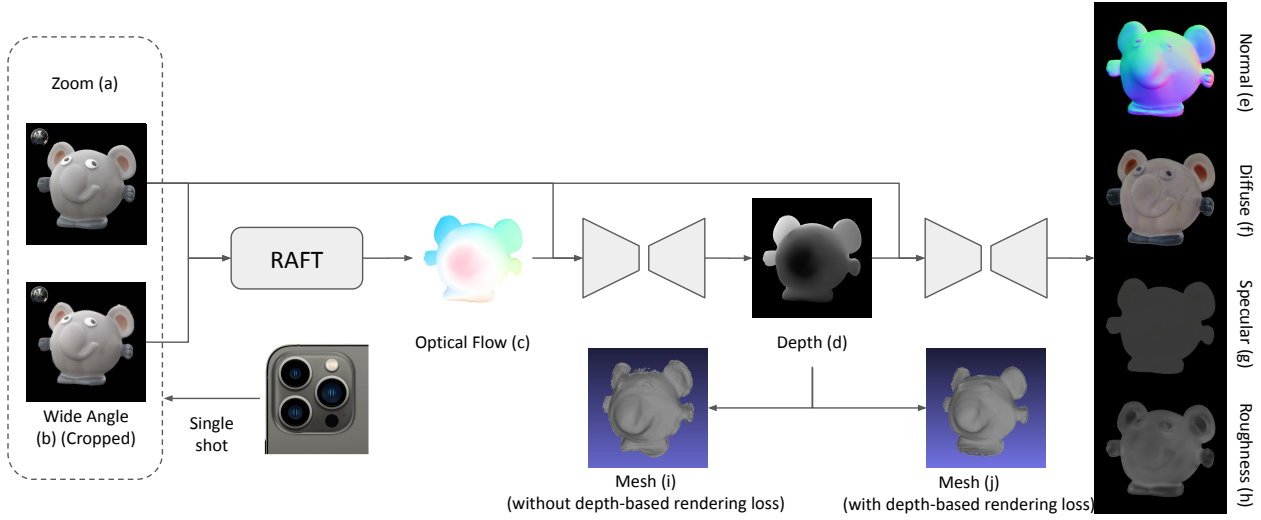


Figure 3.2: Pipeline of our method for estimating shape and SVBRDF for real-world 3D objects. Our pipeline takes two images (a, b) respectively taken by the zoom and wide-angle lenses of a smartphone. We use a retrained RAFT [TD20] to estimate the optical flow (c). We then feed (a, c) to a depth estimation network to obtain the normalised depth (d), and forward (d) together with (a) to the final SVBRDF estimation network to get the remaining SVBRDF maps (e-h). With the proposed depth-based rendering loss, we can qualitatively tell the mesh (j) converted from the estimated depth is smoother and more accurate than the one in (i) obtained from the depth estimated using a network trained without the depth-based rendering loss. At the same time, any details that might be lost at this stage can be partially restored during the subsequent enhancement step.

The key distinction from conventional rendering losses lies in the integration of depth gradient-based shading calculations into the rendering process. Surface normals are generated using gradient variations from the depth map, and BRDF parameters influencing the rendering results (such as roughness), are set to constant values based on standard practices in physically based rendering. Since the goal is to minimise the loss between the rendering results, other parameters have minimal impact as long as they remain consistent during the computation. This maximises the impact of the depth map on rendering results.

This newly introduced loss term encourages the inference of objects with greater surface continuity, thereby enhancing the network’s ability to estimate smoother and more realistic shapes for real-world 3D objects. Further details and the necessity of depth-based rendering loss will be discussed in the following Depth Estimation section.

Moving to the second stage, we feed the estimated depth to condition the SVBRDF estimation in the second network trained with our synthetic data. Again, we deploy a similar network as designed by Deschaintre et al. [DLG21]. We experimented with different inputs to the network and found that the network performs best when there is perfect pixel correspondence between input images. Therefore, we employ the zoom lens image as the main input image for both networks. We use it together with the optical flow field as input for the depth prediction network in the first stage, and we provide the zoom lens image along with the predicted depth map as input to the second network to predict the diffuse albedo, specular albedo, specular roughness, and normal maps as output. To provide higher-quality maps for subsequent rendering tasks, we enhanced the obtained normal map at the end.

We employ an iPhone 13 Pro to acquire the two images for all our experiments. It is worth mentioning that the wide-angle lens image is cropped, and the zoom lens image is down-sampled to match the resolution of the cropped wide-angle lens image. We chose the zoom lens image as the main input image due to its higher native resolution, which preserves more details. The details of each component of the solution will be further discussed in the following subsections.

3.2.2 Explore on Polarisation

We begin with early research approaches. The work of Deschaintre et al. [DLG21] represents the first acquisition method to utilize the combination of flash, polarisation, and deep learning, achieving superior results compared to other single-view capture methods without requiring a complex network architecture.

One of the reasons hindering the ease of use of this method is the relatively complex process of capturing polarisation signals. Deschaintre et al. employ a DSLR camera, a tripod and a linear polarizing filter mounted on the lens and manually rotate the polarizer on the lens to acquire the data. For the three different polarisation states required, the time is tripled compared to single capture methods.

Our goal is to achieve a practical, fast, and convenient capture process on mobile devices.

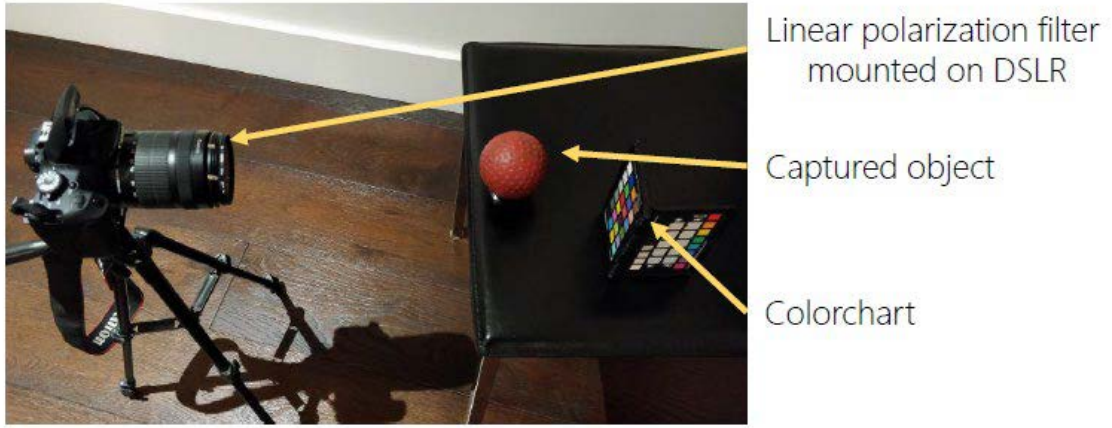
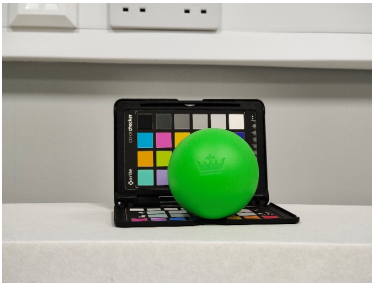


Figure 3.3: A schematic diagram of the capture setup in the Deschaintre et al. [DLG21]

The cameras on flagship smartphones nowadays have reached a very high level, with their resolution and imaging capabilities gradually approaching those of entry-level DSLR cameras. A natural idea is that if we can bring the concept of combining polarisation and deep learning to commercially available smartphones, it will significantly improve the usability of the capture method. Most of the latest smartphones are equipped with multiple lenses (for example, the iPhone Pro series has telephoto, wide-angle, and ultra-wide-angle lenses), which offers the potential for multi-lens collaboration. If we could use multiple lenses on a smartphone to capture images at different polarisation angles simultaneously, we might be able to solve this issue within the time frame of a single capture.

This approach requires figuring out how to enable the three lenses to capture images at different polarisation angles and then merge the data. Similar to how we would attach a polarising filter to a DSLR camera, we could add a polarising filter in front of the smartphone lenses to make them suitable for polarisation measurement. However, the first issue that arises is achieving uniform resolution. As shown in Figure 3.4, there is a significant difference in the resolution of the observed object captured by different lenses. If we want to use the Stokes map as one of the network’s inputs, the images from the three polarisation angles must have the same resolution and be pixel-aligned. However, in smartphones, the multi-lens imaging system is not designed for simultaneous operation, so the working ranges of several lenses (in this case, the distances at which they can successfully focus on the object) hardly overlap. The telephoto lens and the wide-angle lens can operate simultaneously at a distance of around 50 cm, but the telephoto

lens can hardly work with the ultra-wide-angle lens without manual camera adjustments. When attempting to bring the smartphone closer to the object to make the subject within the ultra-wide-angle lens appear less diminutive, the telephoto lens completely loses focus. Conversely, if the phone is moved farther away, the object in the ultra-wide-angle lens will appear extremely low resolution. Since we need to downsample and align the images to match the lens with the worst imaging quality, the overall image quality becomes very limited.



(a) the telephoto lens



(b) the wide-angle lens



(c) the ultra-wide-angle lens

Figure 3.4: Images obtained by three different lenses of OPPO Find X3 Pro

In our research, coordinating the use of three different cameras has proven to be extremely challenging. Besides the mentioned resolution issues, the third or additional cameras in flagship smartphones (we tested three different phones: the iPhone 11 and 12 series, as well as the Android phone OPPO X3) are often designed to assist the main camera in achieving a wider field of view or for close-up shots. These lenses are not intended to be used independently, and as such, their image quality is inherently poor. They are plagued with noise and colour calibration issues, which further contaminate the already weak polarisation detection results.

We also experimented with using a dual-camera system, placing a polarising filter with a horizontal polarisation axis in front of one lens and a filter with a 45-degree axis in front of the other lens. This allowed us to capture I_H and I_{45} in portrait mode. Then, by rotating the phone 90° (with the fixed polarising filters rotating along with the phone), we captured two more landscape photos. This time, we obtained the remaining I_V and I_{135} (Three polarisation angles are sufficient, with the fourth angle serving as an auxiliary role). Using the dual-camera system, we were able to find an overlapping working distance between the two higher-quality lenses, thus achieving better image quality. However, this added to the difficulty of aligning the images. We tried various traditional image alignment algorithms and used a checkerboard

pattern to help identify corresponding points for the projective transformation. However, since the alignment required for calculating the Stokes map needs to be extremely precise, even slight movements of the observation lens would hinder the Stokes map results, not to mention the significant differences caused by the change in such viewpoints.

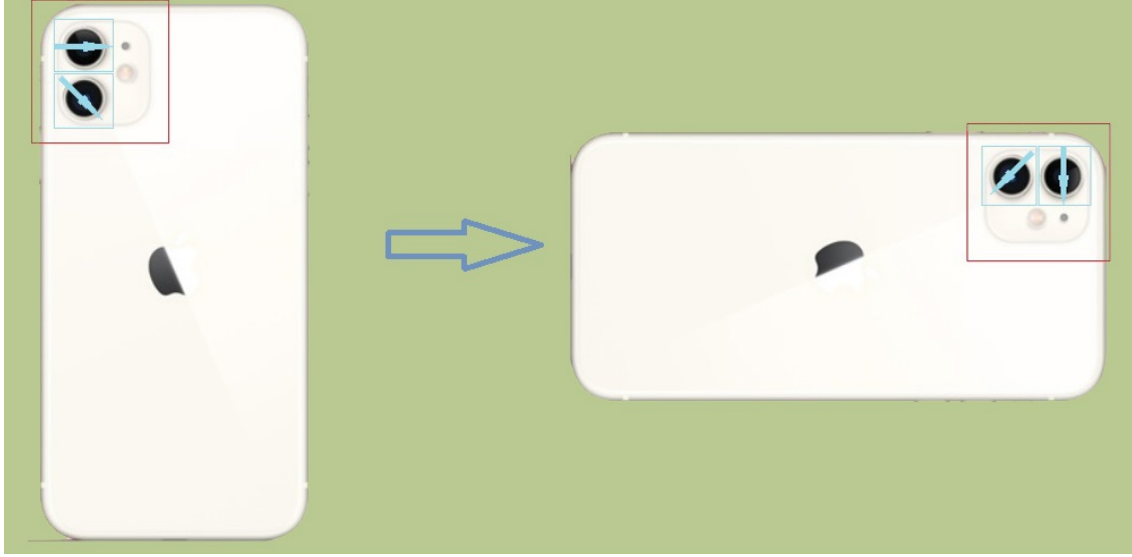


Figure 3.5: A diagram illustrating an image captured twice by rotating the camera to obtain photos at different polarisation angles. The direction of the optical axis of the polarising filter placed in front of the lens is indicated by blue arrows in the diagram.

Additionally, the stringent requirements of polarisation signals on shooting conditions are equally disappointing. Polarisation signals are weaker compared to ordinary optical signals, and in Deschaintre et al. [DLG21] cases, HDR images are required to ensure that the dark areas of the image are clearly visible without overexposing the bright areas. Since HDR image composition requires capturing a sequence of photos with different exposure times, this process takes some time, and the object and camera must remain stationary during the measurement. This is why Deschaintre et al. [DLG21] successful method requires the use of a tripod and fixed flashlights as lighting conditions, ensuring the highest possible image quality and simplifying the problem. When we attempted to remove these restrictions in our experiments, the results became highly unsatisfactory and unacceptable.

At this point, the attempt to use multiple lenses to capture polarised images and calculate the Stokes map as an additional input for the deep learning network proved unsuccessful. Afterwards, we attempted to bypass calculating the Stokes map during the data acquisition

process and instead fed the unaligned polarised images directly into the network to see if it could improve the results. During this exploration, we discovered that while the physical distance between the centres of the wide-angle and ultra-wide-angle lenses hindered pixel alignment, it also provided information we could potentially leverage. After further investigation, we abandoned the idea of using polarisation signals and freed ourselves from the constraints of specific lighting conditions.

3.2.3 Depth Estimation

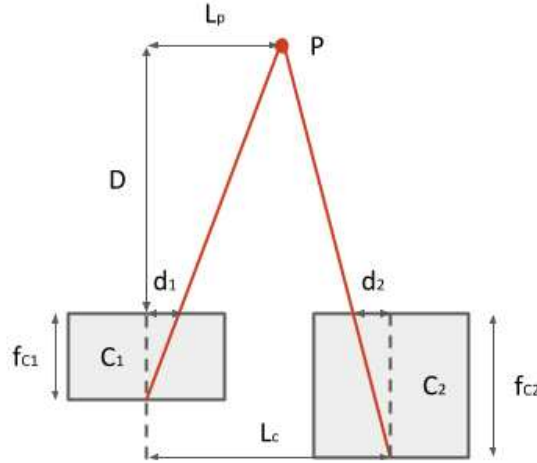


Figure 3.6: A simple camera model: displacement of 3D points becomes ambiguous for the depth in disparity calculated from two cameras with different focal lengths.

Returning to the method we ultimately succeeded in implementing, we begin with depth estimation. Our approach is still based on the idea of providing additional cues to the final SVBRDF network to enhance the overall outcome. Specifically, we successfully utilise the differences between two camera images to obtain depth information.

Consider the two main camera systems, where we have a camera C_1 with focal length f_{C1} and another camera C_2 at L_C apart, with focal length f_{C2} , and a point P at L_P away from the middle line of the camera C_1 , with depth D to the imaging plane of both cameras (see Figure 3.6). The disparity of P between two imaging planes can be calculated as $\Delta dis = |d_1 - d_2|$, where

$$d_1 = \frac{L_p}{D + f_{C_1}} f_{C_1} \quad (3.1)$$

$$d_2 = \frac{L_p - L_c}{D + f_{C_2}} f_{C_2} \quad (3.2)$$

With $f_{C_1} = f_{C_2}$, it can be easily shown that Δdis does not depend on L_P , making the depth estimation trivial. However, in our case the two lenses have different focal lengths, resulting in an ambiguity between L_P and depth D . On the other hand, since L_C can be as short as 10mm, and L_P is quite narrow as we are solving for fine-scale depth of 3D objects, Δdis becomes sub-pixel which is beyond the precision of feature correspondence in existing stereo matching methods. Hence the depth is not resolved with even state-of-the-art deep learning based depth estimation methods [XZ20, LWX*22].

In the problem stated above, we are facing two challenges:

- Establish feature correspondence at sub-pixel or near sub-pixel level and calculate the disparity Δdis .
- Disambiguate the displacement L_P and depth D in the disparity Δdis .

We tackle the first feature correspondence challenge by utilising an optical-flow network, namely, RAFT [TD20]. With its dense 4D correlation volumes and recurrent optical flow updates, we find it generates the best off-the-shelf results compared to other optical flow networks [DFI*15, IMS*17]. Simultaneously, benefiting from fewer restrictions on lighting conditions by optical flow (only requires adjacent input images to be captured under the same lighting conditions) we win the flexibility on lighting conditions. To further improve the results we retrained the RAFT net with our data, details of which are described in Section 4.2.5.

The trained optical flow net is an estimator of the ground truth optical flow f_o . We estimate the optical flow field that maps image I_1 to I_2 , so the estimated optical flow $\hat{f}_o$ has per pixel correspondence to image I_1 .

Next, to solve for the normalised depth, we train an estimator $\hat{F}_D$ that minimises $\mathcal{L} = \|\hat{F}_D(\hat{f}_o, I_1) - D\|_1$. However, without surface awareness, the estimator $\hat{F}_D$ often struggles to

capture the local geometric consistency, resulting in artefacts such as abrupt depth variations that disrupt the perceived smoothness of the surface. Although the global shape may still be preserved, the lack of fine-grained local detail undermines both the numerical accuracy and the visual quality of the output. To address this limitation, we redefine our loss function as follows:

$$\mathcal{L} = ||F_R(\nabla \hat{F}_D(\hat{f}_o, I_1)) - F_R(\nabla D)||_1 \quad (3.3)$$

where F_R is the rendering function based on the Cook-Torrance model as described in 4.2.1, and $\nabla f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$. This loss function incorporates the spatial gradients of the predicted and ground truth depth, enabling the network to focus on capturing local surface variations and geometric features.

By introducing surface awareness through gradient-based loss, the network not only achieves numerically more accurate depth predictions but also generates visually higher-quality results. This is because the gradient information constrains the network to respect the underlying surface continuity and local geometric structures, leading to smoother transitions and better alignment with the ground truth. Furthermore, the improved local consistency directly enhances the rendering process, as realistic surface features play a crucial role in accurate light interaction and overall visual realism.

3.2.4 SVBRDF Estimation

Similar to the network architecture proposed in [DLG21], our SVBRDF estimation network jointly encodes the two inputs: the estimated depth $\hat{D}$ and the cropped image I_w taken by the smartphone zoom lens, and separately decodes through three decoders: a diffuse branch that outputs the diffuse albedo ρ_d , a specular branch that outputs the albedo ρ_s and roughness α_r , and finally a normal branch that decodes to the normal map normal. Our loss function is defined as:

$$\mathcal{L} = \mathcal{L}_d + \mathcal{L}_s + \mathcal{L}_\alpha + \mathcal{L}_{\vec{n}} + \lambda_R \mathcal{L}_R \quad (3.4)$$

where $\mathcal{L}_d$, $\mathcal{L}_s$, $\mathcal{L}_\alpha$ and $\mathcal{L}_{\vec{n}}$ are L_1 loss for diffuse albedo, specular albedo, roughness and normal respectively, and $\mathcal{L}_R$ is the rendering loss $\|F_R(\rho_d, \rho_s, \rho_\alpha, \vec{n}) - F_R(\hat{\rho}_d, \hat{\rho}_s, \hat{\rho}_\alpha, \hat{\vec{n}})\|_1$. We have chosen not to incorporate the depth-based rendering loss here as testing revealed that loss does not yield improvements.

3.2.5 Training and Datasets

We modified RAFT[TD20] to output 32-bit vector files instead of the native 8-bit PNG images. We train separate networks for natural lighting and flash lighting with data specifically generated for each lighting condition respectively. Both networks were trained under 300,000 iterations to achieve convergence results. The training process took approximately 40 hours on a single A5000 card. The initial parameter settings were selected based on the original RAFT implementation and adapted to our specific task. These included a batch size of 8, an initial learning rate of 10^{-4} with a cosine annealing schedule, and the AdamW optimizer. Weight decay was set to 10^{-2} to prevent overfitting, and gradient clipping was applied with a maximum norm of 1.0. The zoom lens image and wide-angle image were centred and cropped, and their resolution was reduced to 512*512 to serve as inputs for the network to manage the training memory efficiently.

For training our two-stage networks, we require a large dataset of objects captured with smartphone multi-lenses under different lighting environments (including flash), along with ground truth labels for diffuse albedo, specular albedo, specular roughness, normal map, depth and optical flow field. Acquiring such a dataset would be extremely challenging and costly, hence we leverage synthetic data to create two datasets of over 150,000 and 5,000 sets of images respectively for natural lighting conditions and flash lighting. Our synthetic training dataset under natural lighting conditions is rendered using a combination of 50 complex meshes that well represent real-world 3D objects, 10 lighting environments that cover typical indoor and outdoor scenes, and 400 different materials (SVBRDFs). We further augment our lighting by rotating each environment map by $\pi/4$ and $-\pi/4$ along the x-axis, resulting in 30 environment maps. Our test dataset uses 10 meshes, 5 lighting environments and 30 materials not included

in the training data. For our flash-only datasets, we use a frontal point light source to generate all images. We fully simulate the perspective difference due to different focal lengths and small baselines as commonly seen on modern smartphone cameras. To better preserve fine details in both the optical flow field and depth map, we save the optical flow field as a 32-bit vector file and the depth map in a 16-bit PNG.

3.2.6 Surface Detail Enhancement

Due to the depth estimation being based on optical flow and the correspondence between the two views, our method may result in slightly blurred surface normals. In the context of BRDF maps, high-frequency surface details are primarily captured by the diffuse albedo map and the normal map. However, when the normal map is slightly blurred, some high-frequency surface details may be attributed to the diffuse albedo map instead of the normal map, as the diffuse albedo inherently contains texture and colour variations that can act as a proxy for surface detail.

To address the limited surface detail in the normal maps, we employ a detail enhancement step similar to that proposed by [RPG14]. We apply $\mathbf{x}$ and $\mathbf{y}$ gradient filters to the estimated diffuse albedo. We then add these gradients with appropriate scaling to the $\mathbf{x}$ and $\mathbf{y}$ components of the estimated normal in the tangent space and renormalize the normal. Finally, we convert the modified normal back to world space. This process allows us to recover some high-frequency information in the normal map (Figure 3.7).

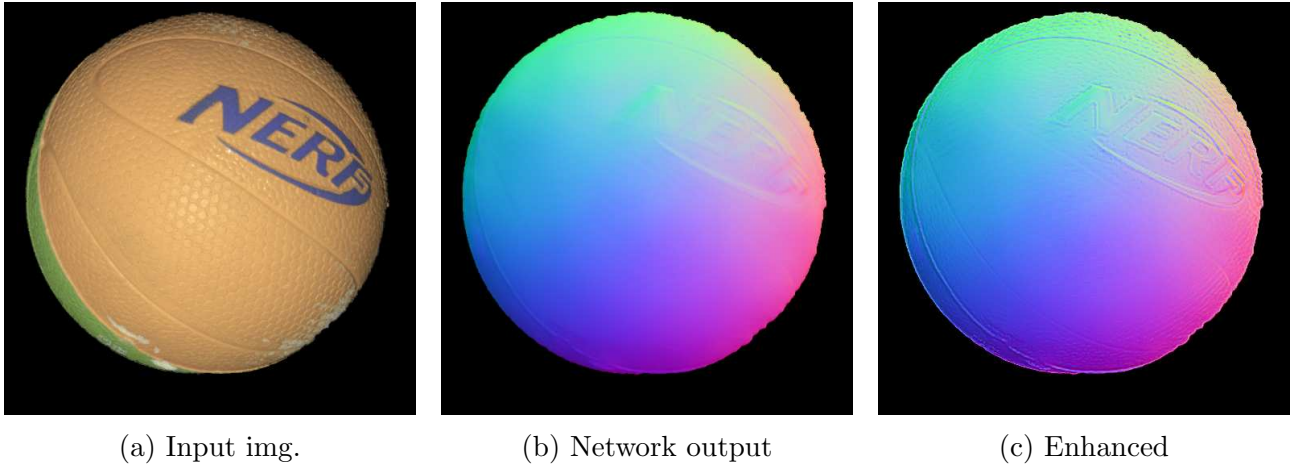


Figure 3.7: An example of surface detail enhancement of normal map of a sponge ball. (b): the normal map directly get from network. (c): enhanced normal map.

Although the enhanced details are not exact, they provide reasonable surface detail in the normal maps and improve the visual quality of the reconstructed surfaces for rendering purposes. We employ both the original predicted normal and the enhanced normal as diffuse and specular normal respectively for rendering using a hybrid normal rendering procedure similar to that proposed by [MHP*07].

3.3 Results

3.3.1 Optical Flow

Figure 3.8 shows a qualitative comparison of the estimated optical flow between the native RAFT net and the retrained net using our data. After retraining, the estimated optical flow is of high quality, and it acts as the ambiguous shape cue for the depth estimation net to infer the detailed depth.

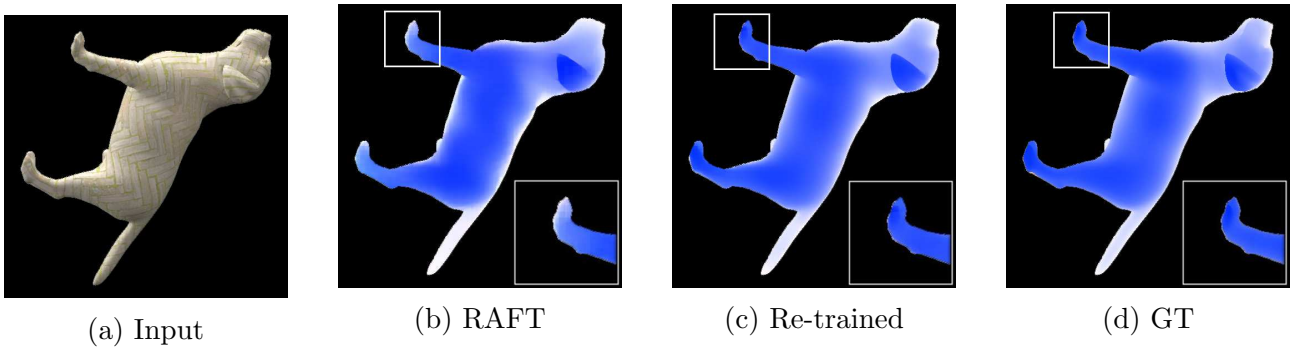


Figure 3.8: Comparison of optical flow quality: 3.8b estimated optical flow from native RAFT [TD20]. 3.8c estimated optical flow from retrained RAFT using our data. The content in the white square is enlarged in the lower right corner to highlight the difference

3.3.2 Depth and SVBRDF

We retrained three state-of-the-art stereo depth estimation networks [XZC*23, LTD21, LWX*22] using the same data as we used for training our network. Figure 3.9 shows comparison results. While these networks work great on datasets with large baselines and other extra constraints, e.g. x-axis consistency, they all fail to deliver good results for depth estimation in our specific setting.

