Deep 3D Reconstruction of Scenes with Complex Surface Materials

Opemipo Adegbemi

Master of Science in Computer Vision, Robotics and Machine Learning

from the

University of Surrey

Department of Electrical and Electronic Engineering

Faculty of Engineering and Physical Sciences

University of Surrey

Guildford, Surrey, GU2 7XH, UK

September 2022

Supervised by: Dr. Jean-Yves Guillemaut

DECLARATION OF ORIGINALITY

I confirm that the project dissertation I am submitting is entirely my own work and that any material used from other sources has been clearly identified and properly acknowledged and referenced. In submitting this final version of my report to the JISC anti-plagiarism software resource, I confirm that my work does not contravene the university regulations on plagiarism as described in the Student Handbook. In so doing I also acknowledge that I may be held to account for any particular instances of uncited work detected by the JISC anti-plagiarism software, or as may be found by the project examiner or project organiser. I also understand that if an allegation of plagiarism is upheld via an Academic Misconduct Hearing, then I may forfeit any credit for this module or a more severe penalty may be agreed.

MSc Dissertation Title: Deep 3D Reconstruction of Scenes with Complex Surface Materials

Author: Opemipo Adegbemi

Author Signature Date: September 6, 2022

Supervisor's name: Dr. Jean-Yves Guillemaut

WORD COUNT

Number of Pages: 58

Number of Words: 17, 739

ABSTRACT

Helmholtz Stereopsis is a method used in 3D reconstruction of scenes from multiple reciprocal pairs of images. A pair of images is reciprocal if the second image is captured with the same camera and light source used to capture the first image, but only after their positions have been swapped. This enables 3D reconstruction of scenes with arbitrary and unknown bidirectional reflectance distribution functions (BRDFs), which differentiates it from existing methods, such as multi-view stereo and photometric stereo, where the surfaces in the scene must be Lambertian or the BRDFs of the surfaces are known beforehand. Although, there have been significant improvements in the original methods, the cumbersome process of capturing the reciprocal pairs of images remains. The purpose of this project is to ease the set-up process by developing deep learning methods that can reconstruct complex scenes from reciprocal pairs of images. This dissertation presents a method for automatically generating reciprocal pairs of images to produce a Helmholtz dataset. 3D models of scenes were taken from a publicly available dataset and rendered in a controlled environment using a rendering engine, such that the environment has only a single light source and two cameras. One camera is used as the reference camera, while the other camera and light source are used to generate the reciprocal pairs of images of the scene. The generated reciprocal pairs of images form the Helmholtz dataset and eliminate the need to manually set up a Helmholtz environment. Furthermore, the dissertation presents a deep multi-view stereo method that learns from the Helmholtz dataset to estimate depths of scenes. The method adapts an existing deep multi-view stereo network to learn the mapping between reciprocal pairs of images of scenes and the corresponding depths of the scenes. The adapted network shows promising results in estimating the depths of simple to moderately-complex scenes, which can be extended to handle more complex scenes. Lastly, the dissertation presents a deep photometric stereo method that learns from the Helmholtz dataset to estimate the surface normals of scenes. The method adapts an existing deep photometric network to learn the mapping between reciprocal pairs of images of scenes and the corresponding surface normals of the scenes. The adapted network showed promising results in normal inference of scenes but perform best in simple models. Nevertheless, it provides a framework that can be extended to better estimate the surface normals of more complex scenes.

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

3D reconstruction is an active research area in computer vision with a wide range of applications in fields, such as game development, movie production, medicine etc. Due to its high demand, significant progress has been made towards developing methods that can reconstruct a variety of scenes with high accuracy and completeness.

1.1 Background and Context

Traditional methods of 3D reconstruction can be broadly classified into Multi-view Stereo, Photometric Stereo and Shape from Silhouette(sfS)[6]. Multi-view stereo reconstructs surfaces from multiple views and cannot accurately model non-Lambertian surfaces. Photometric stereo reconstructs an object from its surface normals by varying the illumination on its captured view. Although, they are able to model non-Lambertian surfaces, the reflectance has to be known beforehand. Shape from Silhouette methods model the shape of an object from its silhouettes, and they do not depend on the reflectance properties of the object. Yet, they can only reconstruct convex surfaces.

Helmholtz reciprocity states that the reflectance properties of a surface does not change if the incident light ray and outgoing light ray are swapped [69]. Helmholtz Stereopsis leverages this reciprocity to ensure that complex scenes with arbitrary and unknown reflectance can be reconstructed from reciprocal pairs of images. For a pair of images to be considered reciprocal, the second image has to be captured after the light and camera positions have been swapped. Although, significant progress has been made over the years to improve this method, considerable efforts are used to set up the scene for Helmholtz Stereopsis.

Fairly recently, deep learning has become the popular choice of method for solving Computer vision problems, due in part to the increase in the availability of large datasets and to the ability of deep neural networks to learn the non-linear relationship between two or more entities. 3D reconstruction is no exception. Deep learning has been integrated in the traditional methods of 3D reconstruction, which has led to promising results [60, 68].

1.2 Objectives

To the best of my knowledge, there is no deep learning framework used for Helmholtz Stereopsis. This project aims to provide deep learning architectures that can be used estimate the depths and surface normals of complex scenes, respectively. The estimated depth maps or normal maps of the scenes can be used to to reconstruct the scenes, which will ease the set-up process required to reconstruct complex scenes from reciprocal pairs of images. The objectives for the project are the following:

- To generate a dataset that is suitable for training a deep, Helmholtz-based neural network
- To adapt a deep, multi-view stereo network for deep Helmholtz Stereopsis
- To adapt a deep, photometric stereo network for deep Helmholtz Stereopsis
  - To refine the adapted networks towards reconstructing complex 3D scenes using Helmholtz Stereopsis
  - To evaluate the performance of the networks on different configurations settings
- To write a dissertation report on the project

1.3 Achievements

I have fulfilled the objectives of the project. I generated a Helmholtz dataset and evaluated it to ensure it is suitable for deep Helmholtz Stereopsis. I adapted a deep multi-view stereo network for deep Helmholtz Stereopsis and used it to estimate the depth maps of a variety of scenes. The adapted network performed well in inferring the depths of simple scenes and produced decent results for fairly-complex scenes. Furthermore, it can handle some occluded areas of a model, although some depth information may be lost in the process, in other regions of the model. Furthermore, I designed different configurations of the adapted network and compared their performance. I showed that certain configurations of the adapted network can improve its performance. Lastly, I adapted a deep photometric stereo network for deep Helmholtz Stereopsis and used it to estimate the surface normals of a variety of scenes. The adapted network produced decent results on simple and fairly-complex scenes. However, as the scenes get more complex, the adapted network struggled to recover the surface normal information of the scenes. I provided possible reasons for this. I also designed different configurations of the adapted network and

compared their performance. I showed that certain configurations of the adapted network can improve its performance.