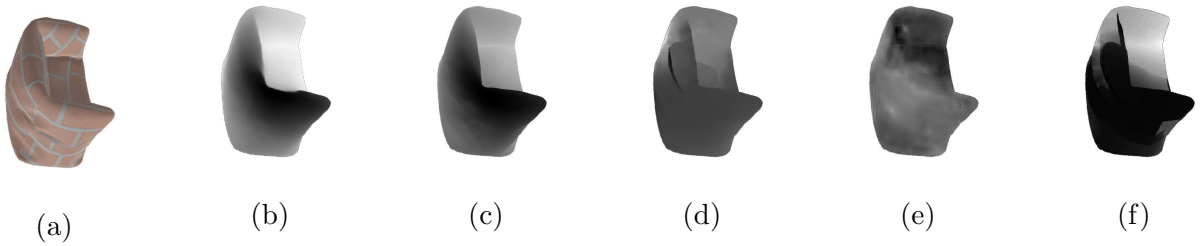


Figure 3.9: Comparison of depth quality: (a) Input image, (b) Our result, (c) Ground Truth (GT), (d) results from Unimatch [XZ20], (e) results from CRNnet [LWX*22], and (f) results from RAFT-Stereo [LTD21]. Compared to existing methods (d), (e), and (f), which exhibit noticeable artifacts and deviations from the GT in (c), our result (b) reducing major errors observed in other methods and improved alignment with the overall structure of the GT.

In Figure 3.10, we demonstrate a few real-world 3D objects, where the first three rows are captured under a natural lighting condition, while the rest of the examples are captured with the phone flash (including one same test objects for comparison under natural vs flash illumination). Figure 3.1 is an additional example that has been acquired under natural illumination. In total, we present four sets captured under natural lighting conditions (2 indoors and 2 outdoors) and five sets captured with flash illumination. These examples demonstrate that our shape-from-optical-flow method ensures robust depth inference under different incident lighting conditions. The estimated depth information further regulates the estimation of the SVBRDF, leading to consistent quality results across the showcased examples.

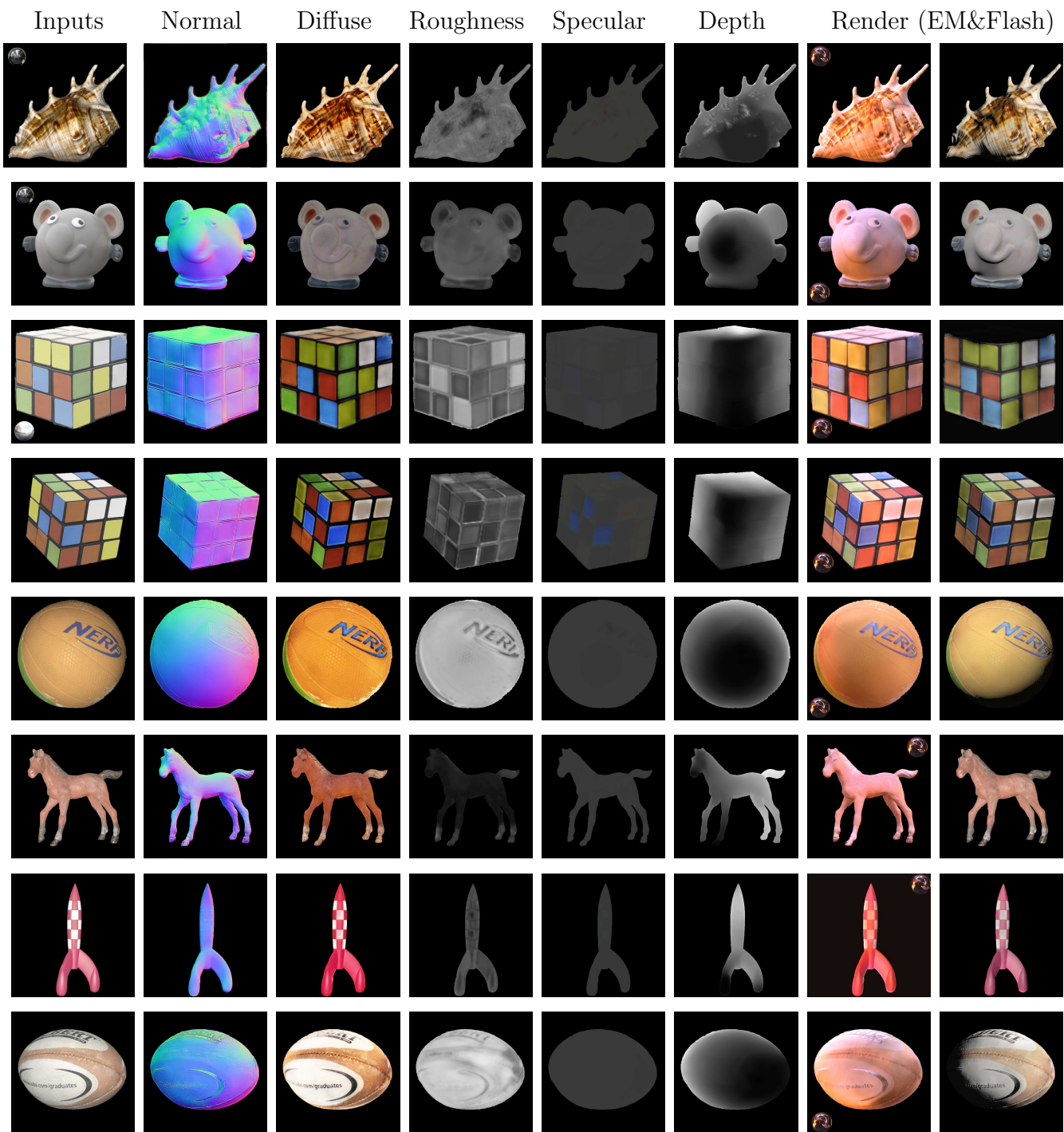


Figure 3.10: We tested our method on various real-world 3D objects under both natural lighting conditions (first three rows) and flashlights (last four rows). Our estimated shape and reflectance maps generate high-quality rendering results under both environmental illumination (Grace cathedral), and flash illumination.

The results indicate that the proposed optical flow-based shape reconstruction method achieves robust depth inference under varying lighting conditions. It maintains consistent result quality in both natural light and flashlight scenarios, demonstrating good adaptability. Notably, the high-quality normal maps, diffuse maps and realistic rendering reconstruction results.

However, the limitations of this method need to be mentioned. One major issue is the lack of material diversity in the training dataset. Specifically, the dataset lacks materials with independent diffuse and roughness characteristics, particularly those with significant spatially varying specular reflectance and more complex geometries. This limitation results in dependency artefacts between roughness and diffuse colour, where the network cannot fully decouple these two parameters in certain cases. Additionally, while the method can generate relatively smooth specular reflectance maps for materials with uniform optical properties, it may oversimplify the representation of specular reflectance in materials with pronounced spatial variations. This phenomenon mainly arises because the training data predominantly features object surfaces with repetitive single-material coverage. Artefact issues become more pronounced in challenging regions, such as adjacent coloured blocks on a Rubik’s cube.

Figure 3.11 shows some rendering comparisons of acquired objects to photographs under flash illumination from two different (novel) directions. Here, we selected the sponge ball shown in Figure 3.10(d) acquired under flash illumination, and the toy rubber elephant shown in Figure 3.2 acquired under natural indoor illumination for the validation against photographs under novel flash illumination conditions. It can be seen that our method correctly produces shading caused by the direction of light on the surface of the object, such as near the trunk of the toy elephant.

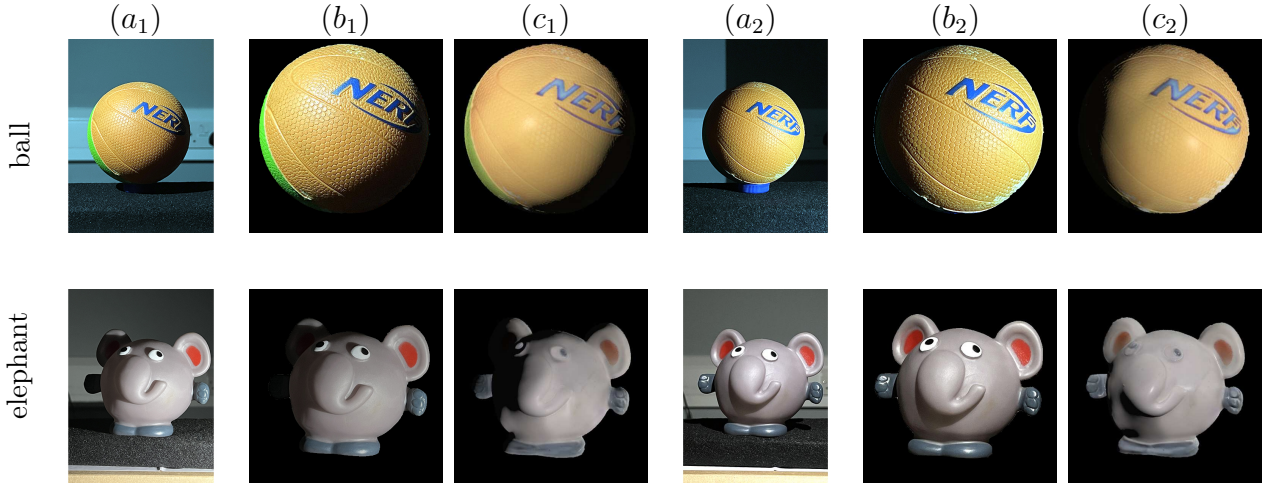


Figure 3.11: Relighting comparisons of acquired 3D objects under novel flash illumination conditions that were not employed during the measurement process. This figure demonstrates the ability of our method to render 3D objects under unseen lighting conditions. The sponge ball was acquired under frontal flash illumination, while the toy elephant was acquired under natural illumination. Photos taken by a smartphone camera under the novel flash angles are provided as ground truth for visual comparison. Specifically, (a_1, a_2) are photos captured under new lighting conditions, representing the ground truth (GT) for comparison. (b_1, b_2) are centred and masked versions of the photos. (c_1, c_2) are relighting render results generated using our method, simulating novel flash illumination conditions. For the sponge ball, the relighting renderings correspond to a flash positioned at 45 degrees top-left (c_1) and 45 degrees bottom-right (c_2). For the toy elephant, the relighting is performed for a flash positioned from the right (c_1) and directly above (c_2). This figure demonstrates the effectiveness of our method in reproducing plausible 3D relighting effects under new illumination conditions.

The rendered results show strong consistency with the corresponding ground truth photos, capturing the material and surface properties with a high degree of accuracy. While minor discrepancies may be observed in some details, the overall quality of the results highlights the robustness of our approach.

We also illustrate our SVBRDF estimation results on synthetic data to directly compare to the ground truth. We do not apply the enhancement method to normal in this comparison, ensuring that the comparison focuses solely on the network’s raw output, providing a more logical and intuitive evaluation of the network’s performance. Figure 3.12 indicates our results preserve high-frequency details very well in both shape and reflectance, and renderings using our estimated maps are close to ground truth renderings.

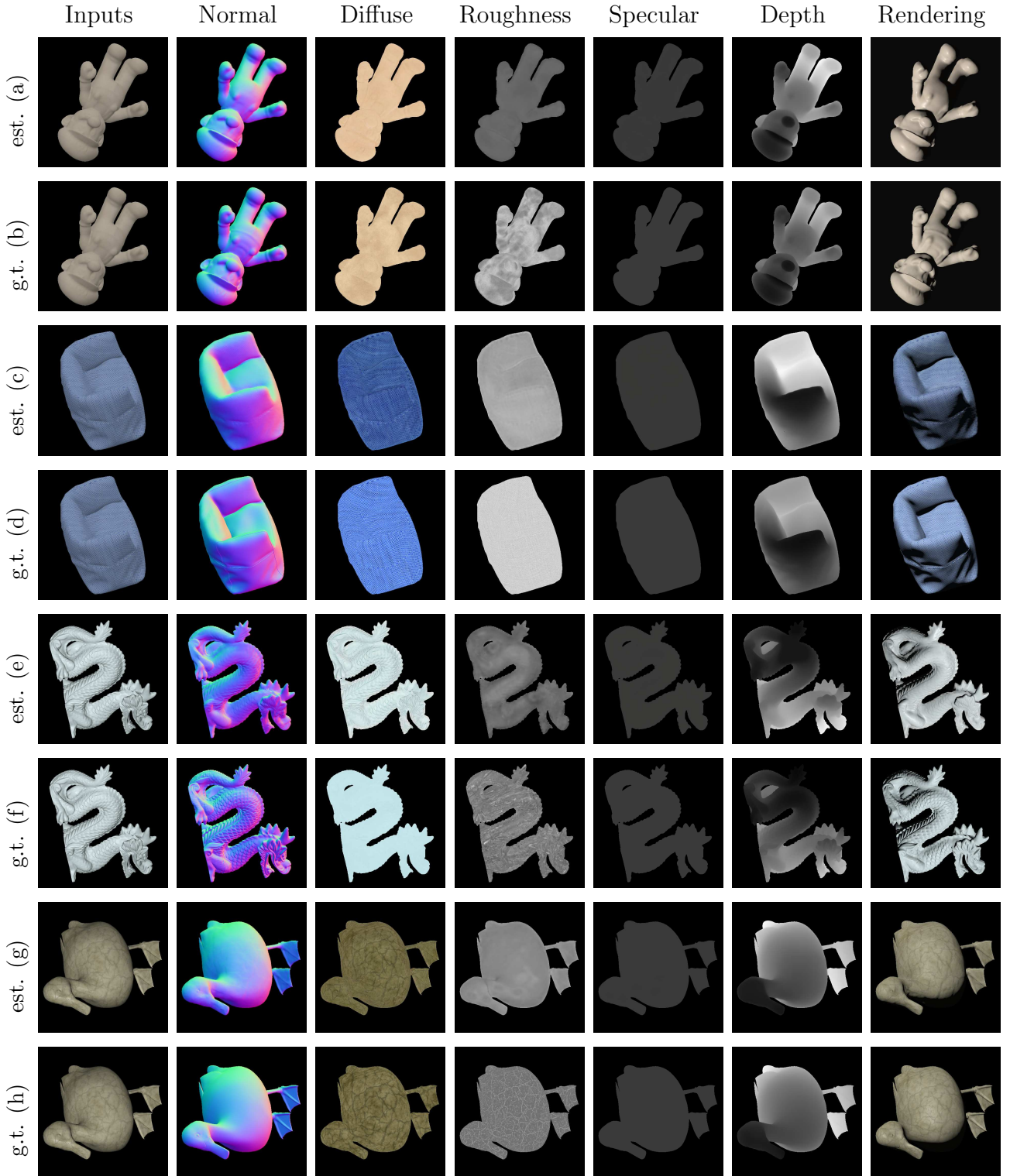


Figure 3.12: We compare our estimated results (a, c, e, g) to g.t. (b, d, f, h) on various synthetic 3D objects not seen during training, under both natural lighting conditions (a, c) and flashlights (e, g). Our estimated maps are accurate in comparison to g.t. maps, and renderings are close to g.t. renderings. We focus on qualitative comparisons through rendered images, which better demonstrate how the predicted maps perform in practice. Rendering results provide an intuitive way to evaluate the consistency of both shape and reflectance properties compared to the ground truth, directly reflecting the visual impact of the network’s predictions.

3.3.3 Qualitative Comparison

We compared our approach with previous successful methods in the next few sections. The flash-based method proposed by Li et al.[LXR*18], requires a single photo captured under flash illumination. And the method proposed by Boss et al.[BJK*20], relies on two photos: one taken under flash and ambient lighting, and the other under indoor ambient lighting only. To be clear, these two methods and our multi-lens methods are all single-view methods. Taking advantage of the multi-lens capabilities of smartphones, our method has an acquisition cost comparable to the single-input method of Li et al., and lower requirements than Boss et al.’s flash + no flash pair.

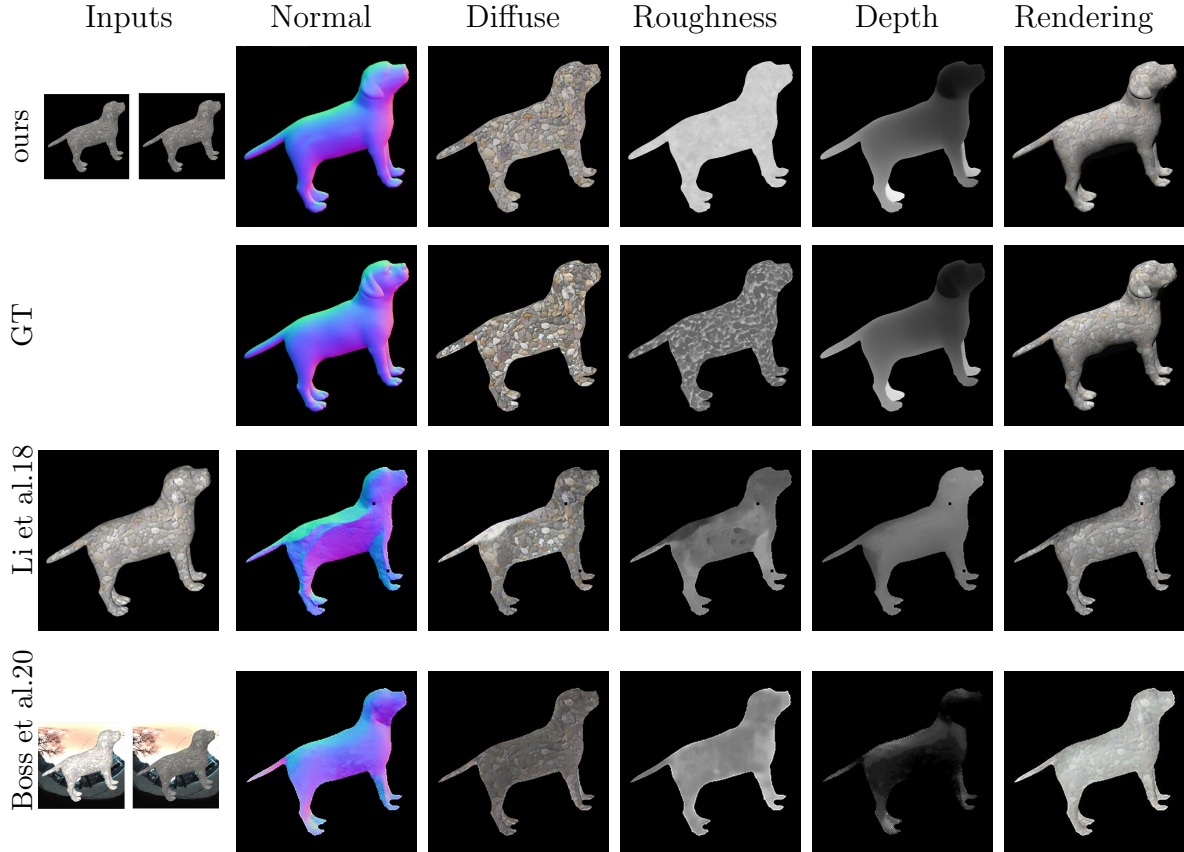


Figure 3.13: We compare to the flash-based method of Li et al. [LXR*18] and the two-shots method of Boss et al. [BJK*20] on a synthetic example. Our estimated maps are closer to g.t. maps than results of [LXR*18, BJK*20], and our renderings are also a better match to the g.t. The method proposed by Li et al.[LXR*18] does not provide specular outputs; therefore, we did not include this comparison.

As can be seen from the comparison Figure 3.13 on some synthetic results, our predicted reflectance and shape maps are qualitatively superior resulting in rendering that is much closer to the ground truth than these previous methods that mostly rely on flash illumination cues.

In Figure 3.14, we compare our results to the method of Li et al. [LXR*18] on two real objects captured with a phone flashlight. The results are consistent with our comparison and analysis of synthetic data which shows our method provides more accurate overall shape and BRDF estimation, while the method of Li et al. [LXR*18] suffers from significant low-frequency bias in the estimated depth and normals.

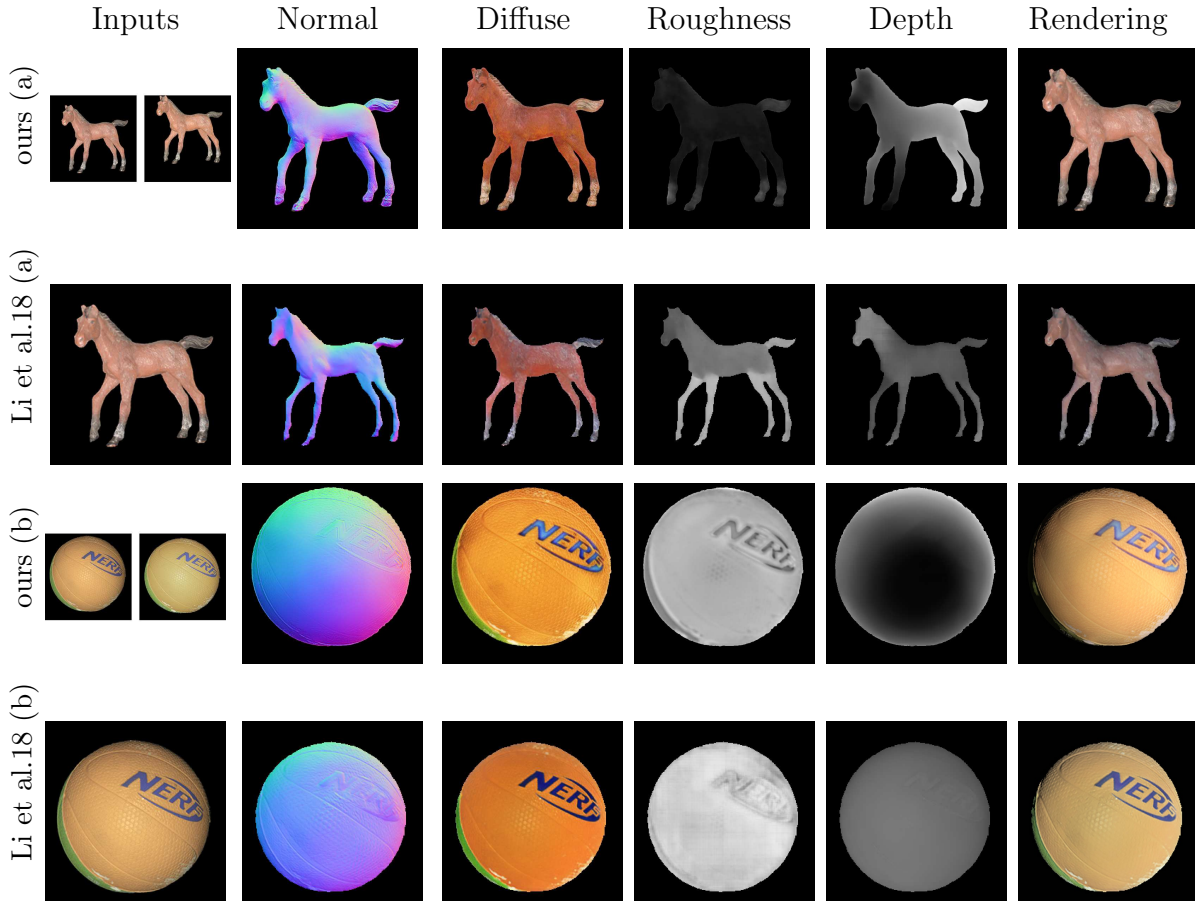


Figure 3.14: We compare some real objects to the flash-based method of Li et al. [LXR*18]. Overall, our approach has a much more correct global shape and does not suffer from low-frequency bias and our rendering results are qualitatively better as our shape estimation is superior. Specifically, for the ball case, Li et al.’s results limited the 3D fidelity of the estimated normals. The normals resemble a flattened disc rather than accurately reconstructing the ball’s curvature, which compresses the numerical range and exaggerates the appearance of finer details. In contrast, our method preserves the ball’s 3D geometry, resulting in a more faithful representation of the true surface shape. Again the method proposed by Li et al.[LXR*18] does not provide specular outputs; therefore, we did not include this comparison.

We also present a qualitative comparison to the method of Boss et al. [BJK*20] for an object acquired in outdoor conditions (Fig. 3.15). Boss et al.’s method strongly relies on flash illumination besides ambient illumination. However, due to the brighter outdoor lighting conditions, the flash illumination is dominated by the ambient illumination. This results in very little difference between the two input images for their method and no strong flash cues which are required by their method. In comparison to results such as the Rubik’s Cube shown in Fig. 3.10, the method of Boss et al. failed to provide reliable results in terms of normal and depth in Fig. 3.15. It is evident that such methods that strongly rely on flash illumination cues are not designed to handle scenarios involving strong ambient illumination, which limits their effectiveness in natural illumination conditions. In contrast, our method is able to handle both flash and natural illumination conditions due to depth estimated using multi-lens imaging.

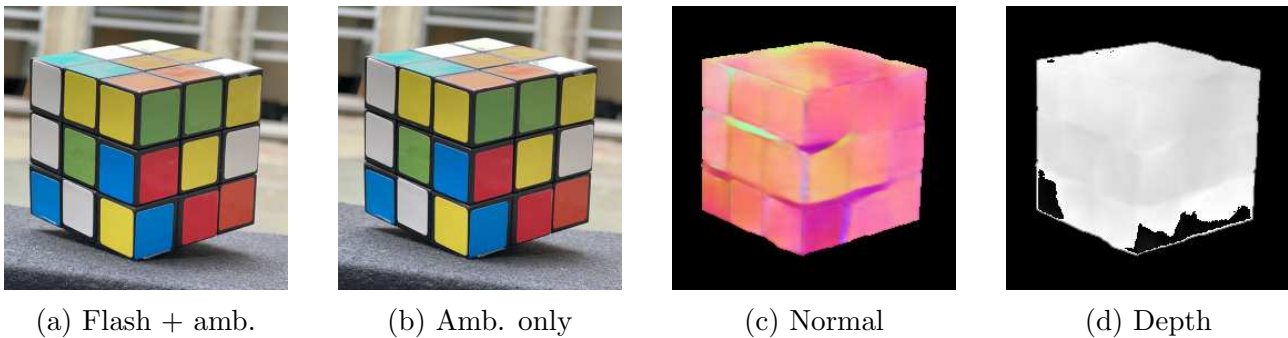


Figure 3.15: A failure case for Boss et al.’s method [BJK*20] under strong natural illumination. Under strong natural lighting, it becomes difficult to distinguish between input images captured with (a), and without (b) flash. The absence of flash cues in the input results in large errors in estimated surface normal and depth.

3.3.4 Quantitative Comparison

In table 3.1 we compare quantitatively to Li et al. [LXR*18] and Boss et al. [BJK*20] on synthetic data using L1 difference. We evaluate the error on the normal maps, depth and renderings instead of SVBRDF maps, as different BRDF models have been chosen by the different methods. This quantitative evaluation is performed on 250 sets of synthetic input images, consisting of renderings produced using 5 meshes, with each randomly rotated 5 times, and 10 SVBRDF. The rendering error is computed over 20 renderings for each result with varying light properties.

	Li et al.	Boss et al.	Ours (Flash)	Ours (EM)
<i>Depth</i>	0.2964	0.2763	0.0473	0.0350
<i>Normal</i>	0.1667	0.1637	0.0952	0.0888
<i>Renderings</i>	0.1252	0.2092	0.0610	0.0690

Table 3.1: Quantitative comparison on synthetic data using L1 differences. Numerically we perform significantly better compared to Li et al. [LXR*18] and Boss et al. [BJK*20] on both flash data and natural illumination data.

Numerical in Table 3.1 suggest both our flash-light and natural light results are significantly superior to Li et al. [LXR*18] and Boss et al. [BJK*20]. Our method shows slightly better rendering results on flash data as there is a shading cue from the flash that helps with the SVBRDF estimation, while the method works slightly better on shape estimation with natural lighting. This is because objects are usually better lit under natural lighting conditions, leading to better performance of optical flow. Note that for all quantitative and qualitative comparisons on synthetic data in the chapter, we only employ the normals predicted directly by the network and do not perform the detail enhancement step on the normals. While angular error is often preferred for evaluating normal maps due to its geometric interpretability, we report results using the L1 for consistency with other evaluations in this work.

3.4 Limitation

Our method is limited to objects of a certain size, wherein the zoom lens captures the entire object, and the distance to the object is typically within 50 cm. We trained the network on specific material data, which predominantly consists of dielectric material samples with a few metal examples. Hence, the network may not perform well on other types of scenes or subjects, such as faces (which are influenced by subsurface scattering) or predominantly metallic objects that were not included in the training data (see Figure 3.16). Metallic objects can also present challenges for optical flow due to bright saturated highlights and texture-less regions. Currently, HDR imaging is required for optimal results, as saturation in images can interfere with optical flow computation, thereby compromising the results.

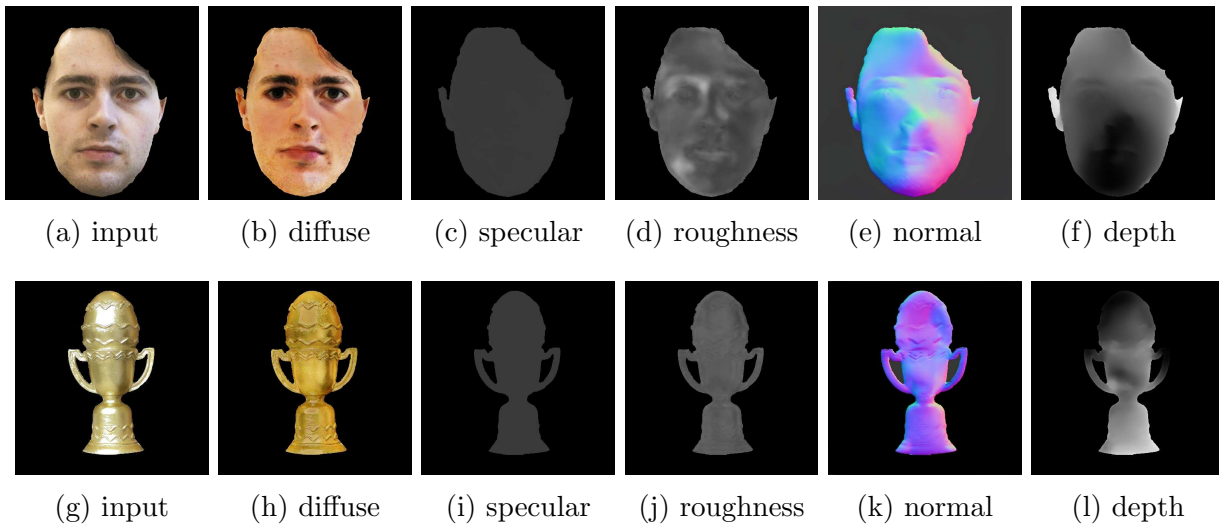


Figure 3.16: Some partial failure cases. Since our network has been trained on dominantly dielectric materials, it doesn't work that well on other types of scenes/subjects such as a face (top row), or a dominantly metallic object (bottom row).

Our network tends to predict slightly blurry surface normals, with high-frequency details in an object's appearance more likely to be interpreted as texture in the diffuse albedo. This occurs because the surface normals are constrained by depth estimated from optical flow-based correspondence between two views, which may be imperfect in texture-less regions. Additionally, due to the limited material diversity in the training dataset, the network struggles to decouple roughness from diffuse colour in some cases, leading to artefacts and oversimplified representations in areas with complex specular materials and geometric details (e.g., adjacent colour

blocks on a Rubik’s cube). To address these issues, future research could focus on expanding the dataset to include more samples with independent optical and geometric properties. Furthermore, incorporating edge-aware loss terms into the network or adopting data augmentation strategies could improve the method’s performance in high-contrast edge regions and areas with complex geometric details.

During the surface detail enhancement step, the gradients are used to enhance the normal in the tangent space with appropriate scaling. The ”appropriate scaling” is determined manually and is largely based on visual evaluation and artistic judgment. This introduces a level of subjectivity to the process, as the scaling factors are adjusted to achieve visually pleasing results rather than being derived from a strict mathematical or physical basis. While effective in many cases, this approach has limitations. For example, it may not perform well on surfaces with highly intricate geometries or under extreme lighting conditions. Furthermore, the reliance on manual adjustment makes it challenging to automate the enhancement process fully.

3.5 Conclusion

In this work, we propose a novel deep learning-based method that utilises smartphone multi-lens imaging to estimate shape and SVBRDF for real-world objects. Initially, we employ optical flow to estimate the flow map between two images. These images are captured with subtle perspective and viewpoint shifts. We then use a trained U-Net with a surface rendering loss to estimate depth from the optical flow map. Finally, the estimated depth is fed, together with the target image, into a third network to estimate the SVBRDF and surface normals. Our method is relatively robust to incident lighting conditions, making it suitable for use under both natural and flash illumination. We have demonstrated that our object SVBRDF and surface normal estimation results are generally superior to other state-of-the-art single-image methods that rely solely on flash illumination. This performance advantage underscores the benefits of leveraging multiple lenses in the imaging setup, particularly when dealing with complex, irregularly shaped objects. In the next chapter, we delve into specific applications involving facial data and explore how leveraging minimal views can enhance reconstruction quality.

Chapter 4

Two-shot Shape and SVBRDF

Reconstruction of Human Faces with Albedo-Conditioned Diffusion

Prologue

Following our discussion on extracting the appearance of everyday objects in the previous section, we now turn our attention to the human head. Reconstructing a relightable 3D human head has been a long-standing research problem. Much like traditional methods applied to objects, most approaches require intricate hardware setups and multi-viewpoint capture under controlled lighting conditions. Humans, unlike inanimate objects, find it difficult to remain perfectly still, and the repeated flashing of lights can be particularly bothersome for the subjects. This underscores the value of lightweight measurement methods.

In this section, we introduce the work we have completed under this framework. We present a two-step deep learning method that directly predicts the shape and spatially varying Bidirectional Reflectance Distribution Function (SVBRDF) of a subject’s face from two images taken from each side of the face. We enhance the quality of SVBRDF prediction by initially predicting the diffuse and specular albedo in image space, then learning a generative model that generates the entire SVBRDF parameters in UV space. We generate the 3D geometry of the subject by directly learning a 2D position map, thus eliminating the need for a pre-learned morphable model. Unlike single-image facial reconstruction methods, our approach obtains clear measurements on both sides of the face using two images. Our method surpasses state-of-the-art techniques in rendering faces at extreme angles and providing texture maps that are directly usable in most rendering systems.

4.1 Introduction

High-quality facial capture has been an important research topic with numerous practical applications. Custom digital facial avatars obtained from facial captures are required in various industries, including film, video gaming, teleconferencing, and more.

While high-quality facial captures can be reliably obtained in controlled studio environments [GRB*18, BBB*10, RGB*20], these methods require complex hardware setups that are not easily accessible. State-of-the-art relighting works have trended towards using high-end light stages for One-Light-At-A-Time (OLAT) captures [MHP*19, BLS*21, SSS*24]. Acquiring an OLAT representation of reflectance with lower-grade setups remains a non-trivial challenge [LLG24].

Consequently, methods that reconstruct a facial avatar using only a few in-the-wild images have gained popularity. This trend is further encouraged by advancements in deep learning, driving state-of-the-art results in facial avatar reconstruction with as few as images. Liu et al. [LCC*22] have proposed a deep learning pipeline for direct facial reconstruction and registration to a template using a multiview data capture process. Simpler diffuse-specular separation methods have been proposed with binary spherical gradient illumination [KZG18], which has also been used to create a 3D facial BRDF dataset comprising various subjects [LMG*20].

While current deep learning methods using sparse view or single-image reconstruction can achieve remarkable results in novel view synthesis and exhibit interesting interpolation results in their feature space, some of these methods rely on an implicit neural field as their representation [ZZSC22, HPX*22, YZM*24]. This Neural Radiance Field representation, similar to OLAT representations, does not provide explicit BRDF parameters required in most rendering pipelines. This limitation reduces the practicality of these methods, as rendering these facial models cannot be directly accomplished with standard rendering engines. Furthermore, these methods often struggle to render the subject at more extreme angles, such as viewing from the chin direction of the target face, resulting in unpredictable artefacts obstructing the face.

On the other hand, while research in facial reflectance capture has primarily focused on ad-

vanced capture setups, facial geometry acquisition has aimed to reduce reliance on specialised systems and minimise the number of required photographs. Traditionally, the two key attributes, reflectance and geometry, have been treated separately. Common approaches to handling facial geometry often rely on 3D morphable models (3DMMs), such as [BRZ*16, BV99], trained on large datasets, as priors for facial geometry. These models have proven capable of delivering satisfactory facial reconstructions from a single photograph [YSN*18, LMG*20, LMP*23]. However, handling reflectance and geometry separately introduces additional complexity for users.

Our work aims to overcome these shortcomings by generating a subject’s facial scan represented by a 3D mesh and a set of SVBRDF maps. To achieve better rendering quality at extreme angles, we capture two images of the subject, each capturing one side of the face. We propose a two-step process that leverages these two photographs to directly generate a set of SVBRDF texture maps, as well as a position map that can be mapped into a 3D mesh. The resulting facial scan provides higher reconstruction and rendering quality compared to state-of-the-art single-shot methods while being directly usable in most rendering engines.

Our main contributions can be summarized as:

- A two-step pipeline that directly predicts the shape and SVBRDF of a subject given two images, negating the need for pre-learned morphable models or volumetric neural fields.
- A generative formulation of 3D mesh reconstruction that uses the popular 2D diffusion models to generate 3D facial geometry.
- A fine-tuning method for denoising diffusion models that further improve the quality of the generated shape and SVBRDF maps.

4.2 Method

4.2.1 Overview

We propose a fully deep learning-based two-stage method based on denoising diffusion models [HJA20]. Our method requires only two photographs of the subject, capturing the left and right sides of the face. With these two photographs, we obtain both the 3D shape and SVBRDF maps of the subject, enabling the creation of high-quality 3D facial avatars from a relatively simple capture process and achieving realistic relighting results.

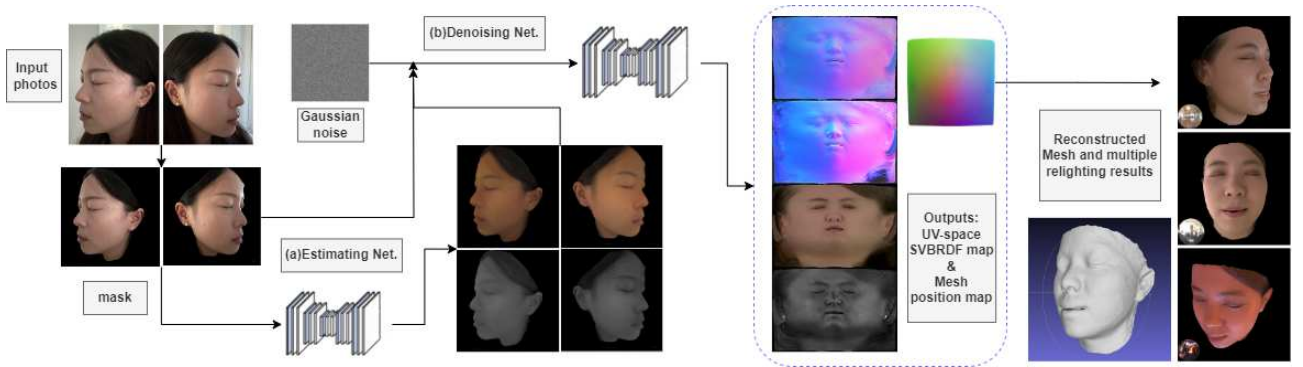


Figure 4.1: Overall pipeline of our proposed method which requires two photographs of each side of the face as input. Our method first predicts the image-space diffuse and specular albedo with a light-weight estimation network, then uses the diffuse and specular albedo images, as well as the original photographs, as guidance to train a denoising diffusion network that predicts the shape and SVBRDF maps in UV-space. Our results enable 3D rendering of the given subject under any desirable environment maps.

Our method first predicts the diffuse and specular albedos in image space. The predicted image-space maps are then concatenated with the captured photographs to serve as guidance for a denoising diffusion model. The diffusion model is subsequently trained to generate the 3D shape and SVBRDF maps in texture space, given the guidance of the captured photographs and image-space albedo maps. Figure 4.1 demonstrates our two-stage pipeline. Each step of the pipeline, along with the process of generating training data, will be explained in the following subsections.

4.2.2 Diffuse and Specular Estimation

We designed an ultra-lightweight network for estimating diffuse and specular albedo pre-diffusion, thereby offloading the heavy processing from the diffusion network. This approach achieves superior spatial consistency in SVBRDF UV maps, as discussed in Section 4.3. Leveraging the normalised rendering used in our data generation, as described in Section 4.2.5, we constructed our network without any normalisation layers. This design choice allows training and inference with input images of dimensions $1024 \times 1024 \times 3$ using standard hardware. We incorporate an attention mechanism utilising additive attention gates [OSF*18], resulting in sharper estimated albedos compared to simple concatenation through additional skip convolutions, as noted by [LLLG24]. Figure 4.2 illustrates the diffuse and specular estimation network architecture.

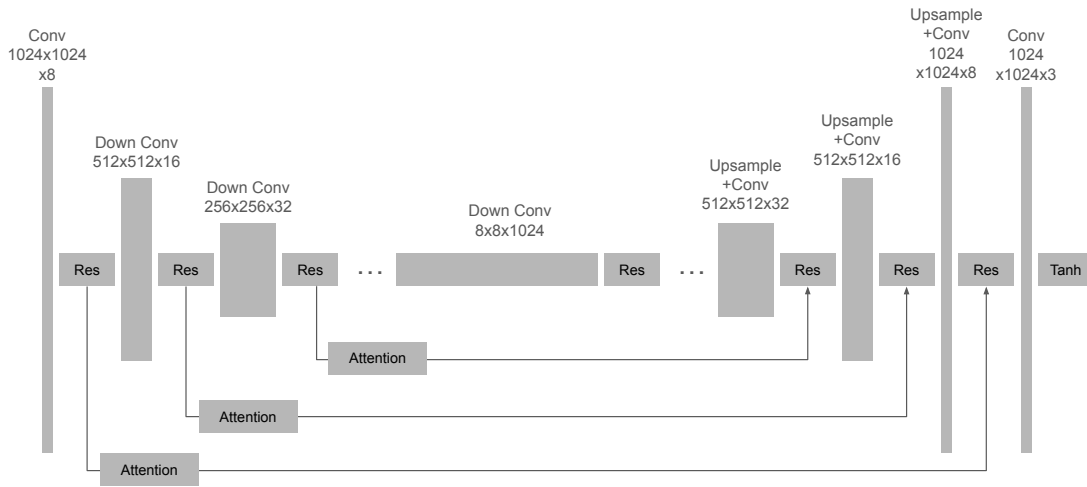


Figure 4.2: Graphical illustration of our simple diffuse and specular estimation network.

Contrary to using the diffusion network for both UV unwrapping and albedo estimation, this pre-diffusion step operates in image space, which is more normalised. By "more normalised," we refer to the use of image space where pixel coordinates are directly mapped, avoiding distortions or inconsistencies introduced during UV unwrapping. This approach simplifies the albedo estimation problem and generally enhances the overall robustness of our model to various

lighting conditions. This will be further demonstrated in ablation studies in Subsection 4.3.3.

4.2.3 Shape and SVBRDF Generation

The second stage of our method uses a diffusion model to generate shape and SVBRDF given the estimated specular and diffuse albedo maps of the subject. The shape of the subject is represented as a 2D position map in UV space. Our model follows the UNet implementation from improved diffusion [ND21], where the model outputs are a series of concatenated UV maps representing shape and SVBRDF. The estimated image-space specular and diffuse maps are directly concatenated to the input noisy UV maps to serve as guidance to the denoising diffusion process. Our training process uses a linear noise scheduling proposed in α (de)Blending [HBC23]. Figure 4.3 shows the internal structure of our diffusion-denoising network. The model is trained with data synthesized and rendered from RealFaceDB dataset [LMG*20](more details in 4.2.5).

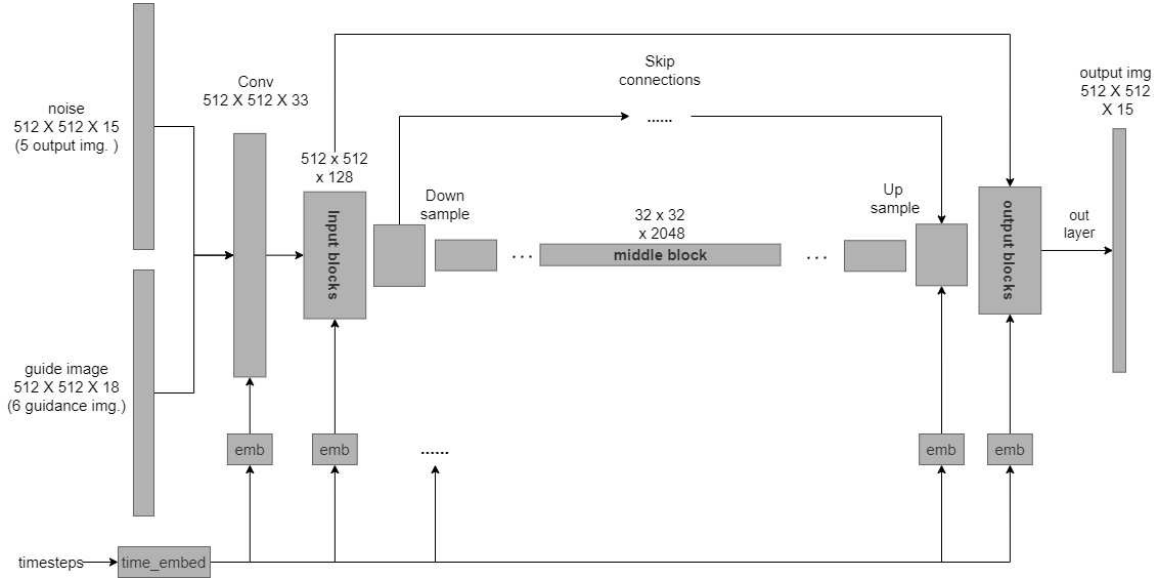


Figure 4.3: Graphical illustration of our shape and SVBRDF generation diffusion denoising network. The guidance images are concatenated to the noisy shape and SVBRDF maps. The timestep is a value between zero and 1 and is passed into all blocks with time embedding except the final output layer.

The directly trained diffusion model guided by image-space diffuse and specular maps learns to generate a rough facial shape and SVBRDF of the given subject. However, certain fine-

tuning strategies are necessary to attain higher-quality maps, which is essential for subsequent rendering tasks. In the next section, we introduce our fine-tuning pipeline that further improves the quality of the results from our trained diffusion model.

4.2.4 Model Fine-tuning

After the early stages of training, the diffusion model learns to generate sufficiently good-quality UV maps from Gaussian noise with a multiple-step denoising diffusion process. Once this has been achieved, we then introduce three fine-tuning methods to directly optimize the final generated result. Specifically, during the fine-tuning phase, we perform an iterative denoising diffusion process (we take 8 iterative steps in practice) to obtain a denoised UV map sample, then we compute the fine-tuning loss functions on the denoised sample and finally use the accumulated gradients to back-propagate and update the diffusion model. We find that the fine-tuning steps are only effective when introduced in the later stages of training since the denoised samples are too noisy in the early stages of training.

Render Loss

For our generated SVBRDF maps, we introduce a render loss to further improve the quality of the generated maps. The render loss has been shown to improve both the predicted SVBRDF maps as well as the rendered results [DAD*18] [DLG21] [FLG23]. The render loss also establishes connections between the relatively independent SVBRDF maps, enhancing the consistency of the maps we obtain. We formulate our render loss as follows:

$$L_r = \left\| F_R(\rho_d, \rho_s, \vec{n}_d, \vec{n}_s, \vec{l}) - F_R(\hat{\rho}_d, \hat{\rho}_s, \hat{\vec{n}}_d, \hat{\vec{n}}_s, \vec{l}) \right\|_1 \quad (4.1)$$

where ρ_d and ρ_s represents the diffuse and specular albedo, while $\vec{n}_d$ and $\vec{n}_s$ represents the diffuse and specular normal vectors. $\vec{l}$ is the given lighting direction. The hat notation (e.g., $\hat{\rho}_d$) indicates the predicted values of these parameters. For our render loss, we randomly

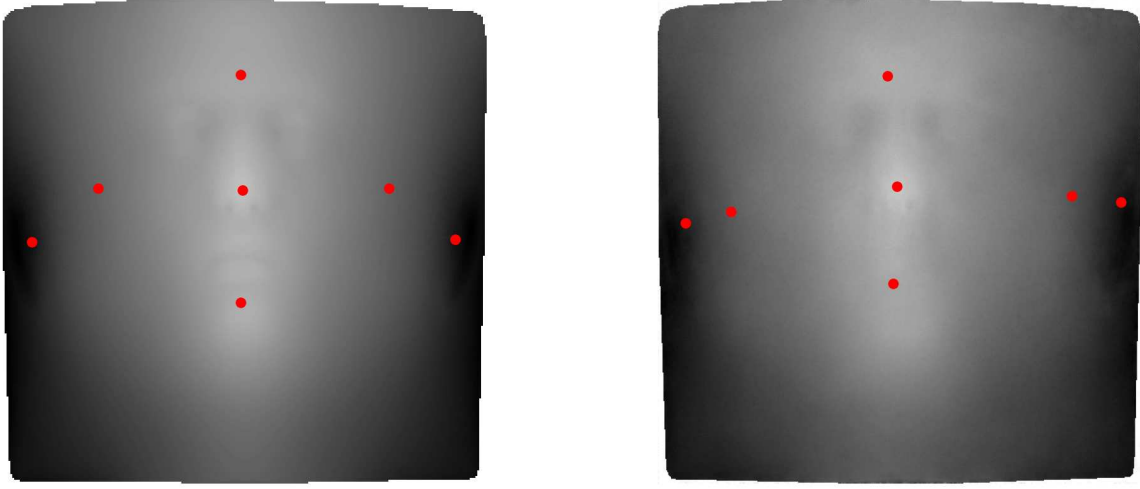
choose a lighting direction during each evaluation. The BRDF model F_R is implemented as a conventional Cook-Torrance model. We chose the L_1 norm because, in our experiments, the differences between L_1 and L_2 norms in terms of the final results were negligible. L_1 was selected primarily for its simplicity in this specific implementation, without compromising the overall quality or consistency of the results.

Our rendering directly operates on the unwrapped UV map, and the rendering result is driven by the four maps we obtained, using a differentiable rendering engine. The resulting rendered output is also in the flat UV space. We also attempted to render directly into the camera space using the predicted results and calculate the loss against the input image to enhance identity preservation. However, this approach faced several challenges, including the lack of pixel-aligned results due to random rotations of input images, the need for additional alignment networks that increased complexity and memory usage, and discrepancies between the differentiable rendering used for loss calculation and the non-differentiable OpenGL rendering engine. Moreover, the method’s reliance on uniform lighting conditions further limited its applicability under unknown lighting environments. Therefore, we ultimately applied the render loss in the UV space, where these issues could be avoided.

Considering the disadvantages mentioned above, the approach of directly calculating the render loss against the input photo fell short in terms of simplicity, applicability, and effectiveness. Therefore, we opted for the most straightforward rendering calculation method for the render loss term to optimize the texture map results.

Landmark loss

For the unwrapped position map, we designed a lightweight landmark key-point detection method based on the local maxima of the position map. By calculating the gradients between pixels, we can identify landmark key points such as the nose tip, chin, forehead, the widest positions on the left and right sides of the face, and the highest points of the cheeks. These key points can be detected reliably using the gradients of the position map and characterize the facial geometry specific to each subject. Figure 4.4 shows an example of the distribution of



(a) Ground Truth.

(b) Generated Map.

Figure 4.4: An example showing the locations of key points found by landmarks detection. The figure shows the unwrapped position map as a grayscale image, and the key points found are represented by red dots. (a): the key points found by applying the landmark detection on one ground truth map. (b): the key points detected on the generated position map in the later-stage of training.

key points in our simplified facial geometric landmark detection method.

We opt for this simple implementation instead of using additional advanced facial landmark detection networks [CZGB23] since these networks operate on facial images rather than position maps. Our key-point detection method maintains a lightweight overall process and provides robust landmark detection that helps with fine-tuning facial geometry.

With landmark key-point detection, we can then propose a landmark loss that computes the distances between the key points on the ground truth and the predicted position maps. The landmark loss improves the mesh quality and ensures that the generated mesh contains the specific characteristics of a given subject that deviates from the average facial geometry. The landmark loss can be formulated as:

$$L_{\text{lmk}} = \sum_{k=1}^K \left\| \vec{d}_k - \hat{\vec{d}}_k \right\|_2 \quad (4.2)$$

where $\vec{d}_k$ represents the ground truth coordinate vector for the k -th key point, and $\hat{\vec{d}}_k$ denotes the predicted coordinate vector. The Euclidean norm $\|\cdot\|_2$ calculates the distance between these vectors, quantifying the spatial error for each key point. This loss is directly applied to the unwrapped position map, rather than to the rendered facial results, allowing it to focus solely on the surface geometry without being affected by other factors involved in the rendering process. Since the SVBRDF map and the mesh are pixel-aligned, it also ensures that there is no misalignment between the mesh and texture.

Smoothness loss

In addition to the landmark loss, we also introduce a smoothness loss to propagate the influence of the key points on the surrounding pixels. The smoothness loss function also acts only on the position map and tries to ensure continuity and consistency in the resulting map by calculating the horizontal and vertical differences between adjacent pixels in the position map:

$$L_{\text{smooth}} = \frac{1}{N} \sum_{i,j} (|I_{i,j} - I_{i+1,j}| + |I_{i,j} - I_{i,j+1}|) \quad (4.3)$$

where $I_{i,j}$ represents the position of the pixel (i, j) in our position map, and N represents the total number of pixel pairs considered in the calculation.

The smoothness loss calculates the absolute differences between each pixel of the output position map and its immediate neighbourhood (in both horizontal and vertical directions). By minimizing the differences, the loss function effectively reduces high-frequency noise in the image and enhances the similarity of pixel values around key features, creating a smoother final mesh.

With the introduction of the 3 fine-tuning loss functions, we can formulate our loss function used in the fine-tuning stage:

$$L_{\text{fine-tune}} = L_{\alpha} + \lambda L_r + \lambda' L_{\text{lmk}} + \lambda'' L_{\text{smooth}} \quad (4.4)$$

Where L_α is the loss function used for naive α (de)blending, λ , λ' , and λ'' are manually selected weights controlling the importance of each of the fine-tuning losses.

4.2.5 Data Generation

In this subsection, we introduce the process of generating the training data used to train our two-stage network.