1.4 Overview of Dissertation

Chapter 2 describes the existing methods of 3D reconstruction and justifies the need for a deep Helmholtz Network. Furthermore, it explores some of the datasets used in deep 3D reconstruction. Chapter 3 describes the rationale behind choosing a dataset that is suitable for deep Helmholtz Stereopsis. Furthermore, it describes the process of generating a Helmholtz dataset from the chosen dataset. The chapter ends with an evaluation of the generated Helmholtz dataset to ensure it is suitable for Helmholtz Stereopsis. Chapter 4 describes a deep multi-view stereo network and a deep photometric stereo network, and how they have been adapted for deep Helmholtz Stereopsis. Chapter 5 discusses the results of the adapted networks on models with varying complexity. Furthermore, it discusses the performance of the networks on different configuration settings. Chapter 6 summarizes the contents of this paper. Furthermore, it talks about the work that has been carried out so far and how this project can be extended to future work.

2 BACKGROUND THEORY AND LITERATURE REVIEW

Due to the high demand of 3D reconstruction, significant progress has been made in developing methods that can reconstruct scenes that are accurate and complete. Section 2.1 describes the classical methods and their limitations. It also describes how Helmholtz Stereopsis can be used to overcome the limitations. Section 2.2 describes deep learning methods that have been developed from the classical methods. Section 2.3 describes some of the datasets that have been used to train deep stereo methods. Finally, section 2.4 summarizes the chapter and justifies the need for a deep learning network for Helmholtz Stereopsis.

2.1 3D Reconstruction Methods

Existing methods of 3D reconstruction can be broadly divided into multi-view stereo, photometric stereo and Shape from Silhouette (SfS) [6].

As the name suggests, multi-view stereo methods are a class of methods that reconstruct an object from multiple views of the object [55]. These include feature-based methods [37, 11, 47, 59, 12], which use feature correspondence across views of an object to produce depth maps for each view. Accurate feature matches are necessary to reconstruct good 3D models. Specularities in the views may lead to erroneous feature matching. Matching features may be incorrectly classified as not matching, as a result of the difference in the specular response of the matching features. Thus, the method is limited to Lambertian surfaces, which are diffuse-only surfaces. Non-Lambertian multi-view methods [28, 29, 45] have been proposed to handle non-Lambertian surfaces, but they are limited in terms of their resolution and accuracy.

Photometric stereo methods [62, 5, 8, 23, 24, 56, 46, 58] recover the surface normals of a scene by varying the direction of illumination on the scene. Although, it is able to reconstruct scenes with non-Lambertian surfaces, the reflectance has to be known beforehand. The reflectance is used in combination with the illumination to generate the surface normals, which is then used to reconstruct the 3D surface. In order to recover the surface normals of a scene with unknown reflectance, assumptions are made about the reflectance. Methods such as [43, 41, 53, 44, 38] separate the specular term and the diffuse term from the radiance of the scene in order to estimate the reflectance model. Other methods such as [21, 7, 19, 57, 26] estimate the reflectance of

parametric models. Although these methods are able to estimate a broad range of reflectance properties, they may be unreliable in non-generic reflectance, such as anisotropic reflectance.

Shape from Silhouette methods [9, 48, 40, 35, 36] reconstruct the shape of an object by back-projecting its silhouette images. As the silhouettes are back-projected, they intersect with one another. This intersection forms the visual hull [34], which is the shape of the object. Although, SfS methods are independent of the reflectance properties and are simple to set up, they struggle to reconstruct concavities from the silhouettes.

Zickler et al. [69] proposed a method called Helmholtz Stereopsis. This method reconstructs unknown and spatially-varying scenes by exploiting Helmholtz reciprocity, which states that for a reciprocal pair of images, the ratio of the emitted radiance to the incident radiance is the same for both images. Considerable progress has been made in improving the Helmholtz Stereopsis method. [51] proposed a Maximum a Posteriori (MAP) formulation to optimize the normal and depth information generated by the standard Helmholtz Stereopsis. In a subsequent paper [50], the MAP formulation is used in the reconstruction of dynamic scenes. Although the use of the formulation showed promising results, they were limited to reconstructing 2.5D scenes. [6] addressed this limitation by proposing a Markov Random Field (MRF) optimization framework to enable the reconstruction of full 3D scenes. Despite the progress made, implementations of HS still has a cumbersome capture set-up to generate the reciprocal pairs of images.

2.2 Application of Deep Neural Networks in 3D Reconstruction

Deep learning has seen a massive growth in popularity since Alexnet [33] was introduced. It has been used extensively in recent years for 3D reconstruction. Deep neural networks have been used to improve traditional multiview stereo. [18] used Convolutional Neural Networks (CNNs) to reconstruct 3D models from a single or multiple views of the model. The CNN architecture used is a recurrent neural network, which allows the architecture to adaptively learn the shape of the object by recovering the features that were lost at earlier layers. The main advantage of the network is that it is able to reconstruct the shape of a model from a single or multiple views of the model with large baselines. Another advantage is that the training process requires minimal supervision (only the the bounding boxes are required). Its main limitation is that it can only generate low resolution voxel grids of models. This limitation was addressed by [42]. A novel representation, called Occupancy Networks was introduced to reduce the memory footprint of the 3D geometry. It is a residual neural network (ResNet) which learns the mapping from every continuous 3D points

(that can be observed in the input) to occupancy values. These occupancy values are then used to progressively extract the surface of the input at a high resolution. Another advantage of this network is that it can take in point clouds or voxel grids as inputs to the network. [66] proposed a deep neural network called MVS-Net to infer the depth of a scene. This inference is based on the plane-sweep algorithm [20], which estimates the depth from a set of depth hypotheses. [17] modifies this architecture into a new architecture called UCS-Net, such that the plane-cost volume with constant depth hypotheses is replaced with relatively-thin cost volumes with spatially-varying depth hypotheses, which will be used to progressively regress the depth in a course-to-fine fashion. This improves the accuracy and completeness of the MVS-Net while using a much lower memory footprint.

Deep neural networks have also been used to improve traditional photometric stereo. [52] introduced a deep fully-connected network that regresses the surface normal per pixel, by feeding a normalized vector into the network, where each element in the per-pixel vector represents the observed intensity under a specific light direction. The limitation of this network is that pre-defined light directions have to be the same for both training and testing, which limits its generalization to different lighting conditions. CNN-PS [25] and PS-FCN [14] are two examples of networks that have been proposed to address this limitation. CNN-PS is a CNN that regresses the surface normal per pixel by feeding an observation map into the network. An observation map is a novel idea proposed in [25] to encode observed intensities at their corresponding 2D light direction coordinates on the map. The map coordinates itself is produced by projecting a unit hemisphere (which spans light directions) onto a plane. The observation map ensures that the network can handle an arbitrary number and order of input images. The limitation of this network is that, as a result of predicting the normal map in a per-pixel manner, it does not directly consider the global information of the scene. Thus, the performance of the network is affected when the scene contains global illumination effects, such as cast shadows and inter-reflections [68]. PS-FCN [14] is a CNN that regresses the normal map all at once by feeding a set of input images and their light directions as inputs to the network. It uses the max-pooling operation to aggregate the features of the input images. This allows the network to be able to handle an arbitrary number and order of input images and light directions. Furthermore, it uses local neighbourhood information to regress the normal map at once, which makes it less susceptible to global illumination effects. However, PS-FCN performs worse when the surface material of the model is spatially-varying. [16] addresses this limitation by concatenating the input images and

normalizing the co-located intensities to unit vectors. In terms of handling uncalibrated lighting, [15] proposed a network, called SDPS-Net, which consists of a Light Calibration Network (LC-Net) and a Normal Estimation Network (NENet). LC-Net estimates the light directions that corresponds to the input images. The estimated light directions and input images are fed as input to the NENet (which uses the same architecture as PS-FCN) to estimate the normal map. Similarly to PS-FCN, SDPS-Net performs worse when the BRDF of the material is spatially-varying.

2.3 Datasets used in Deep 3D Reconstruction

There are various publicly available datasets that have been used for deep 3D reconstruction. This section presents some of the datasets that have been created for deep multi-view stereo and deep photometric stereo. It also presents some other datasets that were originally created for research in deep computer vision problems, and not necessarily for 3D reconstruction but can be used for deep 3D reconstruction.

2.3.1 Datasets used for Deep Multi-view Stereo

DTU

The DTU dataset [27] (figure 2.1) consists of 124 different scenes, where 59 of the scenes have been captured under 49 camera positions and 21 of the scenes have been captured under 64 camera positions. The remaining 44 scenes were captured under 4 camera positions. Each scene is placed in a controlled environment and captured using a structured light scanner, whose position is controlled by a fixed robotic arm. For all camera positions in each scene, 7 images were generated under different lighting conditions. The resolution of the images is 1600 x 1200. The dataset also includes the point clouds of the scenes, along with its surface reconstruction. The diversity of the scenes, in terms of their reflectance, texture and geometric properties makes the dataset a popular choice for training and evaluating multi-view stereo networks. However, the fixed nature of the robotic arm limits the trajectories that the camera can explore, which may affect its generalization to real-world and unseen data.

Figure 2.1: Sample images from the DTU dataset.

ETH3D

The ETH3D multi-view stereo dataset [54] (figure 2.2) consists of different indoor and outdoor scenes that have been captured using a high-resolution DSLR camera. The scenes are split into 13 training scenes and 12 testing scenes. The ground truth geometry was generated using a high-precision laser scanner. Due to its relatively small size, they are better suited as an evaluation benchmark.

Figure 2.2: Sample images from the ETH3D dataset.

BlendedMVS

BlendedMVS dataset [67] is a large-scale synthetic dataset that consists of 113 scenes, where each scene consists of 20 to 1,000 images for a total of over 170,000 images in the dataset. The scenes are carefully-selected reconstructions of mesh models of input images that have been provided to the 3D reconstruction software in Altizure [1]. The reconstructed mesh models were rendered from different camera positions to generate the scene images and their corresponding depth maps. The rendered images were blended with their corresponding input images to recover the lighting from the input images. The cameras that are used to render the images are not restricted like in the DTU capturing process. Thus, they are able to explore a variety of camera trajectories (figure 2.3), which makes the dataset more generalizable to unseen data.

Figure 2.3: Sample images from the BlendedMVS. Extracted from [67].

2.3.2 Datasets used for Deep Photometric Stereo

Blobby dataset

The Blobby dataset [30] (figure 2.4) has seen wide use in deep photometric stereo. The Blobby dataset consists of 10 blob-like models and its associated normal maps, where each model can be rendered in its own lighting environment. Due to its small size and low complexity, the dataset is often used for pre-training a photometric stereo network before the network is trained with a more complex dataset, such as the Sculpture dataset.

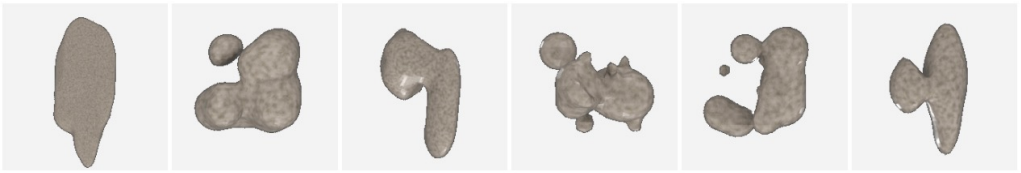

Figure 2.4: Sample images from the Blobby Dataset.

Sculpture dataset

The Sculpture dataset [61] (figure 2.5) consists of 307 sculptures that have been extracted from Sketchfab [4].

Figure 2.5: Sample images from the Sculpture Dataset.

To simulate a diverse range of surface reflectance, the objects in the Blobby and Sculpture datasets are often rendered using surface materials from a BRDF dataset, such as the MERL dataset [39]. The MERL dataset consists of 100 unique BRDFs of real-world materials. Although, the objects in the datasets display a diverse range of reflectance properties, they belong to only a few object categories.

2.3.3 Other Datasets

ShapeNet

The ShapeNet dataset [13] consists of over 300M CAD models that are distributed into more than 3,135 categories [22]. The models in the dataset were collected to facilitate research in computer graphics and vision. In computer vision, it is mostly used for 3D reconstruction and classification problems. The publicly available subsets of ShapeNet are the ShapeNetCore and the ShapeNetSem. ShapeNetCore contains more than 51,000 models that have been classified into 55 categories while ShapeNetSem consists of 12,490 models that have been classified into more than 270 categories. ShapeNetSem contains more varied and more complex models than the ShapeNetCore.

ModelNet40

The ModelNet40 dataset [63] consists of over 12,000 CAD models that have been distributed into 40 categories. It has been used in various computer vision problems such as 3D point cloud

classification, 3D object recognition and 3D reconstruction. As a result of ModelNet and ShapeNet datsets being a collection of CAD models from 3D models websites, ModelNet and ShapeNet share some of their 3D models, which is something to keep in mind if one is trying to train on one dataset and test on the other.

ObjectNet3D

ObjectNet3D dataset [65] consists of over 44,000 CAD models that have been distributed into 100 categories. It also includes over 90,000 images, with more than 200,000 objects in the images. Although, the dataset was created for 3D object recognition and 3D object pose estimation, it can be adapted for use in 3D reconstruction.

2.4 Summary

This chapter introduced the existing methods in 3D Reconstruction and its limitations. It explained how Helmholtz Stereopsis overcomes the limitations. Furthermore, it described some of the applications of deep neural networks in 3D reconstruction, as well as the datasets that are used or can be used to train such networks. Despite the prevalence of deep learning and its applications in 3D reconstruction, there is no deep learning framework for Helmholtz Stereopsis, which will ease the set-up process required to generate reciprocal pairs of images. To this purpose, the UCS-Net and PS-FCN architectures were adapted to deep neural networks that can be used for deep Helmholtz Stereopsis.

3 GENERATING A HELMHOLTZ DATASET

In order to generate a Helmholtz dataset from a pre-existing dataset, each scene in the relevant dataset should be rendered in a Helmholtz environment, where lighting can be controlled. Thus, a dataset that consists of a diverse set of 3D models of scenes (with spatially-varying reflectance), and their corresponding surface materials will be ideal for generating a Helmholtz dataset. Although, there are several datasets that pass these requirements, such as ModelNet40, ObjectNet3D and PASCAL3D+[64], the ShapeNetSem dataset was selected as the basis for generating a Helmholtz dataset, due to its variety and relative complexity. Although, ShapeNetSem does not include every possible reflectance, it includes a sufficient number of spatially-varying reflectance produced by the surfaces of everyday objects. Figure 3.1 shows the material composition of some of the categories of models in the dataset.