Data for Diffuse-Specular estimation network

We created a training set consisting of image-space renderings of faces from 80 subjects, acquired using the setup described in [LLK*22], under 10 environment maps. To mitigate the risk of overfitting, we augmented our training data by randomly rotating each environment map ten times, with the face rendered from three principal viewpoints (frontal, and two sides). Each rotated environment map was normalised to have a diffuse integral of 1 at normal incidence. These augmentations effectively expand the diversity of the training set, creating the equivalent of 24,000 unique samples. This data is sufficient to train our first-stage lightweight network.

Data for Shape and SVBRDF Generation

Directly acquiring a large dataset of subjects with shape and SVBRDF maps and in-the-wild images is infeasible. Instead, we chose to render synthetic data using the RealFaceDB dataset [LMG*20] for training the second-stage denoising network. This dataset contains diffuse albedo, specular albedo, diffuse normal, and specular normal maps, as well as a registered mesh with defined UV parameterisation for over 200 individuals with a wide range of expressions.

The RealFaceDB dataset was obtained using a combination of polarised LED sphere illumination and multi-view facial capture methods. The acquisition setup includes a polarised LED sphere with 168 lights and nine DSLR cameras. Half of the LEDs on the sphere are vertically

polarised (for parallel polarisation), while the other half are horizontally polarised (for cross-polarisation), arranged in an alternating pattern. RealFaceDB employed the polarised-based method proposed in [GFT*11], along with the LED colour space analysis method from [KZG18], combined with the multi-view facial capture technique from [LWZG19]. The captured geometry and reflectance data were then aligned to a universal topology with consistent UV mapping for all subjects, ensuring that all geometry and reflectance data were registered according to the same standard.

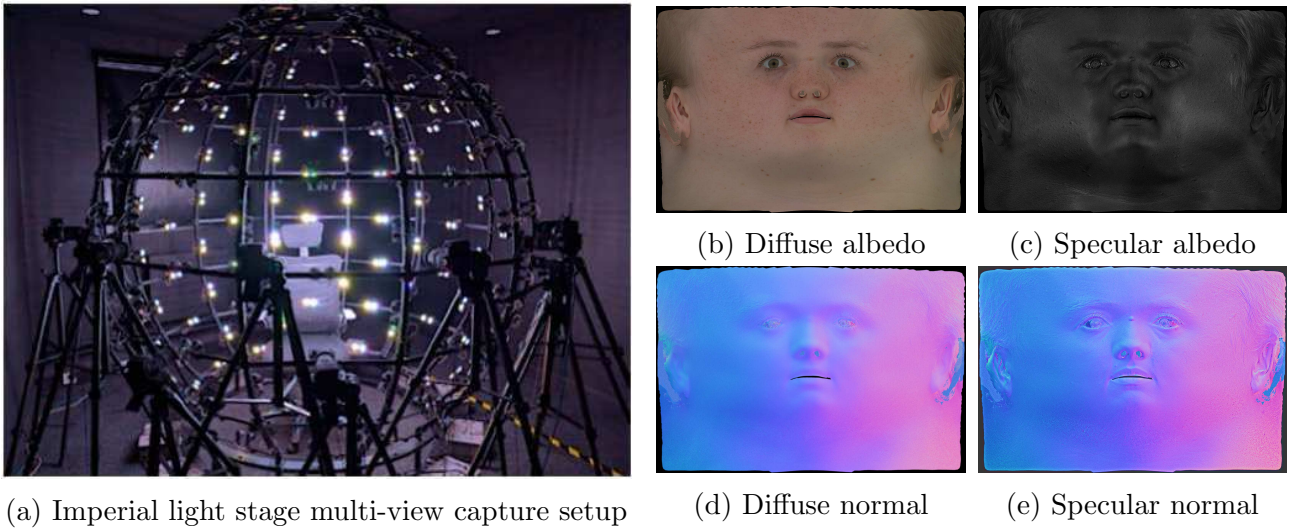


Figure 4.5: An illustration of the capture setup used by the RealFaceDB and (b-e) is one reference sample from the RealFaceDB.

To render in-the-wild images for training, we selected 20 well-lit environment maps covering both indoor and outdoor scenes. The environment maps were further augmented by rotating each environment map by $\pi/4$ and $-\pi/4$ along the longitudinal axis. All subjects were rendered under each environment map, with two images capturing both sides of the face per environment map. The images were rendered at angles of -45 and $+45$ degrees to the side, with a random tilt and rotation of 5 degrees applied to the camera position. This process resulted in a total of 8100 facial data pairs, with two well-lit images corresponding to a set of ground-truth shapes and SVBRDF maps. Due to the limitation of computational (GPU) resources (one Nvidia 4090 card), we down-sampled the SVBRDF maps to a resolution of 512×512 . However, it is important to note that our method is not constrained to any specific resolution and can operate at any necessary resolution, given sufficient GPU resources.

Unwrapping Base Mesh to 2D

The base meshes in the RealFaceDB dataset provide the 3D facial geometry of the subjects. However, these meshes are not directly usable within our 2D diffusion pipeline. Instead, we use the texture coordinates of each vertex of the mesh to establish a mapping between the 3D coordinates and the UV texture space. We then map the XYZ coordinates of the base mesh to the RGB channels on a 2D image, thus obtaining the unwrapped position map of the base mesh. As the base mesh lacks high-frequency details (which are more effectively stored in normal maps) and the number of vertices is limited, the data points on the unwrapped position map are relatively sparse. We therefore interpolate between the sparse vertex coordinates to generate a dense unwrapped position map with smooth variation throughout the image. After unwrapping, the position map is pixel-aligned to the SVBRDF in the UV space, and the process is fully reversible without any loss of information. This allows us to utilise the relatively mature 2D diffusion model to simultaneously learn the SVBRDF maps as well as the shape of the subject. Figure. 4.6 shows one example of an unwrapped 3D mesh.

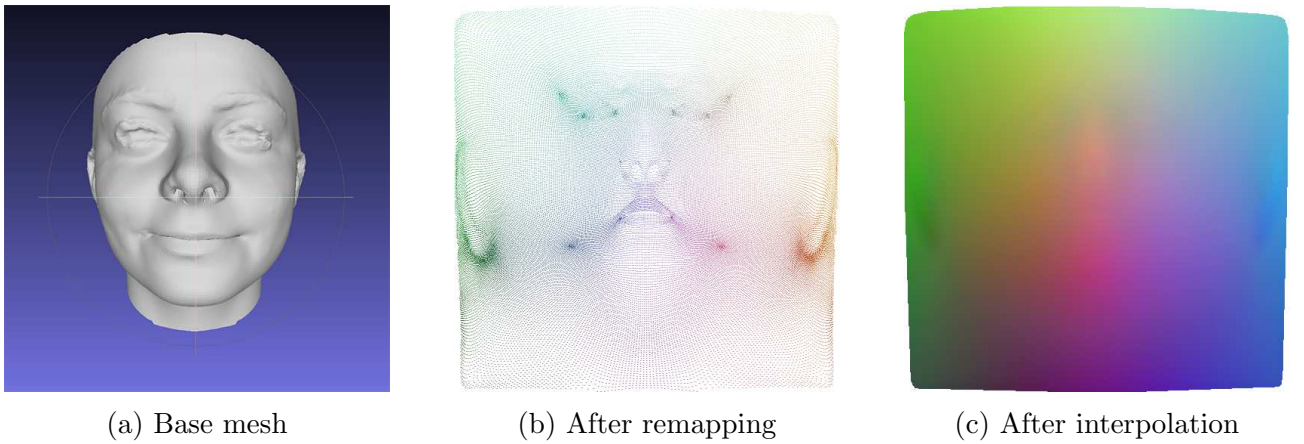


Figure 4.6: UV unwrapping of a base mesh in the RealFaceDB dataset. (a): Basemesh visualization. (b): UV unwrapped image after remapping positions of vertices with texture coordinates. (c): Dense UV position map obtained with linear interpolation of (b).

Test dataset

To create a dataset for evaluation, we manually selected 15 representative individuals from the RealFaceDB dataset, aiming to include as much diversity as possible in terms of gender and

skin tone within the available data. Six additional environment lighting maps were used to render the images for the test dataset. We ensured that the individuals and lighting conditions in the test dataset were not included in the training data.

4.3 Results

In this section, we elaborate on the model’s outcomes from multiple perspectives. Specifically, we compare our rendering results and the generated UV maps with those of existing state-of-the-art methods on two different publicly accessible datasets. We also demonstrate the results under general lighting conditions encountered in real-world settings to show the robustness and end with ablation studies to validate the effectiveness of various components of our method.

4.3.1 Test Dataset Evaluation

In this subsection, we first demonstrate the performance of our method on the isolated test set, showcasing the model’s results at each stage and directly comparing them with the ground truth when feasible.

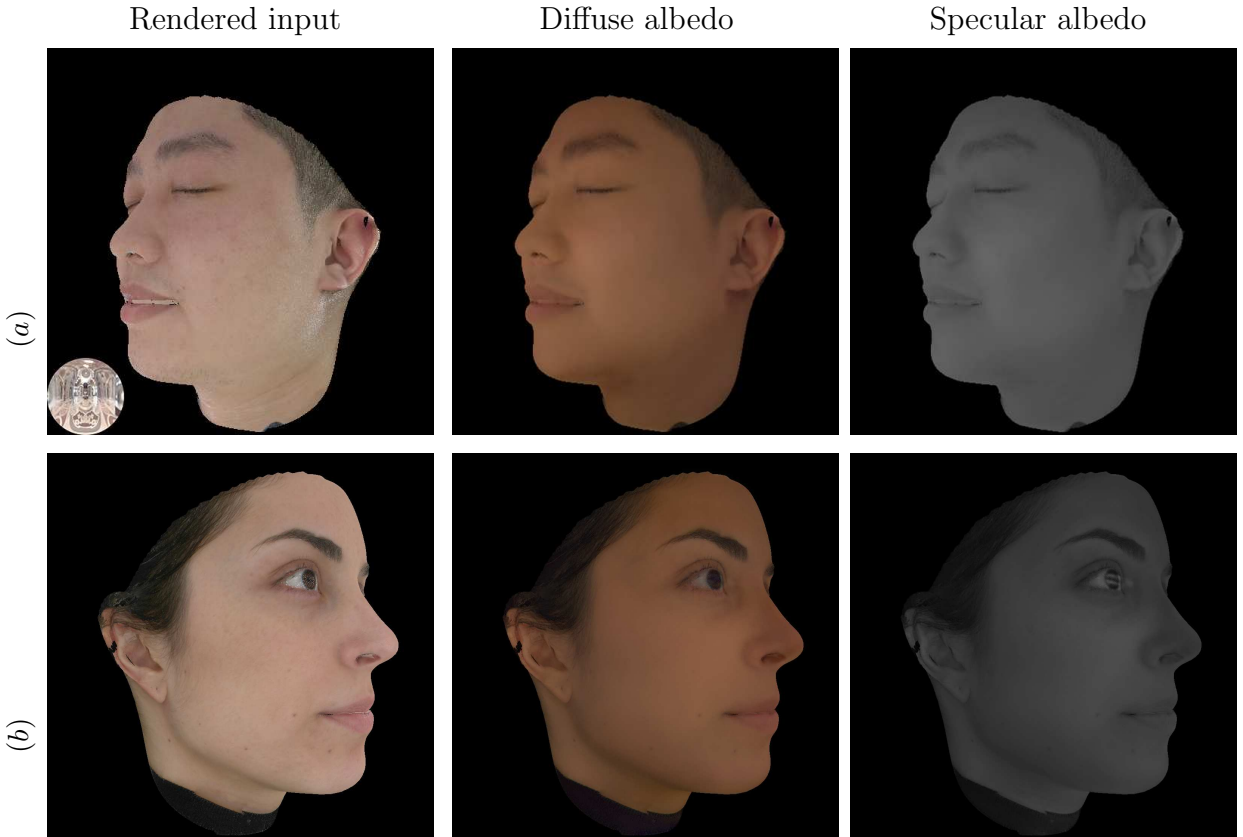


Figure 4.7: Estimated diffuse and specular albedo (right two columns) given a well lit face image (left most column). The input images have been rendered under the St. Peters environment (a), and uniform white illumination (b), respectively.

Figure 4.7 shows the estimated diffuse and specular albedo of two test subjects in image space. Our light-weight network in the first step efficiently estimates the diffuse and specular albedo given an image of the subject lit under a well-lit lighting environment. This estimation simplifies the task of the subsequent diffusion network and makes our method more robust to changes in lighting conditions. The generated texture maps and the position map that are conditioned by Figure 4.7 (a) are shown in Figure 4.8.

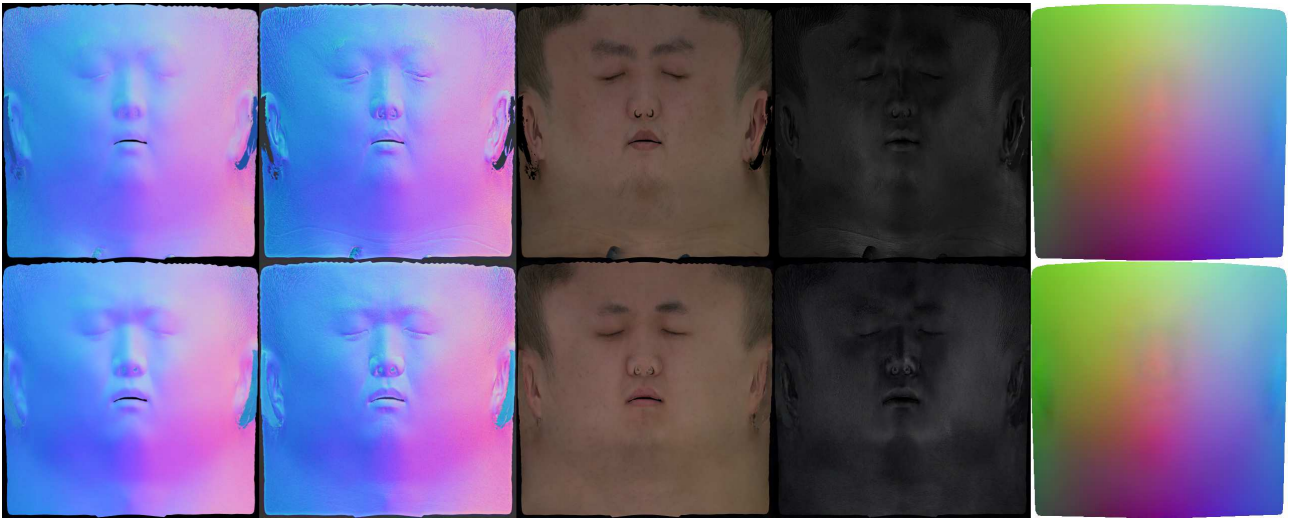


Figure 4.8: Comparison of the results generated by the network and ground truth. The first row shows the ground truth shape and SVBRDF maps, while the second row presents our generated results. From left to right: diffuse normal map, specular normal map, diffuse albedo map, specular albedo map, position map. Through visual inspection, it can be observed that our network generates texture maps that are qualitatively consistent with the ground truth maps.

Our network learns to generate a set of maps that closely match the ground truth maps and the corresponding diffuse and specular albedo images. Figure 4.9 (a) shows the reconstructed mesh based on the generated position map compared to the ground truth mesh. Figure 4.9 (b, c, d) compares the rendered results using the predicted SVBRDF maps and the ground truth maps.

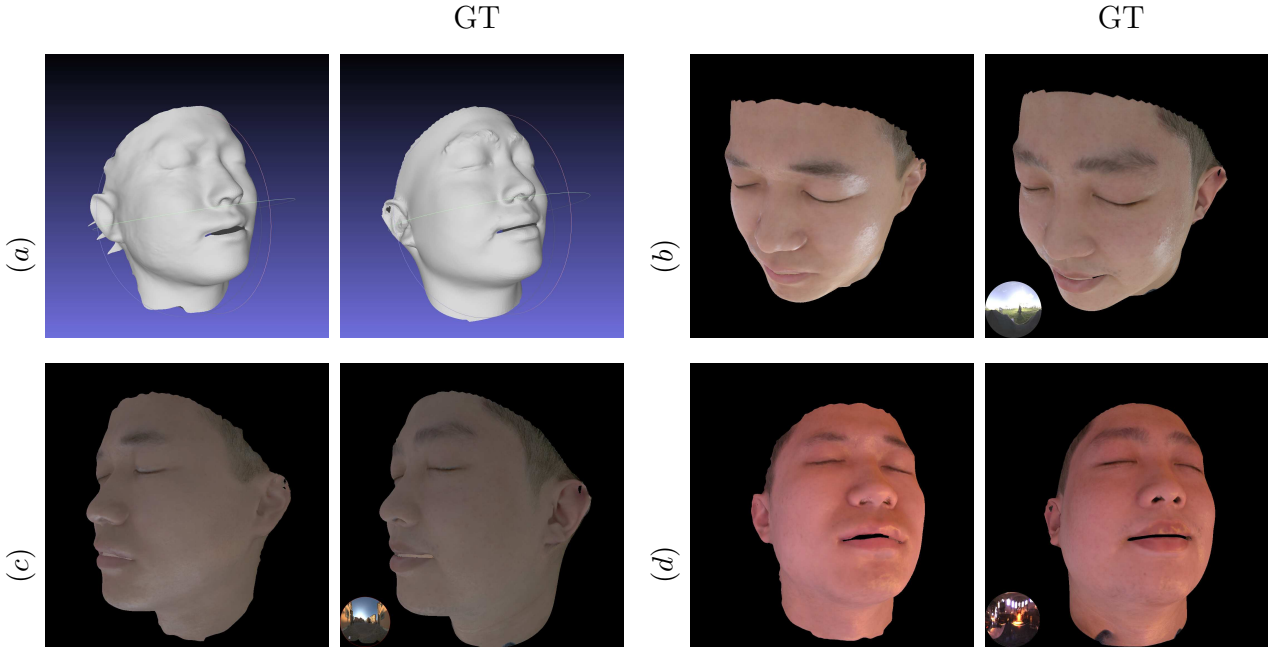


Figure 4.9: The reconstructed mesh (a) and the relighting results under multiple novel lighting conditions (b, c, d), compared to the ground truth (GT). The relighting results are illuminated under Park (b), Pisa (c), and Grace Cathedral (d).

Our method successfully reconstructs a 3D mesh that closely approximates the ground truth mesh and generates SVBRDF maps capable of producing high-quality rendering results. These results indicate that our method can effectively estimate the overall 3D appearance with just two input images while preserving key identity features.

4.3.2 Evaluation on Public Datasets and Field Experiment

In this subsection, we will display the results of our method on real photograph inputs. We will showcase the 3D meshes we obtained, the albedo results in UV space, and the relighting rendering results under novel lighting conditions.

We test our method’s performance on the ILSH [ZJP*23] dataset. The ILSH dataset provides multiple photographs of a subject lit under flat lighting from different angles. This allows us to use actual photographs as inputs instead of using rendered images. Figure 4.10 shows our estimated UV space SVBRDF maps and Figure 4.11 shows the reconstructed mesh, as well as various relight results using our approach.

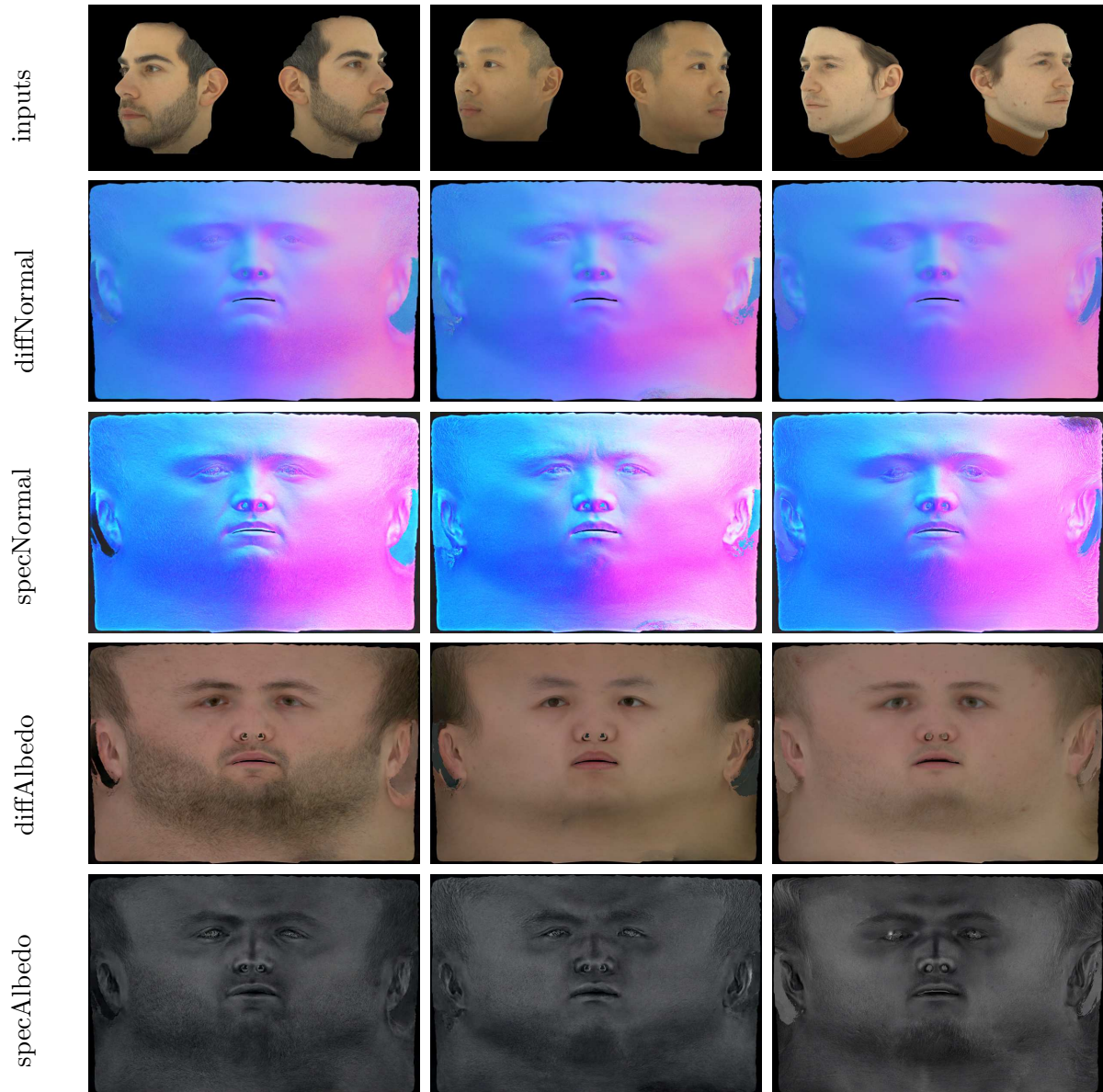


Figure 4.10: Generated results from two uniformly-lit photographs. The first row displays the model’s two input images. Rows 2 to 5 show the generated UV space SVBRDF maps. The remaining mesh and rendering results will be shown in the next figure, Fig 4.11

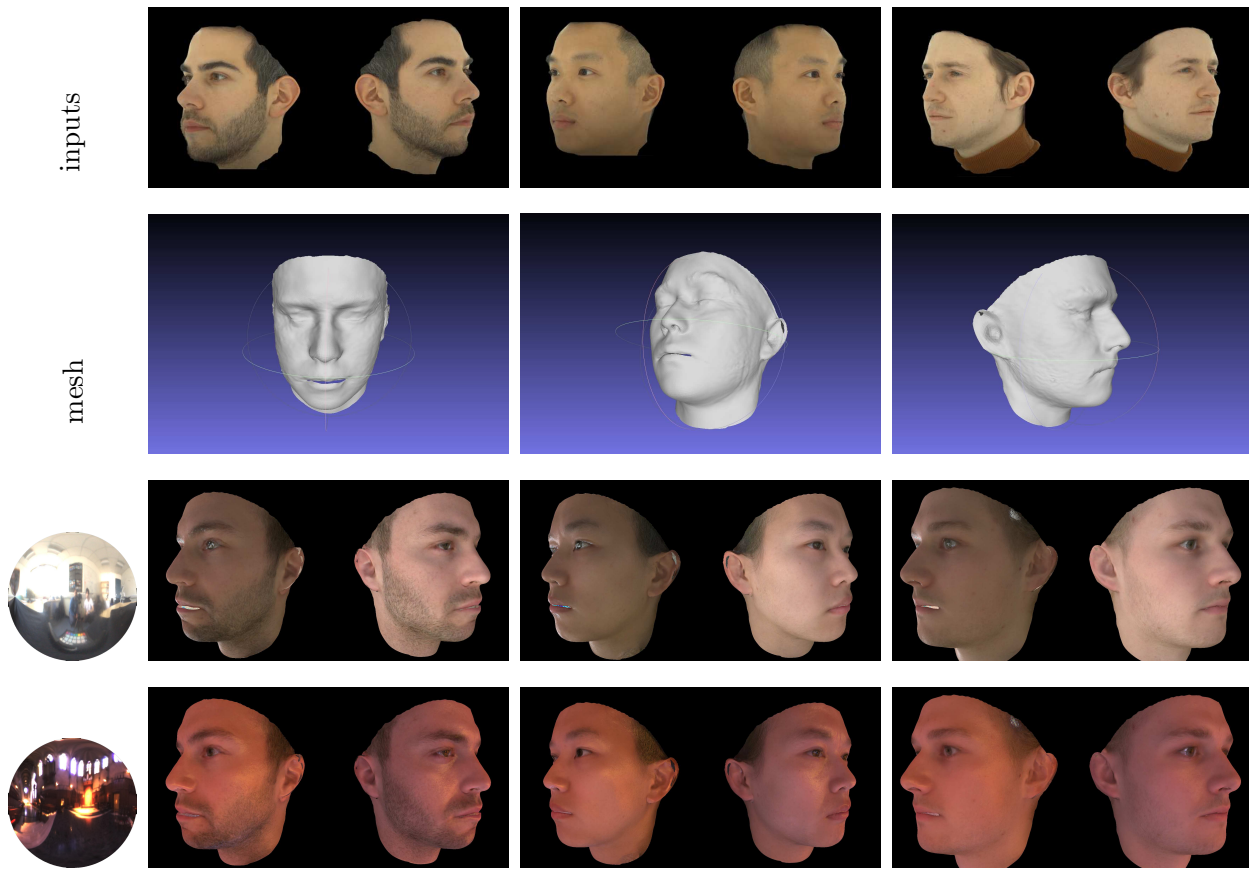


Figure 4.11: Rest results from two uniformly-lit photographs. The first row displays the model’s two input images. while row 2 is the reconstructed mesh. The mesh has been rotated to a certain angle to demonstrate the effectiveness of our 3D restoration. The last 2 rows display the relighting results based on the outputs above. The left side indicates the lighting conditions.

Additionally, we provide comparisons between our method and single-image neural methods in Figure 4.12. We chose to use HeadNeRF [HPX*22] as a representative example of these approaches as they exhibit similar advantages as well as drawbacks. Six unique observation angles are selected. As can be seen in 4.12, our method shows better identity preservation and is more robust in extreme angles. However, we acknowledge that our method requires two photographs instead of one, making it slightly more demanding in terms of input data for achieving high-quality reconstruction.