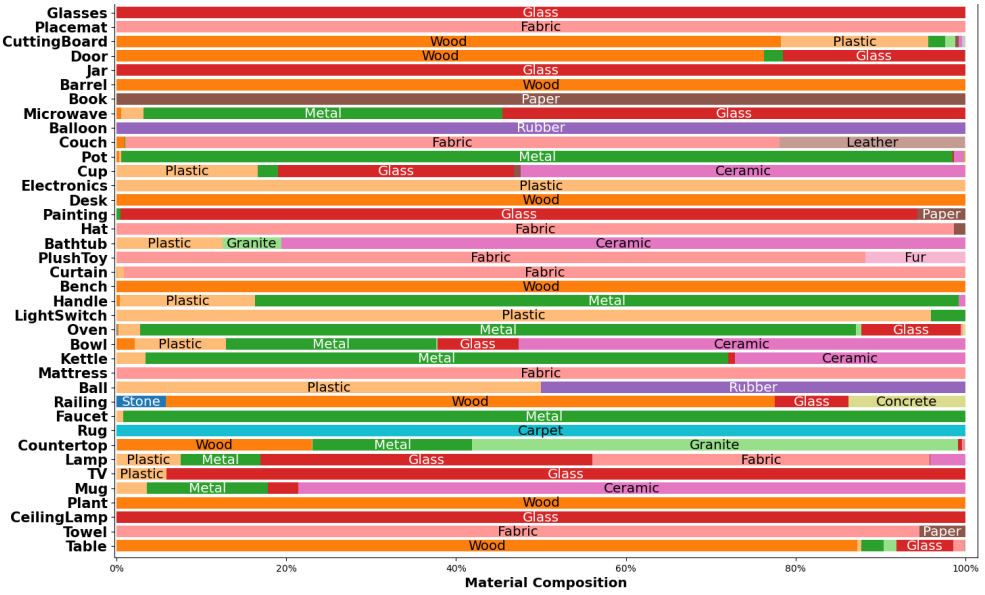

Figure 3.1: Material composition of some of the categories of models in the ShapeNetSem dataset, which were extracted from [10].

3.1 Generating Views Images and their Associated Reciprocal Pairs of Images

To generate realistic models, the models in the Helmholtz dataset were rendered using the Cycles rendering engine, which was developed by Blender [2]. This process was facilitated with the use of the Blender Python API. The pre-rendering process involved converting the models in the ShapeNetSem dataset from the OBJ file format to the Graphics Language Transmission Format

(gLTF) to circumvent the issue of missing triangles and textures that occurred when the models in OBJ format were rendered via Blender. Figure 3.2) shows some of the objects in the Helmholtz dataset.

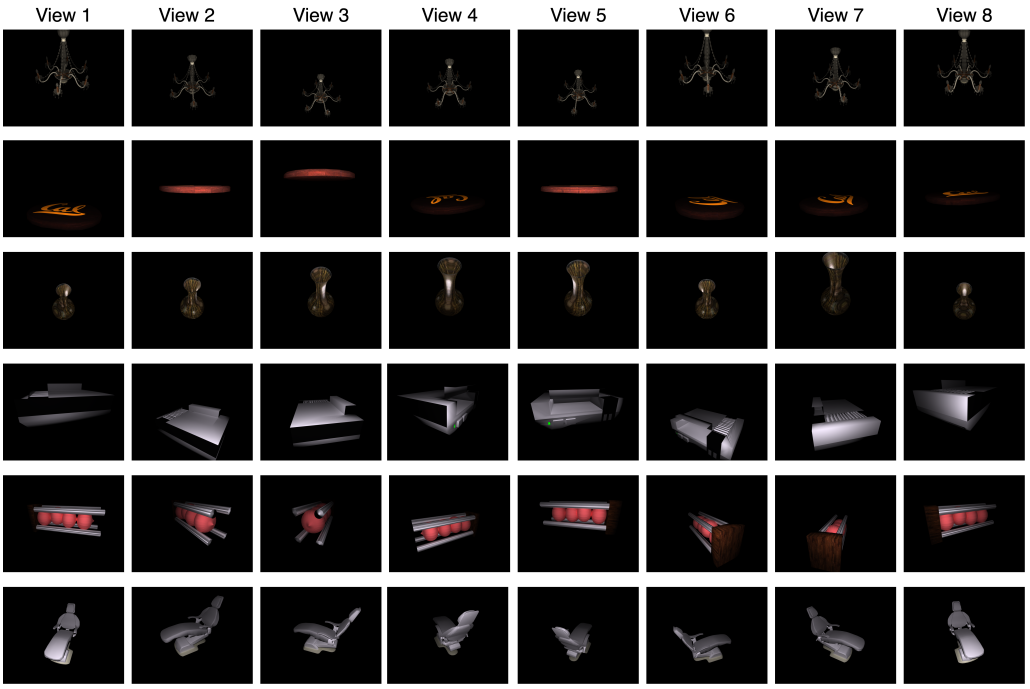

Figure 3.2: View Images of some of the models in the Helmholtz dataset.

Due to limited resources, 4,676 models of the ShapeNetSem dataset were used for the generation of the Helmholtz dataset. For each of the chosen models, 8 views were captured in an environment where the only light source is the reference light L_0, and it is co-located at the reference camera position C_0 (figure 3.3a). Each view $\{\mathrm{V}_i\}_{i=1}^{8}$ is parametrized by an elevation angle, θ_{V_i} and an azimuth angle, ϕ_{V_i}. The elevation angle is randomly chosen from $[-r_{\theta_{\mathrm{V}_i}}, r_{\theta_{\mathrm{V}_i}}]$, where $r_{\theta_{\mathrm{V}_i}}$ is a small radius that is determined by the size of the model. The starting azimuth angle, ϕ_{V_0} is randomly chosen, and each subsequent azimuth angle, ϕ_{V_i} is randomly chosen from $[\phi_{\mathrm{V}_{i-1}} + \frac{\pi}{4} - r, \phi_{\mathrm{V}_{i-1}} + \frac{\pi}{4} + r]$, where r is a small radius. When the views are fused together, it produces a 360° view of the model.

For each view of the model, 4 reciprocal pairs of images are captured. To set up the capture of a reciprocal pair, a camera $\{\mathrm{C}_i\}_{i=1}^{4}$ and a light source $\{\mathrm{L}_i\}_{i=1}^{4}$ are randomly rotated on an invisible wheel (figure 3.3b), where the reference camera and reference light are at the center of this wheel. After capturing the first image of the reciprocal pair, the positions of the light source

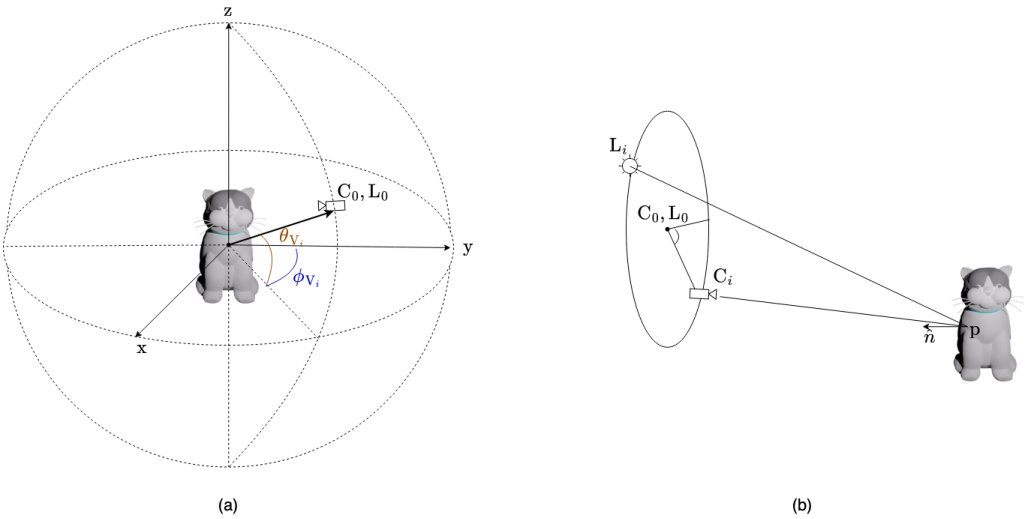

Figure 3.3: (a) Set-up for a view capture. (b) Set-up for a reciprocal pair of images capture.

and the camera are swapped before the second image is captured. Figure 3.4 shows a view of a model in the Helmholtz dataset and its corresponding reciprocal pairs.

The depth and normal maps for each view of the models in the dataset are also generated and serve as the ground truths of the model. Furthermore, the camera parameters and the light source location are saved in order to be used by the multi-view and photometric stereo networks to predict the depth and normal map for each view, respectively. Lastly, a segmentation mask for each view was also generated. It is used by the networks to learn only the relevant information from the ground truth maps. All images were saved as 32-bit floating-point EXR images.

3.2 Evaluating the Helmholtz Dataset

The Helmholtz dataset is evaluated by seeing how well it obeys Helmholtz reciprocity, as well as its response to varying light intensity.

3.2.1 Helmholtz Reciprocity

For a reciprocal pair of images (figure 3.5), let p be a point on the surface, with its corresponding unit normal vector, $\hat{n}$. Let o_l and o_r be the positions of the camera and light source, respectively. The unit vectors, $\hat{v}_l$ and $\hat{v}_r$ are the directions from point p to the positions of the camera and light source, respectively. Helmholtz reciprocity ensures that reciprocal pair of images is constrained by the following equation [69]:

$$\left(i_l \frac{\hat{v}_l}{|o_l - p|^2} - i_r \frac{\hat{v}_r}{|o_r - p|^2}\right) \cdot \hat{n} = w(d) \cdot \hat{n} = 0. \tag{3.1}$$

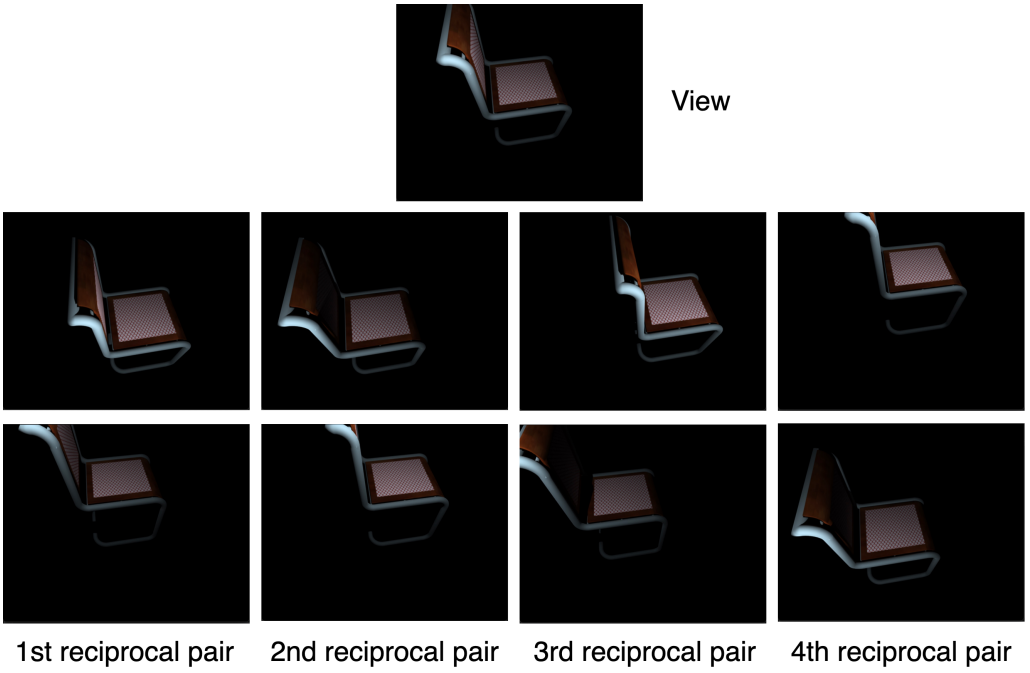

Figure 3.4: A view of a model in the Helmholtz dataset and its corresponding reciprocal pairs.

Where d is the depth at point p. i_l and i_r are the light intensities of point p, as seen in the left and right reciprocal images, respectively.

The equation says that the $\mathrm{w(d)}$ vector and $\hat{\mathrm{n}}$ are perpendicular to each other. To ensure that a captured reciprocal pair of images adheres to the reciprocity, it is sufficient to show that for a sufficient number of points, the angle between $\mathrm{w(d)}$ and $\hat{\mathrm{n}}$ is close to 90 degrees. The following is the algorithm used to test the reciprocity of a reciprocal pair of images that was captured in a Helmholtz environment:

- An object model from the ShapeNet dataset is rendered in a Helmholtz setting. The reciprocal pairs of images are saved in the same fashion as in the generation of the Helmholtz dataset
- A reciprocal pair of images is selected and gamma-corrected
- The 3D vertex points of the object model that are visible in both the left and right reciprocal images are extracted into a list. The normals of each of these points are also rendered onto a normal map.
- A masking function is applied to the reciprocal pair of images and the normal map to extract only the relevant points in the images.