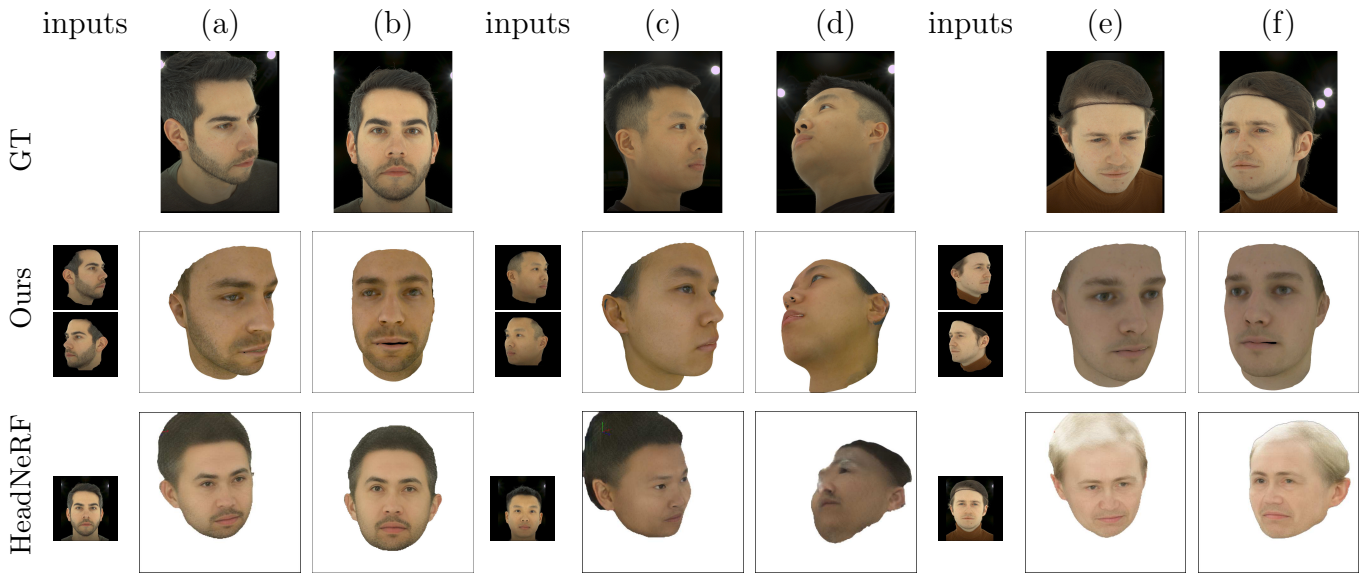


Figure 4.12: Comparison of rendered results between our method and headNeRF [HPX*22]. Top row shows the actual photographs that serve as ground truth. The second and third row shows our rendered results and results rendered by headNeRF respectively. Our method better preserves the identity and is more robust when rendering at extreme angles. While headNeRF requires only one input image, their work struggles to reconstruct details that are not present at the frontal image.

We compared our method with AvatarMe++ [LMP*21] using the Digital Emily dataset [Pro15], which provides the required left and right 45-degree side-profile photographs. AvatarMe++ generates diffuse albedo maps, specular albedo maps, and specular normal maps, enabling a direct comparison of the quality of the reflectance maps. As shown in Fig. 4.13, our method produces SVBRDF maps with higher quality and less noise, particularly on the side of the face.

The red squares highlight specific areas where our method captures smoother and more consistent details compared to AvatarMe++, which tends to introduce noticeable noise and artefacts in these regions. However, we acknowledge that AvatarMe++ may recover slightly more high-frequency details in flatter regions, such as the cheeks, which could contribute to sharper visual features in these areas. These differences reflect the trade-offs between the two methods, with ours prioritizing overall consistency and robustness in capturing facial details across different regions.

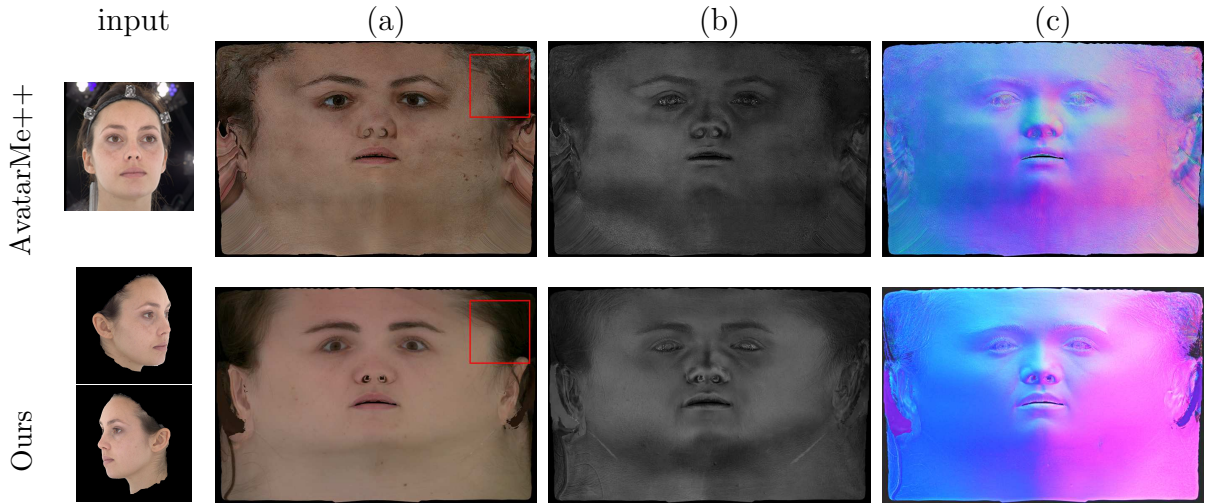


Figure 4.13: Comparison of our generated SVBRDF maps with AvatarMe++ [LMP*21]. We compare the diffuse albedo maps (a), specular albedo maps (b), and the specular normal maps (c). Our generated maps are less noisy and contains more details on the sides.

We also performed in-the-wild captures outside of the laboratory environment to further demonstrate the versatility and robustness of our method. The pipeline figure Fig. 4.1 shows an example of a face reconstruction where the two input images have been acquired in an uncontrolled indoor environment using a phone. Figure 4.14, 4.15 shows some additional results of our method under outdoor lighting conditions. We selected two typical outdoor scenes: an open space outside a building and a parking lot under cloudy weather.

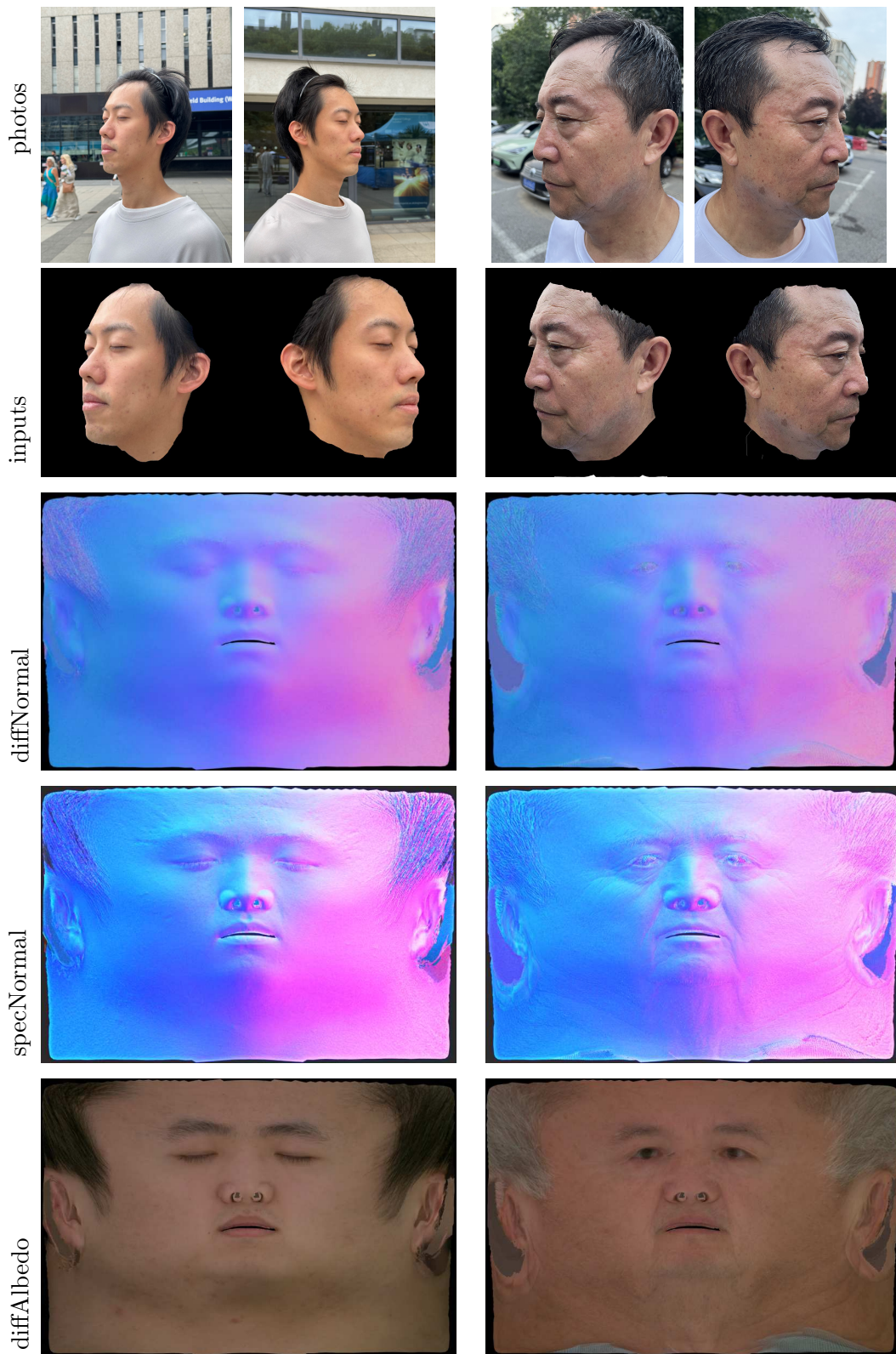


Figure 4.14: Additional results of subjects captured outdoor. The first row displays the raw images captured. The second row shows the model’s two input images. Rows 3 to 6 show the generated UV space SVBRDF maps. The last 2 rows display the relighting results based on the outputs above. The left side indicates the lighting conditions as same as Fig 4.10.

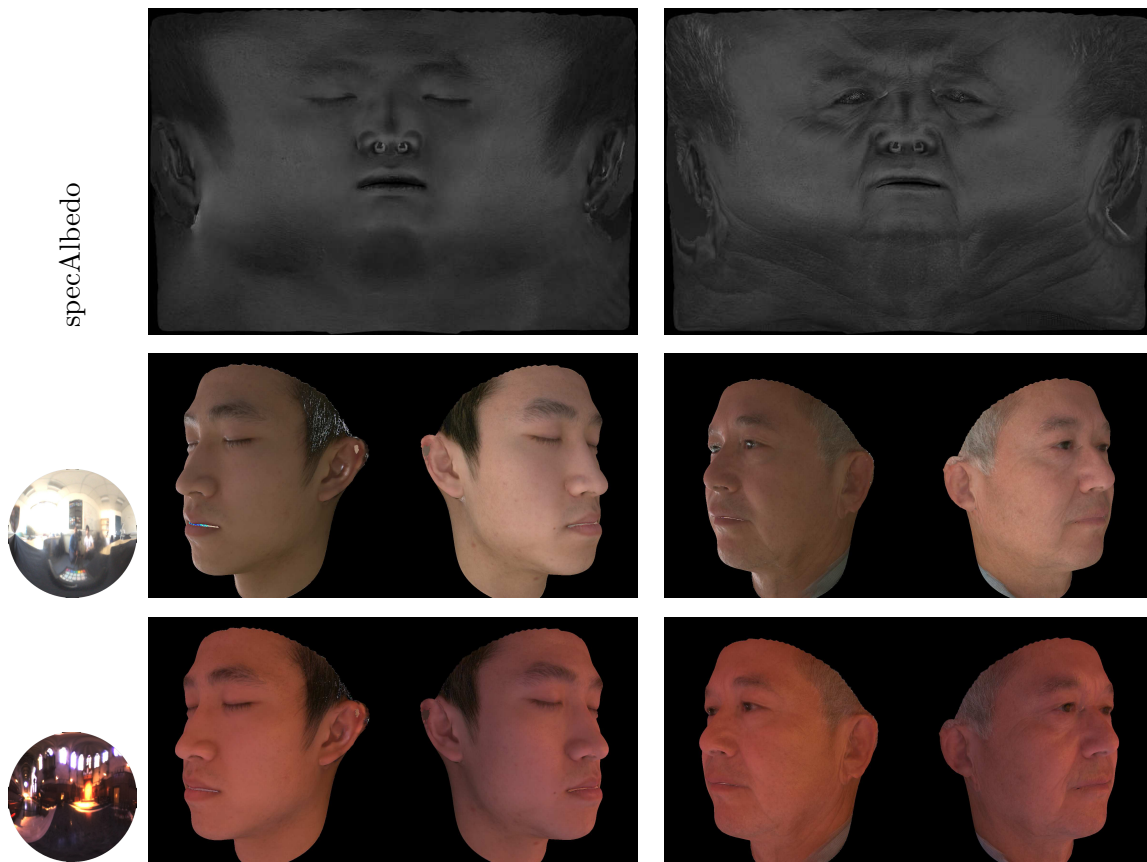


Figure 4.15: Additional results of subjects captured outdoor. The first row displays the raw images captured. The second row shows the model’s two input images. Rows 3 to 6 show the generated UV space SVBRDF maps. The last 2 rows display the relighting results based on the outputs above. The left side indicates the lighting conditions as same as Fig 4.10.

Figure 4.14 and 4.15 shows two side-face photos taken with a smartphone camera, along with network input, output, and relighting results. Our method is robust under lighting variations. The only requirement is the face being evenly illuminated to produce effective results.

4.3.3 Ablation Studies

In this subsection, we perform ablation studies on the number of input images, the importance of estimating the specular and diffuse albedo maps in image space before the diffusion network as well as fine-tuning design choices.

Number of Input Images

We trained our network using either one frontal image ('1 input'), two side images ('2 inputs'), or three images comprising the front and both sides ('3 inputs'). Qualitative results are shown in Figure 4.16, while Table 4.1 presents the root mean square error (RMSE) of each of the SVBRDF maps with different numbers of input images. These results were computed on the test dataset, which consists of 15 individuals from the RealFaceDB dataset, with six unique environment lighting conditions for each individual. This ensures a diverse evaluation covering different facial geometries and lighting environments.

We observe a significant improvement when using two input images compared to one. The difference in quality of the generated SVBRDF maps is particularly noticeable on the sides of the face, where a single frontal image cannot adequately capture the details. In terms of RMSE, using two input images yields a 30% improvement over using one. Conversely, while using three input images further reduces RMSE slightly, the improvement is less pronounced. Therefore, we opt for using two input images, which achieves comparable performance to three inputs while simplifying the capture process.

RMSE (%)	Dif.Nor.	Spe.Nor.	Dif.Alb.	Spe.Alb.	Mesh
1 input	3.634	3.858	3.941	0.0359	3.596
2 inputs	2.790	2.989	2.848	2.727	1.577
3 inputs	2.779	2.943	2.710	2.788	1.506

Table 4.1: Root mean square error (RMSE) of the SVBRDF maps with our model trained with 1, 2, and 3 input images. Significant improvement can be observed when using 2 images compared to 1 image, while the improvement with 3 images over 2 images is negligible.

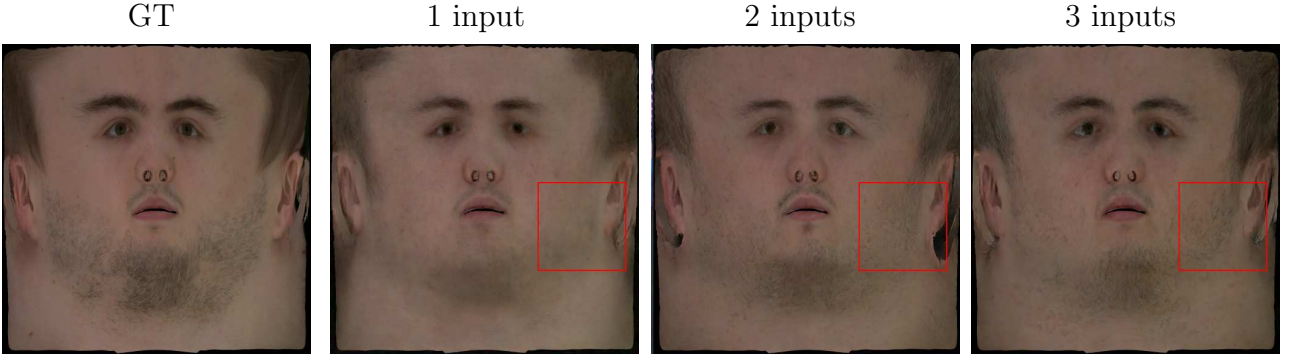


Figure 4.16: Comparison of albedo maps obtained using different number of input images. The map generated map using only the frontal image as input fails to accurately reconstruct the beard on the side cheek due to the lack of information in frontal only image.

Estimation net

In this subsection, we discuss the importance of estimating the specular and diffuse albedo maps in image space before the diffusion network. As previously mentioned, this step offloads the heavy processing from the diffusion network, allowing it to focus more effectively on a single task. Figure 4.17 shows the results of using our current method compared to not using the first network and directly predicting in UV maps. Introducing this pre-diffusion step results in reduced residual specular, a more accurate and consistent representation of the diffuse albedo across the entire surface, less residual shading, and better-preserved details.



Figure 4.17: Comparison of albedo maps obtained with diffuse an specular estimation network or directly predict the UV space maps. In terms of details, the result on the far right exhibits less noise and better consistency compared to the image in the middle.

Additionally, the estimation network generally enhances the overall robustness of our model to

various lighting conditions. Figure 4.18 shows one example that compares the results obtained under the same person’s face rendered under two different lighting conditions as inputs, one of them under Grace Cathedral illumination which has some colour.

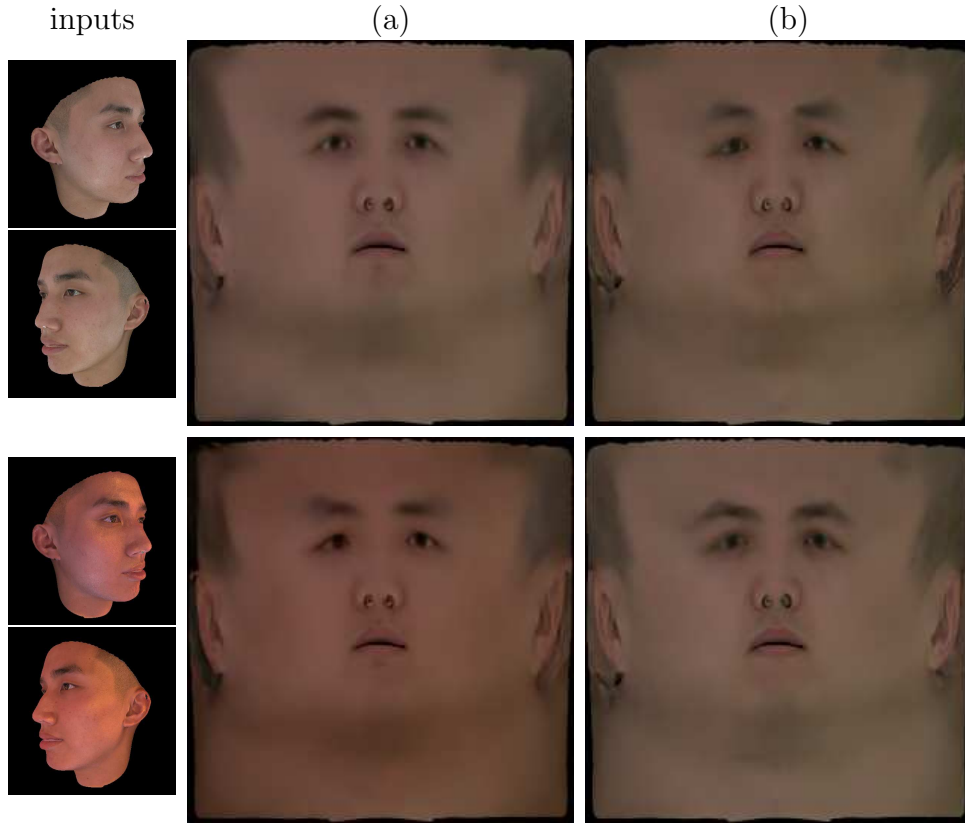


Figure 4.18: Comparison of the generated diffuse albedo map with and without first estimating the diffuse and specular maps in image space. (a): generated diffuse using only the captured photographs. (b): generated diffuse using the captured photographs and the estimated diffuse and specular maps in image space. It can be observed that by first estimating the diffuse and specular albedo, the generated diffuse albedo map is more robust to lighting variations in the captured photographs.

It can be observed when the same rendering inputs are provided to both methods, the method without diffuse and specular albedo estimation in image space exhibits significant colour bias under the Grace Cathedral lighting condition. The right-side result with the estimation net, however, demonstrates robustness to lighting conditions.

RMSE (%)	Dif.Nor.	Spe.Nor.	Dif.Alb.	Spe.Alb.
w.o. Esti.net	2.875	3.235	3.494	2.753
w. Esti.net	2.790	2.989	2.848	2.727

Table 4.2: Root mean square error of the generated SVBRDF maps with and without render loss. The introduction of render loss improves all of the generated SVBRDF maps.

Table 4.2 calculates the root mean squared error on SVBRDF with and without including the first network in the test dataset. It is shown that pre-diffusion processing of the input images in image space is beneficial for the overall results, particularly the improvement in the diffuse albedo map. Additionally, this processing can enhance the model’s tolerance to different lighting conditions, thereby reducing the difficulty of capture.

Fine-Tuning

In this subsection, we experiment with the effectiveness of our fine-tuning methods. We investigate both the effects on geometry reconstruction as well as the quality of the generated SVBRDF maps.

SVBRDF

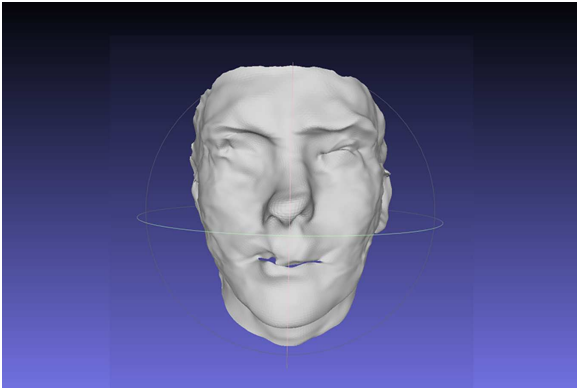
To evaluate the effectiveness of fine-tuning SVBRDF maps with our proposed render loss, we compare the RMSE of the generated maps both with and without the fine-tuning step with render loss. The results are shown in Table 4.3. As we can see, the introduction of render loss further improves the quality of the estimated normal and albedo maps, which improves the realism of the rendered results.

RMSE (%)	Dif.Nor.	Spe.Nor.	Dif.Alb.	Spe.Alb.
w.o. Render Loss	2.989	3.215	3.282	2.899
w. Render Loss	2.790	2.989	2.848	2.727

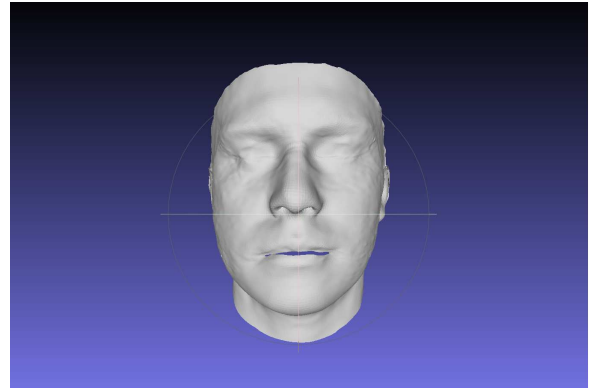
Table 4.3: Root mean square error of the generated SVBRDF maps with and without render loss. The introduction of render loss improves all of the generated SVBRDF maps.

Geometry

Figure 4.19 compares the reconstructed geometry with and without fine-tuning. After fine-tuning with the landmark loss and smoothness loss, the resulting mesh much better represents the geometry of a human face. Introducing the fine-tuning step allows the network to learn the 3D structural information in a 2D position map with the help of the detected positional landmarks. The resulting 2D position map is more spatially consistent which greatly improves the quality of the reconstructed mesh.



(a) Without fine-tuning



(b) With fine-tuning

Figure 4.19: Reconstructed mesh with position maps obtained before fine-tuning (a) and after the fine-tuning stage (b). The improvement is obtained with the introduction of landmark loss and smoothness loss in the fine-tuning stage.

4.4 Limitations

Our results are somewhat constrained by the size and diversity of our training data, as well as the resolution of the output textures, which is limited by GPU memory. The training dataset does not fully encompass a wide range of skin types and facial skeletal structures. This has led our model to exhibit biases towards specific demographic groups, such as more light-skinned subjects, with fewer dark-skinned subjects represented. Similarly, the RealFaceDB dataset [LMG*20] contains more male subjects than females, resulting in slightly poorer performance for females compared to males. The scanned dataset used for creating the training data is relatively small for a generative model, and this limited diversity may lead to slight changes in facial identity during reconstruction. Our method is also restricted to reconstructing the face area only, as the training data does not include hair or the back of the head. Future improvements could include augmenting the dataset with a more balanced and diverse representation, as well as exploring techniques like image super-resolution to enhance the resolution of the SVBRDF maps.

Additionally, the test dataset used in our evaluation consists of only 15 subjects, which limits the statistical significance of our results. While we ensured diversity within the available data, this small scale may not fully reflect the broader applicability of our method.

Finally, ethical considerations play a crucial role in the development and application of facial reconstruction technologies. One of the primary concerns lies in the potential psychological effects on individuals who might misinterpret the reconstructed model as an exact representation of their appearance. This is particularly relevant for individuals with body dysmorphia or heightened sensitivity to perceived flaws. For example, subtle deviations in surface details, texture consistency, or geometry might be interpreted as flaws in their actual appearance rather than limitations of the algorithm. To mitigate these risks, it is essential to ensure that end users—particularly non-experts—are fully informed about the limitations of the reconstruction process. This includes explicitly communicating that the reconstructed model is an approximation and may not perfectly represent the subject’s true appearance. Providing visual examples of expected accuracy ranges will help set realistic expectations for users.

4.5 Conclusion

In this work, we present a two-stage pipeline that generates a 3D relightable facial model of a subject using two images capturing the left and right sides of the face. The pipeline includes a pre-diffusion step for estimating albedo maps and a refinement stage that leverages three fine-tuning loss functions to improve the quality of the generated shape and SVBRDF maps. Our method produces results that are compatible with standard rendering engines, enabling fast 3D rendering under any given environment map.

We demonstrate that our method outperforms current state-of-the-art NeRF-based head reconstruction methods in terms of identity preservation and rendering quality at extreme angles. Our ablation studies show that using two input images provides an optimal balance between reconstruction quality and the ease of data capture, making the method suitable for practical applications.

While our method is effective, we acknowledge its limitations, including biases in the training and test datasets, the absence of hair and non-facial regions in reconstructions, and the need for high-quality input data. These aspects highlight areas for future work, such as expanding the dataset diversity, improving resolution, and addressing non-facial regions.