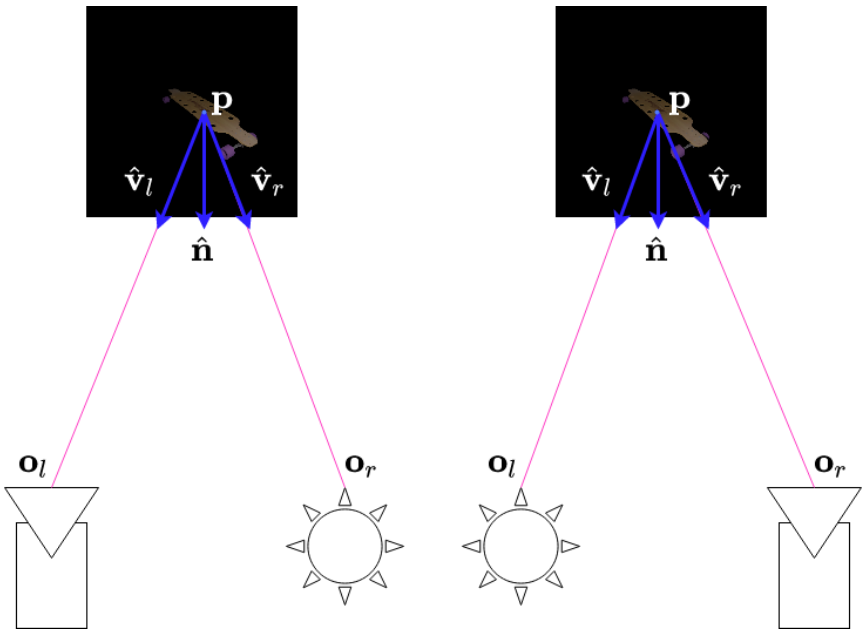

Figure 3.5: Reciprocal pair of images captured by flipping the positions of the camera and light source.

- For each representative 3D point, the camera parameters of a reciprocal pair are used to calculate the function in equation 3.1. The result for each point are stored in a list
- The average of the values in the list shows how well the reciprocal pair obeys Helmholtz reciprocity

The skateboard model was chosen to test the reciprocity (figure 3.6). The number of sampled points in the left reciprocal image was 1,518, and the number of sampled points in the right reciprocal image was 1,616. The number of visible points in the intersecting set that was used to calculate the angle between w(d) and $\hat{\mathrm{n}}$ was 1,382. The average angle was 90.0615 degrees. This value is close to 90 degrees, which means the reciprocal pair adheres to the Helmholtz reciprocity. Although, only a single reciprocal pair in the dataset was evaluated, the result provides a good indication that the other reciprocal pairs in the dataset will adhere to the reciprocity because they were all captured in the same controlled environment.

3.2.2 Proportionality

The Helmholtz dataset can also be evaluated by the proportionality between the light intensity and the radiance of the scene. As the light intensity changes, the scene radiance should change

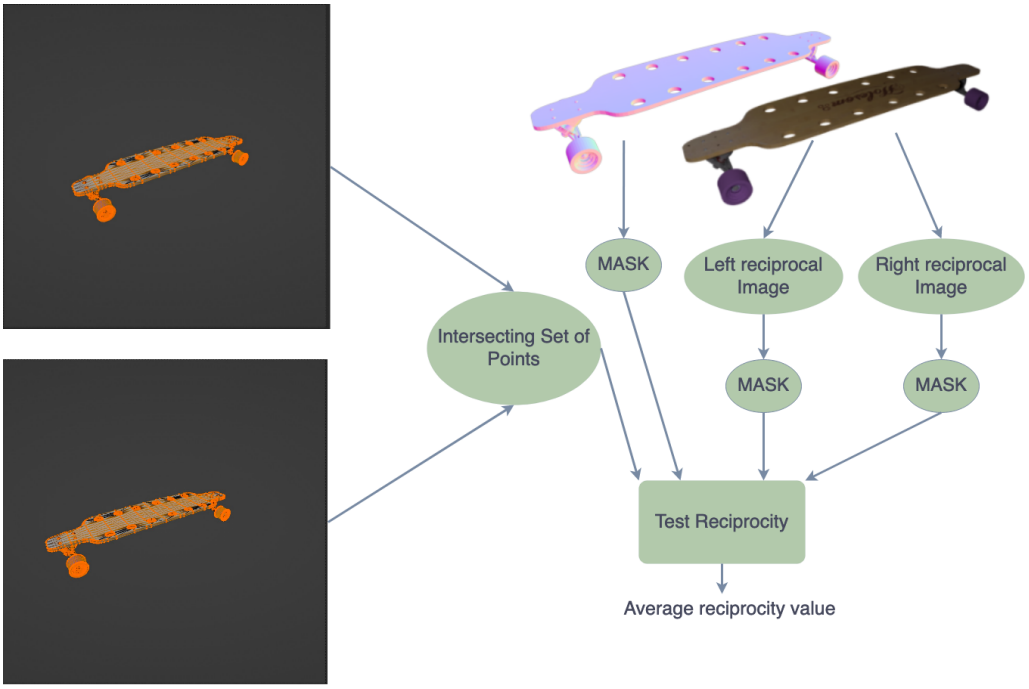

Figure 3.6: Process of calculating the reciprocity value for a reciprocal pair of skateboard images.

proportionally as well. The Helmholtz dataset obeys this proportionality. Figure 3.7 shows an example of this.

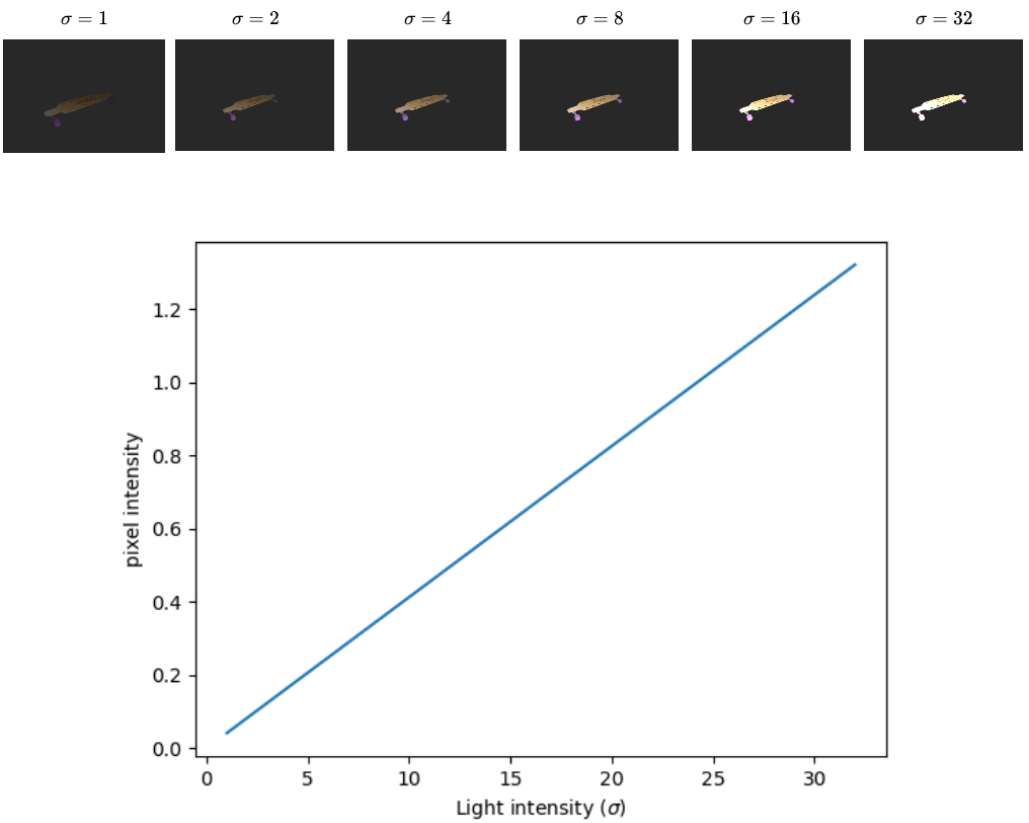

Figure 3.7: As the light intensity increases, the pixel intensity increases proportionally.