Importantly, we recognize the ethical implications of facial reconstruction technologies. Inaccuracies in the reconstructed model could lead to potential psychological effects, particularly for individuals with heightened sensitivity to perceived distortions. To mitigate these risks, future developments should focus on transparency, such as informing end users about the approximate nature of the reconstruction and providing guidelines for interpretation. By addressing these considerations, we aim to ensure that our method is not only technically robust but also socially responsible.

Chapter 5

Conclusion

This thesis focuses on a new method that utilises everyday capture devices to acquire spatially varying bidirectional reflectance distribution functions and geometry through a simplified data acquisition process. The primary goal of this research is to develop a practical, high-quality 3D reconstruction technique that is user-friendly and applicable to real-world scenarios.

To achieve this objective, the research addresses the following key questions. First, we verified that a lightweight approach using only two input images can effectively reconstruct high-quality 3D shapes and SVBRDF maps. By leveraging the multi-lens imaging capabilities of smart-phones, which capture two photos with different focal lengths and slight viewpoint variations, our method significantly outperforms existing single-shot approaches without increasing the operational cost for users. This method demonstrates robust performance under various lighting conditions, including flash illumination, indoor lighting, and outdoor environments, making it suitable for non-laboratory settings.

For a more specific and widely applicable 3D target—human faces—we further explored methods to acquire SVBRDF and geometry with minimal input. The proposed deep learning-based approach requires only two 45-degree side-profile photographs as input. Unlike traditional methods that rely on cumbersome pre-trained deformable models, our method employs a two-stage network structure to decompose the task, enabling end-to-end processing from input images to complete facial SVBRDF UV maps. This design enhances usability. By introducing

fine-tuned loss functions, we further improved detail preservation and reduced artefacts.

Additionally, throughout the study, we ensured that the generated maps meet the requirements for rendering and relighting tasks. The outputs include detailed SVBRDF maps and shape representations, significantly simplifying subsequent use in rendering engines. Both approach provides a practical tool for individual artists to extract 3D appearances in non-laboratory environments and pursue creative projects, while its compatibility with standard rendering engines makes it suitable for professional creators and a broader user base.

Nevertheless, it is important to acknowledge that the lightweight method we developed has limitations in its scope of application. Our multi-lens 3D object capture method is limited to objects of a certain size. To enhance results, we introduced a scaling adjustment step, which somewhat reduces the simplicity of the method. Notably, as the method is based on deep learning, it presents an inevitable challenge—dependence on the training dataset. Neural networks are neither magicians nor alchemists; regardless of how well-designed the architecture is, poor-quality or unrepresentative training data will result in suboptimal outcomes. As highlighted, the advantage of our method lies in generating all detailed SVBRDF maps that directly drive rendering. This requires the network to have ground truth data for these maps during training. Clearly, constructing such a complex real-world dataset with detailed SVBRDF maps for everyday 3D objects and human faces is highly challenging. For training our method, we generated synthetic datasets based on available SVBRDF maps. However, despite our best efforts, synthetic data may not fully reflect real-world conditions. For human faces, the training dataset is even more critical. It must be diverse and representative in terms of skin tones, genders, and other factors, while also addressing ethical concerns. These are significant issues in the development and application of facial reconstruction technology. The lack of training data results in limitations such as restricted reconstruction regions and reduced generalisation capabilities in real-world environments. For facial targets, the final reconstructed model may deviate from the individual's true appearance, potentially causing psychological distress for users sensitive to their appearance (e.g., users with body image disorders). At present, acquiring large, representative training datasets for different targets remains a key challenge in using neural networks to obtain SVBRDF, and it is likely to remain a critical issue for the foreseeable

future.

In summary, there is still a long way to go in the research on reflectance and shape acquisition. It is worth noting the progress made by a series of self-supervised learning methods in the vision and graphics community. For instance, Neural Radiance Fields (NeRF) use self-supervised methods to learn 3D representations of scenes from different 2D viewpoints, although it is primarily used for generating volumetric 3D rendering. NeRF constructs high-quality 3D representations using a set of unannotated 2D images through self-supervised learning. Although NeRF-based 3D reconstruction does not explicitly model lighting or material properties and is difficult to integrate with traditional rendering engines, developing improved self-supervised network architectures for everyday objects and human faces, rather than relying on synthetic data for training, likely represents a promising path forward. This would eliminate the need for known ground truth, allowing neural networks to be trained with real data, thereby removing artificial artefacts or unrealistic perfection from synthetic data.

Bibliography

- [AAL16] AITTALA M., AILA T., LEHTINEN J.: Reflectance modeling by neural texture synthesis. *ACM Trans. Graph.* 35, 4 (July 2016). doi:10.1145/2897824.2925917.
- [AH04] ATKINSON G., HANCOCK E.: Shape from diffuse polarisation. In *Proceedings of the British Machine Vision Conference 2004* (2004), BMVC 2004, British Machine Vision Association, pp. 94.1–94.10. doi:10.5244/c.18.94.
- [AWL15] AITTALA M., WEYRICH T., LEHTINEN J.: Two-shot svbrdf capture for stationary materials. *ACM Trans. Graph.* 34, 4 (July 2015). doi:10.1145/2766967.
- [BBB*10] BEELER T., BICKEL B., BEARDSLEY P., SUMNER B., GROSS M.: High-quality single-shot capture of facial geometry. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 1–9. doi:10.1145/1778765.1778777.
- [BEWW*08] BEN-EZRA M., WANG J., WILBURN B., LI X., MA L.: An led-only brdf measurement device. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (2008), pp. 1–8. doi:10.1109/CVPR.2008.4587766.
- [BJK*20] BOSS M., JAMPANI V., KIM K., LENSCH H. P., KAUTZ J.: Two-shot spatially-varying brdf and shape estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020), pp. 3982–3991. doi:10.1109/CVPR42600.2020.00404.

- [BJTK18] BAEK S.-H., JEON D. S., TONG X., KIM M. H.: Simultaneous acquisition of polarimetric svbrdf and normals. *ACM Trans. Graph.* 37, 6 (Dec. 2018). doi:10.1145/3272127.3275018.
- [Bli77] BLINN J. F.: Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph.* 11, 2 (July 1977), 192–198. doi:10.1145/965141.563893.
- [BLS*21] BI S., LOMBARDI S., SAITO S., SIMON T., WEI S.-E., MCPHAIL K., RAMAMOORTHY R., SHEIKH Y., SARAGIH J.: Deep relightable appearance models for animatable faces. *ACM Transactions on Graphics (ToG)* 40, 4 (2021), 1–15. doi:10.1145/3450626.3459829.
- [BM15] BARRON J. T., MALIK J.: Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 8 (2015), 1670–1687. doi:10.1109/TPAMI.2014.2377712.
- [BRZ*16] BOOTH J., ROUSSOS A., ZAFEIRIOU S., PONNIAH A., DUNAWAY D.: A 3d morphable model learnt from 10,000 faces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), IEEE, p. 5543–5552. doi:10.1109/cvpr.2016.598.
- [BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99. ACM Press/Addison-Wesley Publishing Co., USA, 1999, p. 187–194. doi:10.1145/311535.311556.
- [BWSB12] BUTLER D. J., WULFF J., STANLEY G. B., BLACK M. J.: A naturalistic open source movie for optical flow evaluation. In *Computer Vision – ECCV 2012* (2012), Springer Berlin Heidelberg, p. 611–625. doi:10.1007/978-3-642-33783-3_44.
- [CBZ*22] CHANG D., BOŽIČ A., ZHANG T., YAN Q., CHEN Y., SÜSSTRUNK S., NIESSNER M.: Rc-mvsnet: Unsupervised multi-view stereo with neural ren-

- dering. In *Computer Vision – ECCV 2022* (2022), Springer Nature Switzerland, p. 665–680. doi:10.1007/978-3-031-19821-2_38.
- [CDP*14] CHEN G., DONG Y., PEERS P., ZHANG J., TONG X.: Reflectance scanning: estimating shading frame and brdf with generalized linear light sources. *ACM Trans. Graph.* 33, 4 (July 2014). doi:10.1145/2601097.2601180.
- [CGC*23] CHEN H., GU J., CHEN A., TIAN W., TU Z., LIU L., SU H.: Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (Oct. 2023), IEEE. doi:10.1109/iccv51070.2023.00229.
- [CT82] COOK R. L., TORRANCE K. E.: A reflectance model for computer graphics. *ACM Trans. Graph.* 1, 1 (Jan. 1982), 7–24. doi:10.1145/357290.357293.
- [CZGB23] CHANDRAN P., ZOSS G., GOTARDO P., BRADLEY D.: Continuous landmark detection with 3d queries. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2023), IEEE, p. 16858–16867. doi:10.1109/cvpr52729.2023.01617.
- [DAD*18] DESCHAI NTRE V., AITTALA M., DURAND F., DRETTAKIS G., BOUSSEAU A.: Single-image svbrdf capture with a rendering-aware deep network. *ACM Trans. Graph.* 37, 4 (July 2018). doi:10.1145/3197517.3201378.
- [DAD*19] DESCHAI NTRE V., AITTALA M., DURAND F., DRETTAKIS G., BOUSSEAU A.: Flexible svbrdf capture with a multi-image deep network. *Computer Graphics Forum* 38, 4 (July 2019), 1–13. doi:10.1111/cgf.13765.
- [DFI*15] DOSOVITSKIY A., FISCHER P., ILG E., HAUSSE R P., HAZIRBAS C., GOLKOV V., SMAGT P. V. D., CREMERS D., BROX T.: Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)* (Dec. 2015), IEEE, p. 2758–2766. doi:10.1109/iccv.2015.316.

- [DLG21] DESCHAINTE V., LIN Y., GHOSH A.: Deep polarization imaging for 3d shape and svbrdf acquisition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021), IEEE, pp. 15567–15576. doi:10.1109/CVPR46437.2021.01532.
- [DN21] DHARIWAL P., NICHOL A.: Diffusion models beat gans on image synthesis. *Advances in neural information processing systems 34* (2021), 8780–8794. doi:10.48550/arXiv.2105.05233.
- [FLG23] FAN C., LIN Y., GHOSH A.: Deep shape and svbrdf estimation using smartphone multi-lens imaging. *Computer Graphics Forum 42*, 7 (Oct. 2023). doi:10.1111/cgf.14972.
- [Foo97] FOO S. C.: *A Gonioreflectometer for Measuring the Bidirectional Reflectance of Material for Use in Illumination Computation*. Phd thesis, Cornell University, 1997.
- [GCP*09] GHOSH A., CHEN T., PEERS P., WILSON C. A., DEBEVEC P. E.: Estimating specular roughness and anisotropy from second order spherical gradient illumination. *Computer Graphics Forum 28*, 4 (2009), 1161–1170. doi:10.1111/j.1467-8659.2009.01493.x.
- [GCP*10] GHOSH A., CHEN T., PEERS P., WILSON C. A., DEBEVEC P.: Circularly polarized spherical illumination reflectometry. In *ACM SIGGRAPH Asia 2010 Papers* (New York, NY, USA, 2010), SIGGRAPH ASIA '10, Association for Computing Machinery. doi:10.1145/1866158.1866163.
- [GFT*11] GHOSH A., FYFFE G., TUNWATTANAPONG B., BUSCH J., YU X., DEBEVEC P.: Multiview face capture using polarized spherical gradient illumination. In *Proceedings of the 2011 SIGGRAPH Asia Conference* (New York, NY, USA, 2011), SA '11, Association for Computing Machinery. doi:10.1145/2024156.2024163.

- [GHAO10] GHOSH A., HEIDRICH W., ACHUTHA S., O'TOOLE M.: A basis illumination approach to brdf measurement. *International Journal of Computer Vision* 90, 2 (2010), 183–197. doi:10.1007/s11263-008-0151-7.
- [GLD*19] GAO D., LI X., DONG Y., PEERS P., XU K., TONG X.: Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images. *ACM Trans. Graph.* 38, 4 (July 2019). doi:10.1145/3306346.3323042.
- [GLU12] GEIGER A., LENZ P., URTASUN R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (June 2012), IEEE, p. 3354–3361. doi:10.1109/cvpr.2012.6248074.
- [GPAM*14] GOODFELLOW I., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. *Advances in neural information processing systems* 27 (2014). doi:10.1007/978-3-658-40442-0_9.
- [GPDG12] GUARNERA G. C., PEERS P., DEBEVEC P., GHOSH A.: Estimating surface normals from spherical stokes reflectance fields. In *Computer Vision – ECCV 2012. Workshops and Demonstrations: Florence, Italy, October 7-13, 2012, Proceedings, Part II* (2012), Springer Berlin Heidelberg, pp. 340–349. doi:10.1007/978-3-642-33868-7_34.
- [GRB*18] GOTARDO P., RIVIERE J., BRADLEY D., GHOSH A., BEELER T.: Practical dynamic facial appearance modeling and acquisition. *ACM Transactions on Graphics* 37, 6 (Dec. 2018), 1–13. doi:10.1145/3272127.3275073.
- [GTHD03] GARDNER A., TCHOU C., HAWKINS T., DEBEVEC P.: Linear light source reflectometry. *ACM Trans. Graph.* 22, 3 (July 2003), 749–758. doi:10.1145/882262.882342.

- [GY*19] GUO X., YANG K., YANG W., WANG X., LI H.: Group-wise correlation stereo network. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019), IEEE. doi:10.1109/cvpr.2019.00339.
- [HBC23] HEITZ E., BELCOUR L., CHAMBON T.: Iterative α -(de) blending: A minimalist deterministic diffusion model. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings* (July 2023), ACM, p. 1–8. doi:10.1145/3588432.3591540.
- [HBNK20] HA H., BAEK S., NAM G., KIM M. H.: Progressive acquisition of svbrdf and shape in motion. *Computer Graphics Forum* 39, 6 (Aug. 2020), 480–495. doi:10.1111/cgf.14087.
- [HJA20] HO J., JAIN A., ABBEEL P.: Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems* (2020), Larochelle H., Ranzato M., Hadsell R., Balcan M., Lin H., (Eds.), vol. 33, Curran Associates, Inc., pp. 6840–6851. doi:10.48550/ARXIV.2006.11239.
- [HLVDMW17] HUANG G., LIU Z., VAN DER MAATEN L., WEINBERGER K. Q.: Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 2261–2269. doi:10.1109/CVPR.2017.243.
- [HPX*22] HONG Y., PENG B., XIAO H., LIU L., ZHANG J.: Headnerf: A realtime nerf-based parametric head model. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022), IEEE, p. 20342–20352. doi:10.1109/cvpr52688.2022.01973.
- [HS81] HORN B. K., SCHUNCK B. G.: Determining optical flow. *Artificial Intelligence* 17, 1 (1981), 185–203. doi:10.1016/0004-3702(81)90024-2.
- [HSL*17] HUI Z., SUNKAVALLI K., LEE J.-Y., HADAP S., WANG J., SANKARANARAYANAN A. C.: Reflectance capture using univariate sampling of brdfs.

- In *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 5372–5380. doi:10.1109/ICCV.2017.573.
- [HSV*24] HUANG X., SALAUN C., VASCONCELOS C., THEOBALT C., OZTIRELI C., SINGH G.: Blue noise for diffusion models. In *ACM SIGGRAPH 2024 Conference Papers* (New York, NY, USA, 2024), SIGGRAPH '24, Association for Computing Machinery. doi:10.1145/3641519.3657435.
- [HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778. doi:10.1109/CVPR.2016.90.
- [IMS*17] ILG E., MAYER N., SAIKIA T., KEUPER M., DOSOVITSKIY A., BROX T.: FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), IEEE. doi:10.1109/cvpr.2017.179.
- [Kaj86] KAJIYA J. T.: The rendering equation. *SIGGRAPH Comput. Graph.* 20, 4 (Aug. 1986), 143–150. doi:10.1145/15886.15902.
- [KFR*18] KHAMIS S., FANELLO S., RHEMANN C., KOWDLE A., VALENTIN J., IZADI S.: Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Computer Vision – ECCV 2018* (2018), Springer International Publishing, p. 596–613. doi:10.1007/978-3-030-01267-0_35.
- [Kin13] KINGMA D. P.: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
- [KRFB06] KHAN E. A., REINHARD E., FLEMING R. W., BÜLTHOFF H. H.: Image-based material editing. *ACM Trans. Graph.* 25, 3 (July 2006), 654–663. doi:10.1145/1141911.1141937.
- [KSH17] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. vol. 60, Association for Computing Machinery (ACM), p. 84–90. doi:10.1145/3065386.

- [KTSR15] KADAMBI A., TAAMAZYAN V., SHI B., RASKAR R.: Polarized 3d: High-quality depth sensing with polarization cues. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2015), IEEE, pp. 3370–3378. doi:10.1109/ICCV.2015.386.
- [KZG18] KAMPOURIS C., ZAFEIRIOU S., GHOSH A.: Diffuse-Specular Separation using Binary Spherical Gradient Illumination. In *Eurographics Symposium on Rendering - Experimental Ideas and Implementations* (2018), Jakob W., Hachisuka T., (Eds.), The Eurographics Association. doi:10.2312/sre.20181167.
- [Lam92] LAMBERT J. H.: *Photometrie: (Photometria, sive De Mensura et Gradibus Luminis, Colorum et Umbrae)*. No. 31–33. W. Engelmann, 1892. Originally published in 1760.
- [LBBH98] LECUN Y., BOTTOU L., BENGIO Y., HAFFNER P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324. doi:10.1109/5.726791.
- [LCC*22] LIU S., CAI Y., CHEN H., ZHOU Y., ZHAO Y.: Rapid face asset acquisition with recurrent feature alignment. *ACM Transactions on Graphics* 41, 6 (Nov. 2022), 1–17. doi:10.1145/3550454.3555509.
- [LDPT17] LI X., DONG Y., PEERS P., TONG X.: Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Transactions on Graphics* 36, 4 (July 2017), 1–11. doi:10.1145/3072959.3073641.
- [LK81] LUCAS B. D., KANADE T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI’81: 7th international joint conference on Artificial intelligence* (Vancouver, Canada, Aug. 1981), vol. 2, pp. 674–679.
- [LLK*22] LATTAS A., LIN Y., KANNAN J., OZTURK E., FILIPI L., GUARNERA G. C., CHAWLA G., GHOSH A.: Practical and scalable desktop-based high-quality

- facial capture. In *Computer Vision – ECCV 2022* (2022), Springer Nature Switzerland, p. 522–537. doi:10.1007/978-3-031-20068-7_30.
- [LLLG24] LIN A., LIN Y., LI X., GHOSH A.: High Quality Neural Relighting using Practical Zonal Illumination. In *Eurographics Symposium on Rendering* (2024), Haines E., Garces E., (Eds.), The Eurographics Association. doi:10.2312/sr.20241150.
- [LMG*20] LATTAS A., MOSCHOLOU S., GECER B., PLOUMPIS S., TRIANTAFYLLOU V., GHOSH A., ZAFEIRIOU S.: Avatarme: Realistically renderable 3d facial reconstruction “in-the-wild”. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020), IEEE, p. 757–766. doi:10.1109/cvpr42600.2020.00084.
- [LMP*21] LATTAS A., MOSCHOLOU S., PLOUMPIS S., GECER B., GHOSH A., ZAFEIRIOU S.: Avatarme++: Facial shape and brdf inference with photorealistic rendering-aware gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 12 (2021), 9269–9284. doi:10.1109/TPAMI.2021.3052684.
- [LMP*23] LATTAS A., MOSCHOLOU S., PLOUMPIS S., GECER B., DENG J., ZAFEIRIOU S.: Fitme: Deep photorealistic 3d morphable model avatars. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2023), IEEE, p. 8629–8640. doi:10.1109/cvpr52729.2023.00834.
- [LSC18] LI Z., SUNKAVALLI K., CHANDRAKER M.: Materials for masses: Svbrdf acquisition with a single mobile phone image. In *Computer Vision – ECCV 2018* (2018), Springer International Publishing, p. 74–90. doi:10.1007/978-3-030-01219-9_5.
- [LTD21] LIPSON L., TEED Z., DENG J.: Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *2021 International Conference on 3D Vision (3DV)* (Dec. 2021), IEEE, p. 218–227. doi:10.1109/3dv53792.2021.00032.

- [LWX*22] LI J., WANG P., XIONG P., CAI T., YAN Z., YANG L., LIU J., FAN H., LIU S.: Practical stereo matching via cascaded recurrent network with adaptive correlation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022), IEEE. doi:10.1109/cvpr52688.2022.01578.
- [LWZG19] LATTAS A., WANG M., ZAFEIRIOU S., GHOSH A.: Multi-view facial capture using binary spherical gradient illumination. In *ACM SIGGRAPH 2019 Posters* (July 2019), ACM, p. 1–2. doi:10.1145/3306214.3338611.
- [LXR*18] LI Z., XU Z., RAMAMOORTHY R., SUNKAVALLI K., CHANDRAKER M.: Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics* 37, 6 (Dec. 2018), 1–11. doi:10.1145/3272127.3275055.
- [MHG15] MENZE M., HEIPKE C., GEIGER A.: Joint 3d estimation of vehicles and scene flow. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-3/W5* (Aug. 2015), 427–434. doi:10.5194/isprsannals-ii-3-w5-427-2015.
- [MHP*07] MA W.-C., HAWKINS T., PEERS P., CHABERT C.-F., WEISS M., DEBEVEC P.: Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques* (Goslar, DEU, 2007), EGSR’07, Eurographics Association, p. 183–194. doi:10.2312/EGWR/EGSR07/183-194.
- [MHP*19] MEKA A., HÄNE C., PANDEY R., ZOLLHÖFER M., FANELLO S., FYFFE G., KOWDLE A., YU X., BUSCH J., DOURGARIAN J., DENNY P., BOUAZIZ S., LINCOLN P., WHALEN M., HARVEY G., TAYLOR J., IZADI S., TAGLIASACCHI A., DEBEVEC P., THEOBALT C., VALENTIN J., RHEMANN C.: Deep reflectance fields: high-quality facial reflectance field inference from color gradient illumination. *ACM Transactions on Graphics* 38, 4 (July 2019), 1–12. doi:10.1145/3306346.3323027.

- [MKI03] MIYAZAKI, KAGESAWA, IKEUCHI: Polarization-based transparent surface modeling from two views. In *Proceedings Ninth IEEE International Conference on Computer Vision* (2003), pp. 1381–1386 vol.2. doi:10.1109/ICCV.2003.1238651.
- [MKZ*21] MA X., KANG K., ZHU R., WU H., ZHOU K.: Free-form scanning of non-planar appearance with neural trace photography. *ACM Trans. Graph.* 40, 4 (July 2021). doi:10.1145/3450626.3459679.
- [MMZ*18] MEKA A., MAXIMOV M., ZOLLHÖFER M., CHATTERJEE A., SEIDEL H.-P., RICHARDT C., THEOBALT C.: Lime: Live intrinsic material estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 6315–6324. doi:10.1109/CVPR.2018.00661.
- [MP43] MCCULLOCH W. S., PITTS W.: A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics* 5, 4 (1943), 115–133.
- [MPBM03] MATUSIK W., PFISTER H., BRAND M., MCMILLAN L.: A data-driven reflectance model. *ACM Trans. Graph.* 22, 3 (July 2003), 759–769. doi:10.1145/882262.882343.
- [Mue43] MUELLER H.: The foundation of optics. *Journal of the Optical Society of America* 33, 12 (1943), 503–509.
- [MWL*99] MARSCHNER S. R., WESTIN S. H., LAFORTUNE E. P. F., TORRANCE K. E., GREENBERG D. P.: Image-based brdf measurement including human skin. In *Rendering Techniques '99: Proceedings of the Eurographics Workshop in Granada, Spain, June 21–23, 1999* (1999), Springer Vienna, pp. 131–144. doi:10.1007/978-3-7091-6809-7_13.
- [ND21] NICHOL A. Q., DHARIWAL P.: Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning* (18–24 Jul 2021), Meila M., Zhang T., (Eds.), vol. 139 of *Proceedings of Ma-*

- chine Learning Research*, PMLR, pp. 8162–8171. doi:10.48550/arXiv.2102.09672.
- [Nic65] NICODEMUS F. E.: Directional reflectance and emissivity of an opaque surface. *Applied optics* 4, 7 (1965), 767–775. doi:10.1364/ao.4.000767.
- [NLGK18] NAM G., LEE J. H., GUTIERREZ D., KIM M. H.: Practical svbrdf acquisition of 3d objects with unstructured flash photography. *ACM Trans. Graph.* 37, 6 (Dec. 2018). doi:10.1145/3272127.3275017.
- [ON94] OREN M., NAYAR S. K.: Generalization of lambert’s reflectance model. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1994), SIGGRAPH ’94, Association for Computing Machinery, p. 239–246. doi:10.1145/192161.192213.
- [OSF*18] OKTAY O., SCHLEMPER J., FOLGOC L. L., LEE M., HEINRICH M., MISAWA K., MORI K., McDONAGH S., HAMMERLA N. Y., KAINZ B., GLOCKER B., RUECKERT D.: Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv: 1804.03999* (2018). arXiv:1804.03999.
- [PH10] PHARR M., HUMPHREYS G.: *Physically Based Rendering, Second Edition: From Theory To Implementation*, 2nd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2010.
- [Pho75] PHONG B. T.: Illumination for computer generated pictures. *Commun. ACM* 18, 6 (June 1975), 311–317. doi:10.1145/360825.360839.
- [PJBM22] POOLE B., JAIN A., BARRON J. T., MILDENHALL B.: Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).
- [PNS18] PARK J. J., NEWCOMBE R., SEITZ S.: Surface light field fusion. In *2018 International Conference on 3D Vision (3DV)* (2018), pp. 12–21. doi:10.1109/3DV.2018.00013.
- [Pro15] PROJECT T. W.: Digital emily 2 dataset, 2015.

- [PSR*17] PANG J., SUN W., REN J. S., YANG C., YAN Q.: Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)* (Oct. 2017), IEEE, p. 878–886. doi:10.1109/iccvw.2017.108.
- [RBL*22] ROMBACH R., BLATTMANN A., LORENZ D., ESSER P., OMMER B.: High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022), IEEE, p. 10674–10685. doi:10.1109/cvpr52688.2022.01042.
- [RFB15] RONNEBERGER O., FISCHER P., BROX T.: U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2015), vol. 9351 of *LNCS*, Springer, pp. 234–241. doi:10.1007/978-3-319-24574-4_28.
- [RGB*20] RIVIERE J., GOTARDO P., BRADLEY D., GHOSH A., BEELER T.: Single-shot high-quality facial geometry and skin appearance capture. *ACM Transactions on Graphics* 39, 4 (Aug. 2020). doi:10.1145/3386569.3392464.
- [RPG14] RIVIERE J., PEERS P., GHOSH A.: Mobile surface reflectometry. In *ACM SIGGRAPH 2014 Posters* (July 2014), SIGGRAPH '14, ACM, p. 1–1. doi:10.1145/2614217.2630589.
- [RRFG17] RIVIERE J., RESHETOUSKI I., FILIPI L., GHOSH A.: Polarization imaging reflectometry in the wild. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–14. doi:10.1145/3130800.3130894.
- [RZ10] ROMEIRO F., ZICKLER T.: Blind reflectometry. In *Computer Vision – ECCV 2010* (2010), Springer Berlin Heidelberg, p. 45–58. doi:10.1007/978-3-642-15549-9_4.
- [Sch94] SCHLICK C.: An inexpensive brdf model for physically-based rendering. *Computer Graphics Forum* 13, 3 (1994), 233–246. doi:10.1111/1467-8659.1330233.