4 DEEP HELMHOLTZ STEREOPSIS

This chapter describes the UCS-Net and PS-FCN architectures, and how they were adapted to be suitable for deep Helmholtz Stereopsis. Section 4.1 discusses the pipeline of the UCS-Net architecture. Section 4.2 discusses how UCS-Net was adapted for deep Helmholtz Stereopsis in order to estimate the depth map of a scene. Section 4.3 discusses the pipeline of the PS-FCN architecture. Finally, section 4.4 discusses how the PS-FCN was adapted for deep Helmholtz Stereopsis in order to estimate the normal map of a scene.

4.1 UCS-Net

UCS-Net (figure 4.1) predicts the per-view depth map at different scales, where each scale corresponds to a particular stage. At the first stage, the network predicts the depth map from a cost volume of planes, where each plane corresponds to a fixed depth hypothesis. At subsequent stages, the sample space for the depths are refined from the previous stage, such that the depth hypotheses for the cost volume are spatially-varying. The predicted depth map is also scaled up as it progresses through the stages. The following subsections go into detail on how the network generates a depth map for a single view.

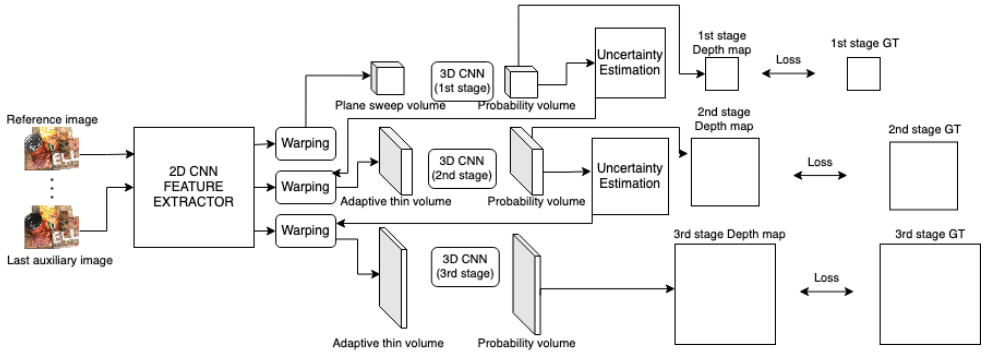

Figure 4.1: UCS-Net architecture.

4.1.1 Auxiliary Image Selection

N images, which consist of a reference image I_1 and $N-1$ auxiliary images $\{\mathrm{I}_i\}_{i=2}^{N}$, are fed as input into the network. Each input image has a size of $W \times H$, where W is the width of the image and H is the height. The auxiliary images are selected according to a global view selection score. Each image pair is assigned a view selection score, such that the image pairs with the strongest

visibility association will have the highest score. Thus, the selected auxiliary images will be the images that have the strongest visibility association with the reference image. Let i be a view image and j be another view image, the view selection score, $\mathrm{s}(i, j)$ is calculated according to [66] as:

$$\mathrm{s}(i,j) = \sum_{\mathbf{p}} \mathcal{G}(\theta_{ij}(\mathbf{p}) \tag{4.1}$$

$$\theta_{ij}(\mathbf{p}) = (\frac{180}{\pi}) \arccos((\mathbf{c}_i - \mathbf{p}) \cdot (\mathbf{c}_j - \mathbf{p})) \tag{4.2}$$

$$\mathcal{G}(\theta) = \begin{cases} \exp(-\frac{(\theta-\theta_0)^2}{2\sigma_1^2}), & \theta \leq \theta_0 \\ \exp(-\frac{(\theta-\theta_0)^2}{2\sigma_2^2}), & \theta > \theta_0, \end{cases} \tag{4.3}$$

where **p** is an element of a sparse set of points, where each point in the sparse set is visible in both view images i and j, $\theta_{ij}(\mathbf{p})$ is the baseline angle of **p** and **c** is the camera centre. θ_0, σ_1 and σ_2 were set to 5, 1 and 10 respectively.

4.1.2 Extracting Feature Maps

The first part of the network is a 2D UNet [49]. It is an encoder-decoder network (figure 4.2) with skip connections, that extracts the features of the input images at different scales. Each input image I_i uses the same network and share the same weights. The encoder has eight convolutional layers, where each convolutional layer is followed by a batch normalisation (BN) layer and a Rectified Linear Unit (ReLU) layer. Using a stride of 2, the third and the sixth layers of the encoder down-samples their input by a factor of 2,which enables the decoder to generate high-level features at three different scales. The decoder consists of seven convolutional layers. Out of the convolutional layers, only the deconvolutional layers are followed by a BN and ReLU layers. The up-sampled outputs of the second and fifth layers of the decoder are concatenated with the outputs of the fifth and second layer of the encoder network, respectively to recover important features that had been lost. The final outputs of the decoder are produced by its first, fourth and final layers, where the outputs are a 32-channel feature map, $\mathrm{F}_{i,1}$ with a size of $\frac{W}{4} \times \frac{H}{4}$, a 16-channel feature map, $\mathrm{F}_{i,2}$ with a size of $\frac{W}{2} \times \frac{H}{2}$, and an 8-channel feature map, $\mathrm{F}_{i,3}$ with a size of $W \times H$ respectively.

The feature maps are passed to their respective stages to generate depth maps at the same resolution as their corresponding feature map, with each subsequent stage generating a more refined depth map.

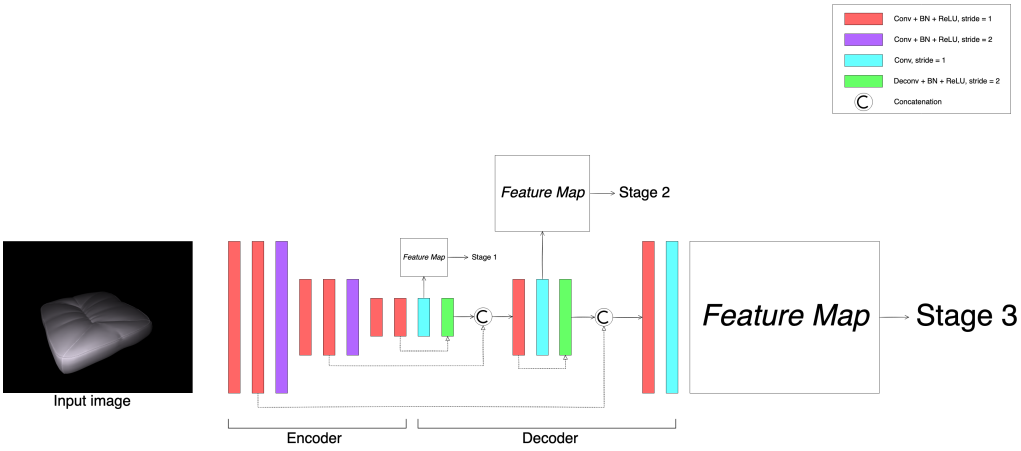

Figure 4.2: A 2D UNet that extracts feature maps at three different scales from an input image