- [SDWMG15] SOHL-DICKSTEIN J., WEISS E., MAHESWARANATHAN N., GANGULI S.: Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)* (2015), pp. 2256–2265. doi:10.48550/arXiv.1503.03585.
- [SHC*22] SAHARIA C., HO J., CHAN W., SALIMANS T., FLEET D. J., NOROUZI M.: Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), 1–14. doi:10.1109/tpami.2022.3204461.
- [SHK*14] SCHARSTEIN D., HIRSCHMÜLLER H., KITAJIMA Y., KRATHWOHL G., NEŠIĆ N., WANG X., WESTLING P.: High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition* (2014), Springer International Publishing, p. 31–42. doi:10.1007/978-3-319-11752-2_3.
- [SRT16] SMITH W. A. P., RAMAMOORTHY R., TOZZA S.: Linear depth estimation from an uncalibrated, monocular polarisation image. In *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII* (2016), Springer International Publishing, pp. 109–125. doi:10.1007/978-3-319-46484-8_7.
- [SSG*17] SCHOPS T., SCHONBERGER J. L., GALLIANI S., SATTLER T., SCHINDLER K., POLLEFEYS M., GEIGER A.: A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), IEEE, p. 2538–2547. doi:10.1109/cvpr.2017.272.
- [SSS*24] SAITO S., SCHWARTZ G., SIMON T., LI J., NAM G.: Relightable gaussian codec avatars. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2024), IEEE, p. 130–141. doi:10.1109/cvpr52733.2024.00021.

- [Sto52] STOKES G. G.: On the composition and resolution of streams of polarized light from different sources. *Transactions of the Cambridge Philosophical Society* 9 (1852), 399–416. doi:10.1017/cbo9780511702266.010.
- [SYLK18] SUN D., YANG X., LIU M.-Y., KAUTZ J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (June 2018), IEEE, p. 8934–8943. doi:10.1109/cvpr.2018.00931.
- [SZ15] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2015). arXiv:1409.1556.
- [TD20] TEED Z., DENG J.: Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II* (Cham, 2020), Vedaldi A., Bischof H., Brox T., Frahm J.-M., (Eds.), vol. 12347 of *Lecture Notes in Computer Science*, Springer International Publishing, pp. 402–419. doi:10.1007/978-3-030-58536-5_24.
- [THZ*21] TANKOVICH V., HANE C., ZHANG Y., KOWDLE A., FANELLO S., BOUAZIZ S.: Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021), IEEE. doi:10.1109/cvpr46437.2021.01413.
- [TKR16] TAAMAZYAN V., KADAMBI A., RASKAR R.: Shape from mixed polarization. *arXiv preprint arXiv:1605.02066* (2016).
- [TS67] TORRANCE K. E., SPARROW E. M.: Theory for off-specular reflection from roughened surfaces*. *J. Opt. Soc. Am.* 57, 9 (Sep 1967), 1105–1114. doi:10.1364/JOSA.57.001105.

- [War92] WARD G. J.: Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph.* 26, 2 (July 1992), 265–272. doi:10.1145/142920.134078.
- [Whi05] WHITTED T.: An improved illumination model for shaded display. In *ACM SIGGRAPH 2005 Courses on - SIGGRAPH '05* (2005), SIGGRAPH '05, ACM Press. doi:10.1145/1198555.1198743.
- [WMLT07] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.: Microfacet Models for Refraction through Rough Surfaces. In *Rendering Techniques* (2007), Kautz J., Pattanaik S., (Eds.), The Eurographics Association. doi:/10.2312/EGWR/EGSR07/195–206.
- [Wol90] WOLFF L. B.: Surface orientation from two camera stereo with polarizers. In *Optics, Illumination, and Image Sensing for Machine Vision IV* (Apr. 1990), Svetkoff D. J., (Ed.), vol. 1194, SPIE, p. 287. doi:10.1117/12.969861.
- [WWZ16] WU H., WANG Z., ZHOU K.: Simultaneous localization and appearance estimation with a consumer rgb-d camera. *IEEE Transactions on Visualization and Computer Graphics* 22, 8 (2016), 2012–2023. doi:10.1109/TVCG.2015.2498617.
- [WZ15] WU H., ZHOU K.: Appfusion: Interactive appearance acquisition using a kinect sensor. *Computer Graphics Forum* 34, 6 (2015), 289–298. doi:10.1111/cgf.12600.
- [XDPT16] XIA R., DONG Y., PEERS P., TONG X.: Recovering shape and spatially-varying surface reflectance under unknown illumination. *ACM Trans. Graph.* 35, 6 (Dec. 2016). doi:10.1145/2980179.2980248.
- [XLZ*23] XU X., LIN Y., ZHOU H., ZENG C., YU Y., ZHOU K., WU H.: A unified spatial-angular structured light for single-view acquisition of shape and reflectance. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2023), IEEE, p. 206–215. doi:10.1109/cvpr52729.2023.00028.

- [XZ20] XU H., ZHANG J.: Aanet: Adaptive aggregation network for efficient stereo matching. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020), IEEE, p. 1956–1965. doi:10.1109/cvpr42600.2020.00203.
- [XZC*23] XU H., ZHANG J., CAI J., REZATOFIGHI H., YU F., TAO D., GEIGER A.: Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 11 (2023), 13941–13958. doi:10.1109/TPAMI.2023.3298645.
- [YMHR19] YANG G., MANELA J., HAPPOLD M., RAMANAN D.: Hierarchical deep stereo matching on high-resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 5515–5524. doi:10.1109/cvpr.2019.00566.
- [YSN*18] YAMAGUCHI S., SAITO S., NAGANO K., ZHAO Y., CHEN W., OLSZEWSKI K., MORISHIMA S., LI H.: High-fidelity facial reflectance and geometry inference from an unconstrained image. *ACM Trans. Graph.* 37, 4 (jul 2018). doi:10.1145/3197517.3201364.
- [YZM*24] YANG H., ZHENG M., MA C., LAI Y.-K., WAN P., HUANG H.: Vrmr: A volumetric relightable morphable head model. In *ACM SIGGRAPH 2024 Conference Papers* (2024), pp. 1–11. doi:10.1145/3641519.3657406.
- [ZJP*23] ZHENG J., JANG Y., PAPAIOANNOU A., KAMPOURIS C., POTAMIAS R. A., PAPANTONIOU F. P., GALANAKIS E., LEONARDIS A., ZAFEIRIOU S.: Ilsh: The imperial light-stage head dataset for human head view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 1112–1120. doi:10.1109/iccvw60793.2023.00119.
- [ZZSC22] ZHUANG Y., ZHU H., SUN X., CAO X.: Mofanerf: Morphable facial neural radiance field. In *European conference on computer vision* (2022), Springer, pp. 268–285. doi:10.1007/978-3-031-20062-5_16.

Appendix A

Appendix

A.1 Mobilephone Image Pre-processing

The thesis is based on using commercially available devices to enable ordinary users to conduct their own appearance capture. None of the captures involved high-performance DSLR cameras, laboratory mechanical cameras, or other professional equipment. In the vast majority of cases, we employed mobile phones for image capture. Compared to DSLR cameras, mobile phone lenses have limited performance and fewer adjustable parameters. The aperture size of each lens in a multi-lens smartphone camera is fixed and cannot be adjusted by the user. Many specialised professional cameras today are highly advanced and assist users by performing necessary pre-processing. However, mobile phones and other basic capture devices may lack these capabilities. In the appendix, we will outline some necessary steps for pre-processing after capturing images using mobile phones or other lightweight devices to achieve the highest possible image quality, facilitating further usage.

Distortion Correction

Mobile phone lenses are compact, and their optical design is constrained by space and cost. To achieve a wide field of view or high zoom levels, mobile phone lenses typically employ wide-

angle designs, which are prone to distortion, particularly barrel distortion (where straight lines near the edges of the image are stretched into arcs). Additionally, compared to professional DSLR cameras, mobile phones have smaller image sensors, meaning that light must focus on a smaller sensor at wide angles. Due to the size limitations of mobile phone lenses, the number of lens elements is also restricted, further exacerbating distortion issues.

Distortion is a type of aberration, describing the phenomenon where straight lines on an object appear curved after passing through the lens. The angle of the light rays relative to the optical axis causes changes in the lens's magnification, resulting in distortion. This effect is especially pronounced when the lens has strong refractive power. While distortion may be less noticeable when using telephoto lenses, it is highly prominent with wide-angle lenses. If we aim to combine information from different lenses, distortion correction becomes essential.

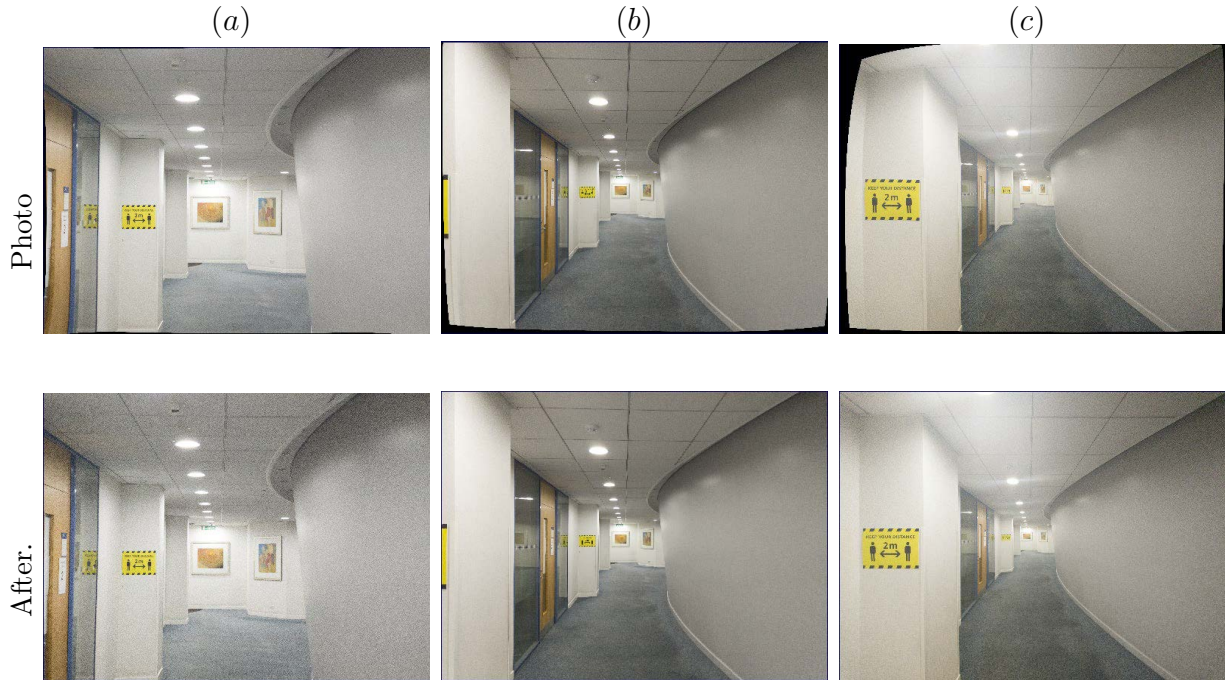


Figure A.1: Images were taken with three different lenses of the mobile phone, and the results after distortion correction. (a) telephoto lens. (b) wide-angle lens (c) ultra-wide-angle lens.

Currently, the principles and algorithms for camera lens correction are well-developed. By capturing a set of black-and-white checkerboard images, we can directly calculate the camera's intrinsic parameters and distortion coefficients. These parameters can then be applied to the images captured by the camera to produce corrected output. For example, the correction results for the three lenses of the OPPO Find X3 Pro are shown in Figure A.1. The images were

taken in the third-floor corridor of the Huxley Building. It can be observed that the telephoto lens exhibits minimal distortion, while the ultra-wide-angle lens shows the most significant distortion. It can also be noted that some pixels at the edges of the image are lost. Therefore, when observing the test object, it is advisable to place the object in the centre of the lens. In the corrected images, the overall alignment is restored, and the door frame and floor return to their vertical orientation.

HDR Image Composition

In everyday life, high dynamic range (HDR) images are no longer considered highly specialised or unfamiliar. When capturing a photo of an object or a landscape, the primary consideration is to prevent excessive light from entering the shutter, which can cause overexposure. This can be adjusted by either reducing the exposure time or decreasing the aperture size. However, such adjustments not only darken the brighter areas of the scene but also render the darker areas even less visible due to insufficient light. Therefore, HDR imaging can be employed in many scenarios to preserve as much detail as possible in the darker regions of an image.

To create an HDR image, multiple photos of the same scene are captured, each with a different exposure value (EV) (typically one with normal exposure, one overexposed, and one underexposed). These photos are then combined into a single composite image. The composition process begins by aligning the images and subsequently merging them based on the weighted brightness areas of each image. In the bright regions, the underexposed image has a greater weight, while in the darker regions, the overexposed image is given more weight. This approach preserves more details across the entire image. The resulting composite image can be saved in common formats through tone mapping for display on standard monitors or saved directly in a floating-point format to retain as much information as possible.

Many modern smartphones now include algorithms for automatically capturing HDR images, but they typically do not provide users with access to specific exposure settings. If further adjustments are required, HDR image construction must be done manually by capturing multiple shots.

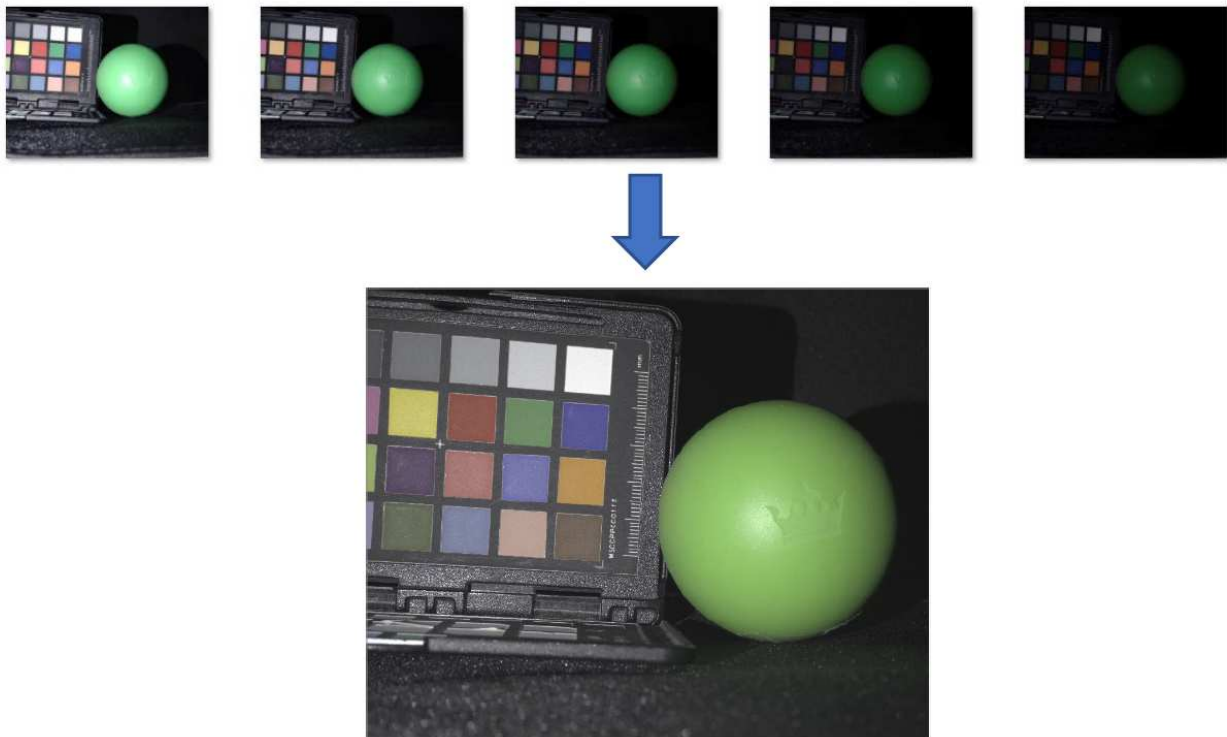


Figure A.2: One example using five photos with different exposure times to synthesise HDR images

Colour Correction and White Balance

White balance is a standard step in image processing. Due to different lighting conditions and camera settings, the colour temperature and tone of an image may vary. White balance adjusts the colours in a photo to bring them closer to their natural state. Whether using a smartphone or another imaging device, the process of capturing colours from the sensor and rendering them into an image is typically automatic. Different manufacturers and lenses have distinct settings for colour generation, which can result in significant variations. When possible, the best approach for multi-angle shooting is to use a single camera. If specific angles must be fixed, it is advisable to use cameras of the same model to set up the scene for multi-angle shots, as this helps minimise colour differences and reduces interference caused by colour discrepancies.

Additionally, manual white balance correction can also be beneficial. By placing a colour chart within the field of view of the object being measured and adjusting the pixels in the image to match the colours indicated on the chart, colour bias can be minimised.



Figure A.3: One example for a green fitness ball photos before and after white balance processing (a): Original image; (b): After white balance processing

When experimenting with multiple lenses on smartphones, we clearly observed differences in colour when capturing the same object with different lenses. This objective variation can also affect subsequent image-based methods. Notably, Apple's iPhone series performs better in this regard compared to OPPO phones.

A.2 Rendering

Realistic rendering is a crucial component throughout this thesis. Our goal is to provide better input for the rendering process to achieve ultimately realistic results. Furthermore, rendering is not just the final step of the entire pipeline. Throughout the thesis, we use synthetic rendering data to prepare datasets for training neural networks, and we also introduce the rendering process during training to add additional loss terms that improve results. Therefore, we dedicate this section to introducing rendering.

Rendering is a fundamental process in computer graphics, responsible for converting 3D models into 2D images by simulating the interaction of light with materials and geometry. Through a series of algorithms and processes, it processes geometric information, materials, lighting, and other data in a virtual scene to generate the final image displayed on the screen. The main tasks of rendering include loading models, managing scenes, calculating lighting and shadows, and composing images. Depending on the specific purpose of the rendering, the process can be real-time (e.g., for games and virtual reality applications) or offline (e.g., for movie effects and architectural visualisation). The technical requirements and implementation methods of rendering may vary across different applications. For example, in real-time rendering, performance optimisation is crucial, whereas in offline rendering, achieving photorealism is often the key goal.

The development of rendering technology can be traced back to the early 1960s. The earliest rendering methods were based on simple rasterization algorithms, such as the Z-buffer algorithm. As computer graphics evolved, more complex rendering techniques were introduced. In 1979, Turner Whitted proposed the ray tracing algorithm [Whi05], marking an important milestone in the history of rendering technology. Ray tracing is a physically-based rendering technique that generates images by simulating the interaction between light and objects, producing highly realistic visual effects. It can simulate complex optical phenomena such as shadows, reflections, and refractions. Although computationally expensive, ray tracing laid the foundation for modern offline rendering.

Entering the 21st century, with rapid advancements in hardware, Physically-Based Rendering (PBR) has gradually become mainstream. This method more accurately simulates the interaction between light and materials in the physical world. At the same time, real-time rendering technology has also made significant progress and is widely applied in video games and virtual reality.

A.2.1 Basic Elements of Rendering

The basic rendering process involves converting the geometric, lighting, and material information of a 3D scene into a 2D image. In other words, it entails providing these settings or inputs to the computer and using graphics algorithms to generate the desired 2D image. This process encompasses several aspects, including geometry modelling, materials and textures, lighting models, and camera models, all of which play a crucial role in image generation. The Spatially Varying Bidirectional Reflectance Distribution Function introduced in the previous section has already addressed a significant part of this work. Here, we will introduce the remaining elements before discussing the entire rendering process step by step.

Geometry Modeling

Geometry modelling is the first step in rendering, determining the shape and structure of objects in a scene. Polygon meshes primarily describe the vertices, edges, and faces of objects in 3D space. However, other geometric representations, such as NURBS surfaces or voxels, rely on different mathematical or volumetric structures to define shape and structure, offering flexibility for various applications and levels of detail. Among them, polygon meshes are the most prevalent geometric model, typically consisting of many triangles, making them suitable for representing complex shapes—an ideal choice for capturing the irregular appearance of real-world objects. Surface models use mathematical formulas to define smooth surfaces and are commonly employed in industrial design and CAD applications, such as drawing an I-beam structure. Voxel models, on the other hand, use 3D pixels to represent volume data and are suitable for applications such as medical imaging and scientific visualisation.

Light Sources

Light sources are a key element in rendering, determining the lighting effects in a scene. Different types of light sources affect the appearance of shadows, reflections, and diffuse lighting. Common light sources in rendering include point lights, directional lights, spotlights, and ambient lights. Each has different physical properties and lighting effects suited to various rendering scenarios.

- **Point Lights:** Point lights are the simplest light source model. They have no volume in 3D space and emit light uniformly in all directions from a single point. The intensity of point lights decreases with distance, typically following the inverse square law: if the distance doubles, the light intensity is reduced to one-quarter of its original value. Point lights are ideal for simulating small real-world light sources, such as bulbs or candlelight. In Chapter 3, we simulate lighting using a smartphone flash, which falls into this category.
- **Directional Lights:** Directional lights simulate distant light sources, such as the sun. Since the light source is infinitely far from the scene, all rays reaching the object's surface are parallel, and the light intensity does not attenuate over distance. Directional lights are commonly used to simulate uniform lighting over a wide area, such as sunlight in outdoor scenes.
- **Spotlights:** Spotlights are a type of point light source with a constrained beam. They concentrate light in a specific direction and have a defined cone angle, meaning the light is strongest in the central area and gradually weakens as it moves away. Spotlights are often used to simulate stage lighting, flashlights, or other situations where focused beams are needed. To avoid confusion, Figure A.5 is provided to compare the effects of spotlights and directional lights in an imaging scene.
- **Ambient Light:** Ambient light is a special light source that does not simulate specific light paths but represents indirect lighting in a scene. It provides a base level of brightness, ensuring that objects are still visible even in areas without direct lighting.

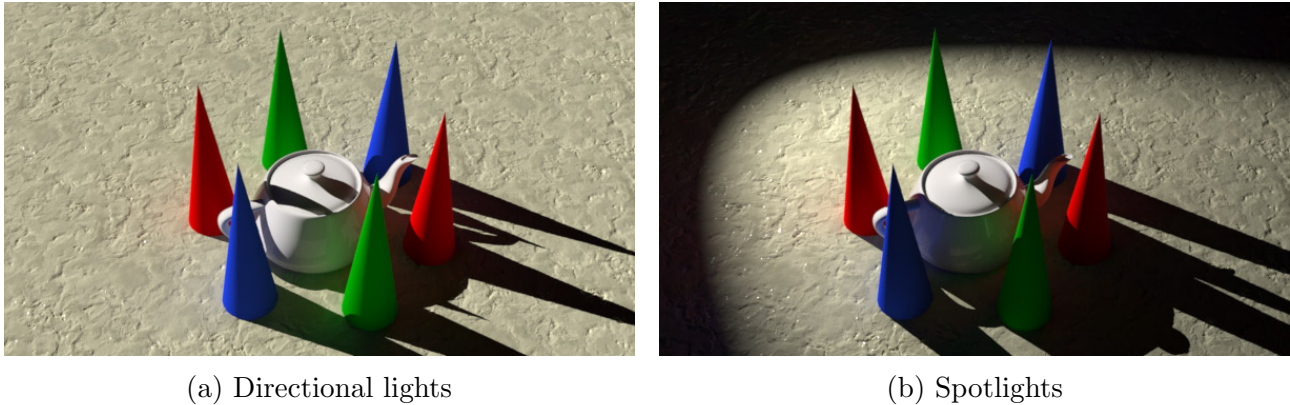


Figure A.4: A simple scene with an object and several polygons, demonstrating the difference between directional light and spotlight illumination.

In addition to directly setting light sources in the scene, using environment light maps is a more common way to simulate complex lighting. Light maps are a precomputed lighting technique in which lighting information is calculated before rendering using methods such as ray tracing or path tracing. This lighting information is stored in a light map (typically a 2D texture), where each texel contains precomputed lighting data for a specific area. During rendering, objects in the scene map these light textures to receive their lighting information, bypassing the need for real-time complex light calculations. Light maps are widely used in game engines and virtual reality, particularly in static scenes where performance is a concern.

Moreover, the biggest advantage of using light maps is that they can incorporate actual measured lighting data. By photographing mirrored spheres, an HDR environment map can be created, capturing real-world lighting conditions that can be directly used in rendering. Natural scenes often involve multiple reflections and scattering of light, which are difficult to simulate manually. By using real-world environment maps, rendering can automatically achieve these complex lighting effects, such as soft light reflected from the sky or buildings.

Camera Model

With the objects (geometry model), light sources, and the method of interaction between light and objects (as discussed in the previous section on BRDF), we also need to specify the viewpoint from which we observe the object, i.e., how to project the 3D scene onto a 2D screen

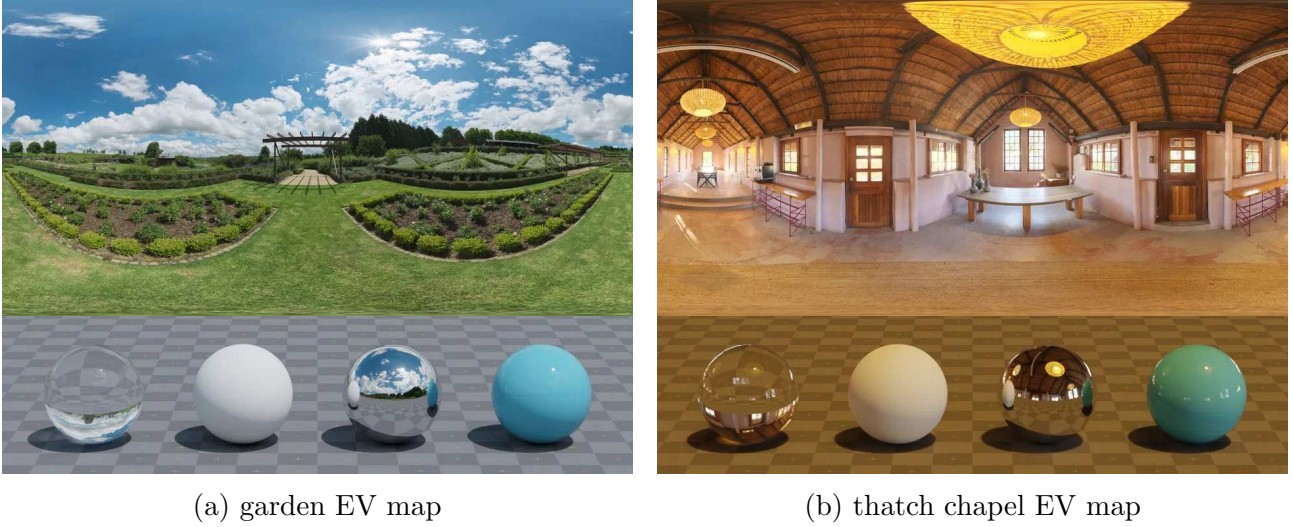


Figure A.5: Two examples of light maps. (a) Garden EV map representing an outdoor environment, and (b) Thatch Chapel EV map representing an indoor environment. Both maps were captured by artists and sourced from the free, open-source visual sharing website, Polyhaven.

using a virtual camera.

The camera model simulates the operation of a real-world camera and defines parameters such as the viewpoint, field of view, and focal length. Common camera models include two types:

- **Perspective Projection:** Perspective projection simulates the effect of objects appearing smaller as they move farther from the camera. In perspective projection, distant objects appear smaller, while closer objects appear larger. When rendering target objects to be used as training data, we use this type of projection method to replicate real-world conditions as closely as possible.
- **Orthographic Projection:** Orthographic projection does not create a size difference for objects at varying distances, meaning all objects are scaled proportionally to their actual size. This type of projection is often used in engineering drawings or special effects rendering.

For orthographic projection, the camera operates in an idealised abstract state. For perspective projection, camera parameters such as field of view, position, and orientation control how the scene is displayed on the screen, much like how a photographer adjusts their camera in the real world. The field of view determines how much of the scene the camera can capture, usually

expressed in degrees. A larger field of view allows the camera to see more of the scene, while a smaller field of view magnifies the scene, focusing on more details—similar to the difference between wide-angle and telephoto lenses in real-world photography. A wide-angle view (e.g., 35 degrees or less) creates perspective distortion, making nearby objects appear larger and distant objects appear smaller, enhancing the sense of depth in the scene. A narrow-angle view (e.g., 20 degrees or less) reduces the field of view, making objects appear more compact, which is useful for closely inspecting specific objects or details.

The camera position and angle are relatively straightforward to describe. The camera's position determines which objects appear in the frame and their spatial relationships. In 3D space, the camera can be moved and positioned along the X, Y, and Z axes, while the camera's orientation is usually represented by the pitch, yaw, and roll angles. By adjusting the camera's field of view, position, and orientation, the rendering system can generate the desired image view.

A.2.2 Basic Rendering Process

Rendering technology has developed to the point where many rendering engines have emerged, such as Unity, Unreal Engine, and Blender. These rendering engines are widely used in different application scenarios and can produce high-quality rendering results. Rendering engines can be broadly divided into two categories: real-time rendering engines (e.g., game engines) and offline rendering engines (e.g., renderers used for movie effects and architectural visualisation). Real-time rendering engines prioritise rendering efficiency to ensure interactivity and smoothness, while offline rendering engines focus more on realism and detail, often using more complex ray tracing and global illumination algorithms to generate highly realistic effects.

Using OpenGL's rendering pipeline as an example, the rendering process can be broken down into several stages. Through these stages, we can see how the settings discussed earlier impact the final rendering result. OpenGL is a rasterisation-based rendering method. The main advantage of rasterisation is its extremely high computational efficiency, which can be hardware-accelerated on modern GPUs. It is also very suitable for handling a large number of triangles and geometries, making it well-suited for complex scenes. Through rasterisation, the GPU can

efficiently project thousands of triangles onto the screen and process every pixel in a short amount of time.

OpenGL’s rendering pipeline is divided into several parts, with the user-configurable components typically concentrated in several shaders.

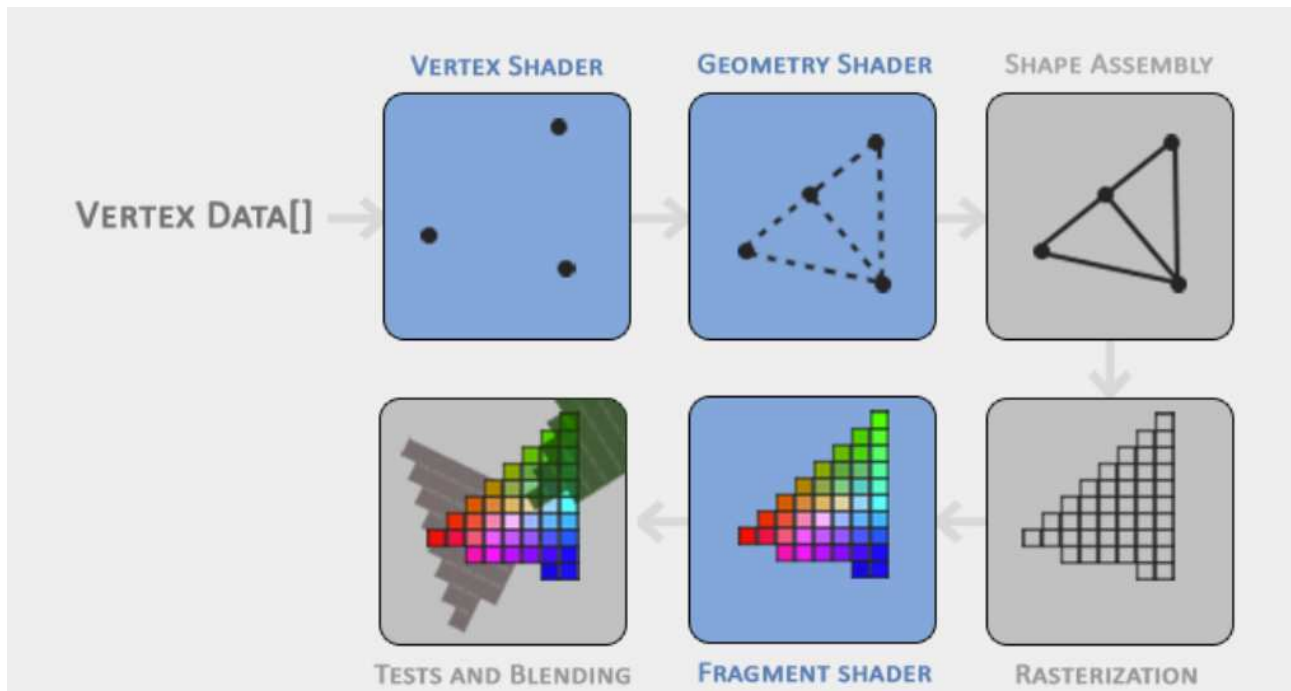


Figure A.6: Key stages in the OpenGL rendering pipeline, illustrating the flow from vertex data to rasterization. Image sourced from the open-source tutorial LearnOpenGL.

The graphics pipeline is quite complex and contains many configurable parts. However, in most cases, we only need to work with the vertex and fragment shaders. The geometry shader is optional and usually left to its default configuration.

The first stage to focus on is vertex shading. In this stage, the 3D geometric data of the objects in the scene is input into the rendering pipeline. The user needs to specify in the vertex shader how the engine processes these vertices (each vertex in the polygon mesh of the object to be rendered). The engine transforms them from local coordinate space to world coordinate space and finally into screen space under the camera view, mapping the object’s vertices from 3D space to 2D screen space. This step involves geometric transformations, such as model transformation, view transformation, and projection transformation (e.g., perspective projection or orthographic projection), which determine the object’s position and size on the screen.

The camera's field of view, position, and orientation play a role at this stage, determining how we observe objects in the scene from a specific angle. The camera settings also affect the perspective and field of view of the scene.

After vertex shading, the vertices in 3D space are assembled into basic geometric primitives (e.g., triangles). These primitives are then converted into 2D pixels (or fragments) in a process known as rasterisation. During rasterisation, the geometric information of the triangles is mapped onto a pixel grid on the screen. This step mainly converts 3D objects into 2D pixels, determining which objects will be rendered and which will be occluded. Additionally, Z-buffer technology is used to handle depth information, ensuring correct visibility sorting.

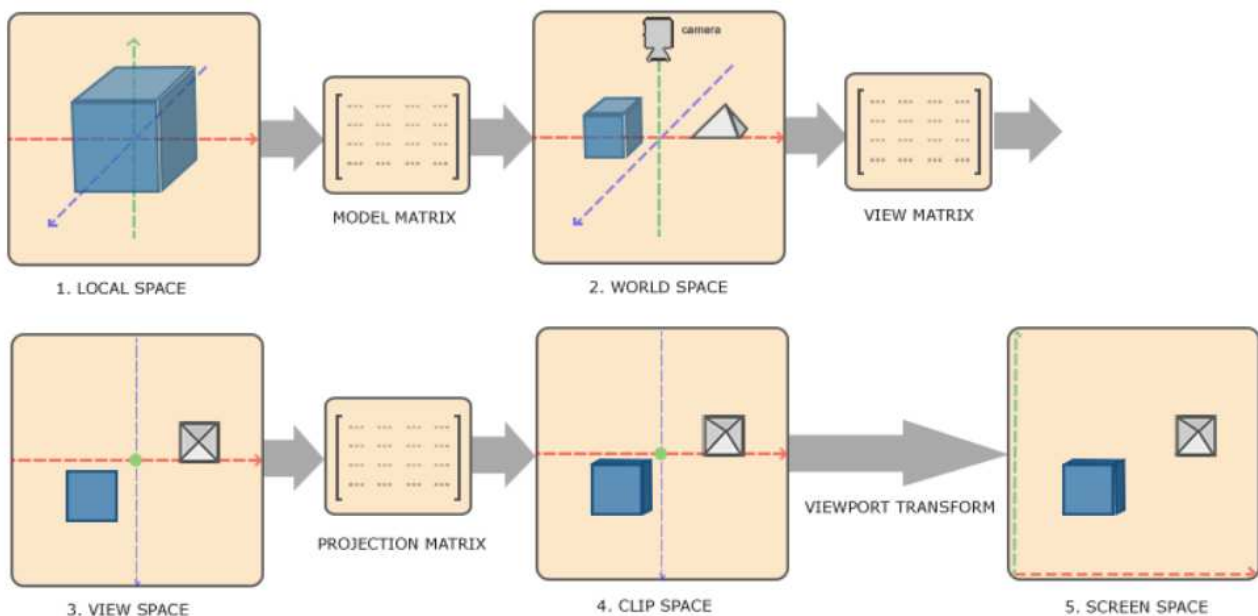


Figure A.7: Coordinate transformations in the vertex shader, illustrating transformations from local space to screen space through model, world, view, and projection matrices. Image sourced from the open-source tutorial LearnOpenGL.

After the fragments are generated during rasterisation, the fragment shader is responsible for computing the colour value of each fragment (pixel). At this stage, the scene's lighting model and material properties are applied. The light source types (e.g., point light, directional light, ambient light) and the material's BRDF determine the reflection and shadow effects on the object's surface. This is where the complex interaction models between light and objects, which we discussed earlier, are defined. The rendering engine automatically applies the user-defined calculations to each pixel, ultimately determining the colour value of each pixel on the

screen.

The entire process may seem straightforward. However, in reality, whether the rendered object appears realistic or photo-like largely depends on the quality of the SVBRDF maps provided to the rendering engine and model. Once we have highly detailed and high-resolution albedo texture maps, we can freely adjust the camera position, change lighting conditions, zoom in and out of the scene, and make any other modifications as needed.

In recent years, with the advent of hardware-accelerated ray tracing technologies like NVIDIA RTX, real-time ray tracing has gradually become mainstream. By combining the advantages of rasterisation and ray tracing, modern rendering engines can achieve real-time effects such as reflections, refractions, and global illumination in games. This marks a new level of performance and realism in rendering engines, making it possible to generate movie-quality visuals in real-time applications. Rendering engines have become indispensable tools in fields such as gaming, film, and architectural visualisation. Our research utilises rendering technology as a powerful tool, so we will not delve further into it here. In Chapter 3 and Chapter 4, we extensively demonstrate rendering results based on the material maps and shapes we acquired, all made possible by the strong support of rendering engines (specifically OpenGL).

A.3 Deep Learning

This section will present some high-level general information about deep learning. Deep learning is a member of the machine learning family and is based on artificial neural networks (ANNs). The term "deep" refers to the use of multiple layers in the neural network. The enormous impact that deep learning has made on our lives hardly needs elaboration. Its origins can be traced back to 1943 when McCulloch and Pitts [MP43] introduced a neural network computational model using circuits, which described how neurons in the brain function.

Simply put, for a node in a neural network, we provide an input (denoted as x), pass it through a node (a neuron), and produce an output (denoted as y). By doing this, we establish a mapping between inputs and outputs. In practice, each neuron structure can take multiple inputs $x_1, x_2, \dots, x_n$, each multiplied by its corresponding weight $w_1, w_2, \dots, w_n$, and the results are summed with a bias term b . The sum is then passed through an activation function f to produce the final output y_{out} . This method allows us to map the input to the desired output by weighting the factors that influence the result.

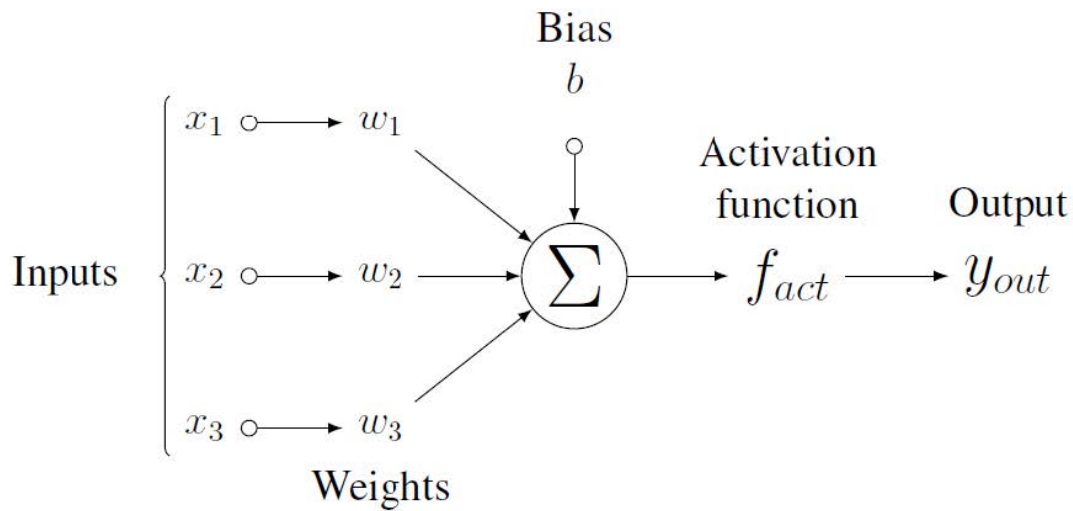


Figure A.8: A simplified diagram of a single-layer perceptron neural network. The diagram illustrates the process of combining multiple inputs x_1, x_2, x_3 with their respective weights w_1, w_2, w_3 , adding a bias term b , and applying an activation function f_{act} to produce the output y_{out} .

Figure A.8 illustrates a single-layer neuron, while a complete neural network is formed by

stacking these individual neurons together. Like building blocks or constructing a house, using simple repetitive structures does not prevent one from achieving impressive functionality. By adding more layers between the input and output, the network becomes more powerful and capable of handling more complex tasks.

The growing interest in deep learning is not difficult to understand. Compared to other machine learning methods, such as SVM (Support Vector Machines), deep learning is better suited to handle the current growth in data size and computational power. Furthermore, the almost yearly updates to GPU graphics cards provide the hardware support for scientists to explore more complex network structures. For instance, the RTX 4090, released in September 2022, features 76 billion transistors, 16,384 CUDA cores, and 24 GB of high-speed Micron GDDR6X memory. A deep learning network can better and more efficiently utilise these computational resources and vast amounts of data from around the world, unlike traditional machine learning algorithms that struggle with large-scale data and face performance bottlenecks.

A.3.1 Train a Network

After determining that we want to solve our problem using a deep learning network, we need to complete several steps to 'teach' the network how to handle the inputs we provide. This process is similar to how humans learn a new skill, so the operation that imparts a specific function to the network is called training.

In a traditional deep learning framework, the training process is composed of several steps. Let's take a classic example: providing the network with photos of cats or dogs, and having the network read the photos and return a decision, indicating whether the image is of a cat or a dog.

Before training, the first step is data preparation. This is the 'guidebook' the network uses for learning. Deep learning models typically require a large amount of training data. For image-related tasks, the input data is a set of images, such as photos of cats and dogs, with each image labelled as either a cat or a dog.

The entire dataset needs to be divided into three parts:

- Training set: Used to train the model.
- Validation set: Used to evaluate the model's performance during training.
- Test set: Used to evaluate the model's final effectiveness.

In fact, preparing the data is crucial for a network, and we will dedicate a subsection to it later.

Once the data is ready, network training can officially begin. For a network in its initial state, the first step is forward propagation. In the example of the cat-and-dog classification task, we input an image into the model and allow it to process the image data layer by layer, extracting features and ultimately producing a prediction. The model's output may be two numbers representing the probability that the image is of a cat or a dog. Since the network is just starting its training, the output may be quite inaccurate, resulting in a gap between the model's output and the true label. At this point, we introduce a loss function to quantify this difference. Common loss functions, such as cross-entropy loss and various others designed by researchers, measure the gap between the model's predicted result and the correct answer. Afterwards, based on the computed loss function, we perform backpropagation.

During the backpropagation phase, the model calculates the error in each layer's parameters (weights and biases) based on the loss function. The error is propagated back through the network using the chain rule, and optimization algorithms like gradient descent are employed to update the model's parameters. After each update, the model's loss value should gradually decrease. This process is repeated multiple times, usually over many epochs. In each epoch, the entire training set is used to adjust the model, with the goal of refining its parameters. The entire training process can be imagined as a cycle: input image $\rightarrow$ model prediction $\rightarrow$ error calculation $\rightarrow$ model parameter adjustment. After many iterations, the model becomes more accurate and better at recognizing or classifying the input images.

A.3.2 Prepare Training Data

If no special design is introduced, the model's performance will improve after a series of operations. This is easy to understand—updating the network's parameters after backpropagation reduces the loss, essentially guiding the network in the right direction. However, a single image cannot fully teach the network what it needs to do. For example, if the network only sees a picture of a cat on the grass (labelled as a cat), it might mistakenly establish a relationship such as 'images with a grassy background point to cats,' which is obviously incorrect. This illustrates a key point in deep learning: the quality and characteristics of the training set directly influence the training effectiveness and performance of a deep learning model.

A good training set typically possesses the following characteristics:

- Sufficient Sample Size

Deep learning models require large amounts of data, especially when the task is complex. A sufficient number of samples helps the model generalise better, meaning it can perform well on new data. Generally, the more data there is, the more accurately the model can learn patterns.

- Diverse Coverage

The training set should include various scenarios for the target task. For instance, in image classification, if the goal is to identify different breeds of cats and dogs, the training set should encompass images of cats and dogs of different breeds, poses, and lighting conditions.

- Representativeness of Real-World Data

The training set should resemble the data encountered in actual applications. If the data used for real-world applications differs greatly from the training set, the model's real-world performance may suffer. For example, if all the images in the training set are high-quality and clear, but the test data is blurry or of low quality, the model may not perform well.

- **Balanced Sample Distribution**

The training set should avoid the issue of class imbalance. For instance, in a classification task, if there are many more images of cats than dogs, the model might be biased towards predicting the images as cats. Therefore, the number of samples for each class should be roughly equal.

- **Data Diversity**

To ensure that the network doesn't only work under specific conditions, the dataset must be diverse. For image tasks, the training set should include images taken under different conditions, such as varying lighting, angles, backgrounds, and resolutions. This ensures that the model can still make accurate predictions when faced with data from different environments. If all the training images of cats and dogs were taken under fluorescent lighting, the model's predictions might be unreliable when faced with an image of a cat taken under moonlight. Additionally, there may be random noise or outliers in the dataset. The model needs to make correct judgments even under such conditions, so having a moderate amount of noisy data can help improve the model's robustness (resistance to interference).

- **Accurate Labels**

The labels for the training data must be precise. Incorrect labels will lead the model to learn incorrect information, reducing its performance. For instance, if some images of cats are mislabeled as dogs, the model might confuse the features of cats and dogs.

- **Proper Preprocessing**

The training set often requires preprocessing, such as normalisation (scaling pixel values to between 0-1 or -1 to 1), removing noise, and filling in missing values. Proper preprocessing can help the model converge faster and improve accuracy.

From the above discussion, it is clear that building a training set tailored to specific goals is not an easy task. Deep learning networks are not magic; they cannot create something out of nothing. Sometimes, they are not even that 'smart'—they cannot learn anything beyond what

they are explicitly taught (and sometimes they even struggle with individual examples within the dataset).

In addition to ensuring the quality of the dataset, data augmentation methods can also be employed. To increase the diversity of the training set, we can apply transformations such as flipping, rotating, scaling, cropping, and colour adjustments. This approach is especially effective when the training data is limited.

A.3.3 Deep Learning Frameworks

As neural networks have evolved, their internal structures and layers have become increasingly complex. The good news is that deep learning frameworks like PyTorch and TensorFlow are widely available, offering users a range of predefined functions and modules that greatly simplify the process of constructing neural networks. Users can easily call these modules to define common network layers, such as Convolutional Layers, Pooling Layers, and Fully Connected Layers. Additionally, these frameworks integrate commonly used loss functions (e.g., cross-entropy loss and mean squared error) and optimizers (e.g., SGD, Adam, RMSProp), so users no longer need to implement complex mathematical calculations and gradient update algorithms from scratch. Instead, they can directly use these modules for model training and optimization.

As neural network models have grown in size, traditional CPUs have become inefficient at handling large-scale computational tasks. The introduction of GPUs has therefore become critical for deep learning training. NVIDIA, as a leader in the GPU field, provides robust hardware support for deep learning development. NVIDIA launched the CUDA (Compute Unified Device Architecture) toolkit, specifically designed for parallel computing tasks on GPUs, enabling the acceleration of large-scale matrix computations and tensor operations. Both PyTorch and TensorFlow are deeply integrated with CUDA, allowing users to specify whether to use the GPU. These frameworks automatically allocate computational tasks to the GPU, significantly improving training speed.

In addition, NVIDIA has introduced the cuDNN (CUDA Deep Neural Network) library, a GPU

acceleration library specifically designed for deep learning, further optimizing convolution operations and gradient calculations in neural networks. For developers, there is no need to worry about low-level hardware optimizations—they can focus on designing network architectures and preparing datasets while deep learning frameworks and hardware libraries handle large-scale computation tasks automatically.

Our work, as described above, is based on the PyTorch and TensorFlow frameworks and developed using NVIDIA’s GPU hardware. This approach is currently employed by the vast majority of researchers.

A.3.4 Diffusion Model Principles

In this subsection, we will briefly introduce how the diffusion generative model works. The general idea is to create a network that can map from random noise to a target image, thereby achieving 'random generation.'

The network's ability to create something from nothing consists of two main steps. First, in the generation process, noise is gradually added to an original image until it resembles Gaussian noise. At each time step (usually determined by a preset number of T steps), the model adds a certain proportion of noise to the image, forming part of the Markov chain. By the time T steps are completed, and the image data is transformed into pure noise. The implementation of this step is not fixed; for example, in the Iterative (de)Blending model [HBC23], the noise injection process is achieved through alpha-blending operations.

$$x_\alpha = (1 - \alpha)x_0 + \alpha x_1 \tag{A.1}$$

where x_0 is a random Gaussian noise, x_1 is one of target images, α is a blending parameter that produces the noisy image. This operation gradually generates a blended image, proportionally combining high-quality images with noise. The process resembles transitioning from order to disorder, pushing the data closer to a standard Gaussian distribution.

The second step is the denoising process (reverse diffusion). In Diffusion Models, the key to generating images lies in how noise is gradually removed, reconstructing the original image. This reverse process is achieved by training a neural network, whose goal is to learn how to recover the image from the noisy state at each step. The model must estimate the amount of noise at each time step and apply denoising operations to restore the image's details. This step is crucial for the model's ability to 'generate' an image. Intuitively, such a deblending operation is impossible for a human to perform. Various combinations of images and noise can create these noisy images, and traditional methods cannot establish a mapping between two fixed points.

Fortunately, when this mapping is divided into many steps and iterated multiple times across

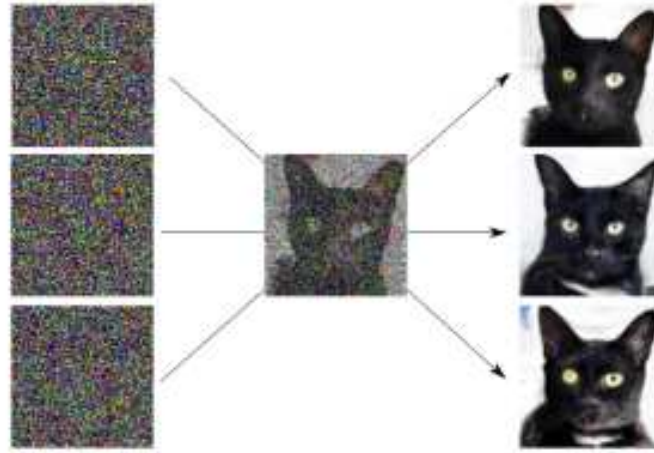


Figure A.9: Decomposition of a noisy image into different noise components and their respective reconstructions [HBC23]. The figure illustrates that the decomposition of a noisy image can be arbitrary.

time steps, a deterministic mapping emerges. The randomness linking noise and the final result converges to a deterministic path. This is similar to the Central Limit Theorem in statistics: averaging many random things eliminates randomness and converges to their expected value. Therefore, after training, the network can use this deterministic reverse mapping to gradually recover a noisy image into a clean image, allowing it to generate new target images from random noise.

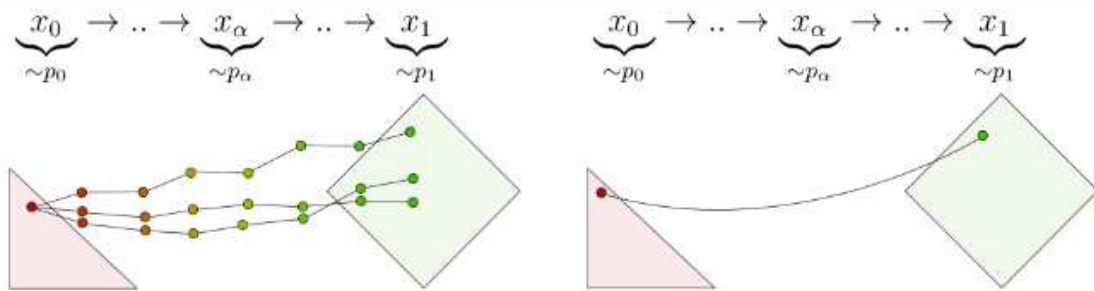


Figure A.10: An infinitely subdivided separation operation will eventually converge to a fixed path where the randomness is averaged out [HBC23].

A.4 Abbreviations

ANN	Artificial Neural Networks
AR	Augmented Reality
BRDF	Bidirectional Reflectance Distribution Function
BTF	Bidirectional Texture Function
CNN	Convolutional Neural Network
cuDNN	CUDA Deep Neural Network
DDPM	Denoising Diffusion Probabilistic Models
DOP	Degree of Polarisation
FOV	Field of View
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HDR	High Dynamic Range
IR	Infra-Red
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LiDAR	Light Detection and Ranging
MPI	Message Passing Interface
NeRF	Neural Radiance Fields
POI	Plane of Incidence
PWC-Net	Pyramid Warping and Cost volume Network
RAFT	Recurrent All-Pairs Field Transforms
ReLU	Rectified Linear Unit
RGB	Red Green Blue
RMS	Root Mean Squared
SIFT	Scale Invariant Feature Transform
SURF	Speeded-Up Robust Features

SVBRDF	Spatially-Varying BRDF
ToF	Time of Flight
TRS	Transmitted Radiance Sinusoid
U-Net	U-Net Convolutional Network
VAE	Variational Autoencoder
VGG	Visual Geometry Group
VR	Virtual Reality
RTX	Ray Tracing Texel Extreme