4.1.3 Stage 1: Depth Map Prediction

Generating the Cost Volume

At the first stage, each auxiliary feature map is warped with respect to the reference image to ensure that the feature maps are fronto-parallel. The warping is performed for each depth hypothesis, d. At this stage, a uniform set of depth hypotheses (sampled from a predefined depth range) is used for the warping process. The homography, $\mathrm{H}_i(d)$ that warps an auxiliary image coordinate $(x_i d, y_i d, d)$ to its new image coordinate $(x', y', 1)$ to produce the fronto-parallelism is given as:

$$\mathrm{H}_i(d) = \mathrm{K}_i \mathrm{T}_i \mathrm{T}_1^{-1} \mathrm{K}_1^{-1}, \tag{4.4}$$

where K_i and T_i contain the intrinsic and extrinsic parameters of input image I_i, respectively.

The new image coordinates are used to sample pixels from the auxiliary feature maps to generate a feature volume for each auxiliary feature map. Each channel of the feature volume consists of some number of planes, where each plane represents a depth hypothesis. The number of planes corresponds to the number of depth hypotheses. The reference feature map is also converted to a feature volume.

The cost volume C at a stage, is generated using the variance of the feature volumes, V_i:

$$\mathrm{C} = \frac{\sum_{i=1}^{N} (\mathrm{V}_i - \overline{\mathrm{V}})^2}{N}, \tag{4.5}$$

where $\overline{\mathrm{V}}$ is the average volume of all the feature volumes, and all the operations used to calculate the variance are element-wise operations.

Regularizing the Cost Volume and Generating Probaility Map

The cost volume is regularized by feeding it through it through a 3D UNet architecture (figure 4.3). The architecture has eleven 3D convolutional layers, where each convolutional layer, except the final layer, is followed by a BN layer and a ReLU layer. The encoder consists of the first seven 3D convolutional layers. Using a stride of 2, the second, fourth and fifth layers of the encoder down-samples their input by a factor of 2. The decoder has four layers, where the up-sampled outputs of the first three layers of the decoder are added to the outputs of the first, third and fifth layers of the encoder, respectively to recover lost information. The output of the last layer of the network is a one-dimensional channel probability volume, where each plane in the volume represents a probability map that corresponds to a depth hypothesis.

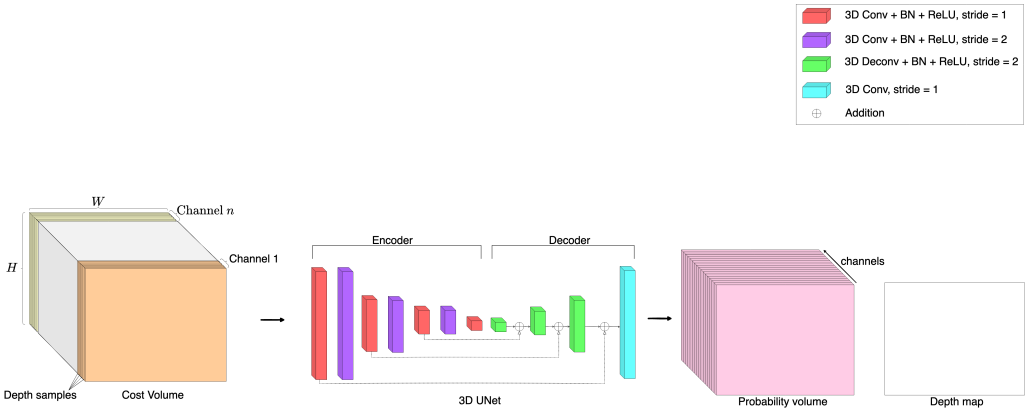

Figure 4.3: A 3D UNet is used to regularize the cost volume to generate a probability volume. Softmax is applied along the channels on the probability volume to generate the initial depth map for the current stage.

Estimating Initial Depth Map

The initial depth map is generated by applying softmax along the channels of the probability volume to normalize the depth probabilities. The depth map is estimated by calculating the depth value for each pixel in the depth map as the expected depth value of the probability distribution created from the values of the collocated pixels in the probability volume:

$$\hat{\mathrm{L}}(x) = \sum_{j=1}^{D} \mathrm{L}_j(x) \cdot \mathrm{P}_j(x), \tag{4.6}$$

where D is the number of planes in the probability volume, $\mathrm{L}_j(x)$ is the depth hypothesis at pixel x of the j-th plane, P_j is the probability that the depth at pixel x is $\mathrm{L}_j(x)$, and $\hat{\mathrm{L}}(x)$ is the expected depth value at pixel x.

The generated depth map is in the same resolution as its corresponding feature map.

4.1.4 Subsequent Stages: Depth Map Prediction

Subsequent stages use the same pipeline as stage 1, except that the depth hypotheses are sampled according to the per-pixel uncertainty of the predicted depth map of the previous stage. The variance of the planes in the probability volume is computed as:

$$\hat{\mathrm{V}}_k(x) = \sum_{j=1}^{\mathrm{D}_k} \mathrm{P}_{k,j}(x) \cdot (\mathrm{L}_{k,j}(x) - \hat{\mathrm{L}}_k(x))^2, \tag{4.7}$$

where D_k is the number of planes in the probability volume at stage k, $\mathrm{L}_{k,j}(x)$ is the depth hypothesis at pixel x of the j-th plane at stage k, $\mathrm{P}_{k,j}(x)$ is the probability that the depth at pixel x is $\mathrm{L}_{k,j}(x)$, $\hat{\mathrm{L}}_k(x)$ is the expected depth value at pixel x and stage k, and $\hat{\mathrm{V}}_k(x)$ is the variance of the probability distribution at pixel x and stage k.

Depth Uncertainty at stage k is defined as the confidence interval, $\mathrm{C}_k(x)$, which is calculated from the expected depth value, $\hat{\mathrm{L}}_k(x)$ and variance, $\hat{\mathrm{V}}_k(x)$ at pixel x and stage k:

$$\mathrm{C}_k(x) = [\hat{\mathrm{L}}_k(x) - \lambda\hat{\sigma}_k(x), \hat{\mathrm{L}}_k(x) + \lambda\hat{\sigma}_k(x)], \tag{4.8}$$

where $\hat{\sigma}_k(x) = \sqrt{\hat{\mathrm{V}}_k(x)}$ and λ is a scalar value that determines the size of the confidence interval. Each pixel x at stage $k+1$ uniformly samples D_{k+1} depth values from the confidence interval, $\mathrm{C}_k(x)$.

This ensures that each of the subsequent stages uses spatially-varying depth hypotheses from the confidence interval of their previous stage (figure 4.4). The confidence interval decreases as we progress through the stages, which makes the network more confident and more accurate in its depth estimation.

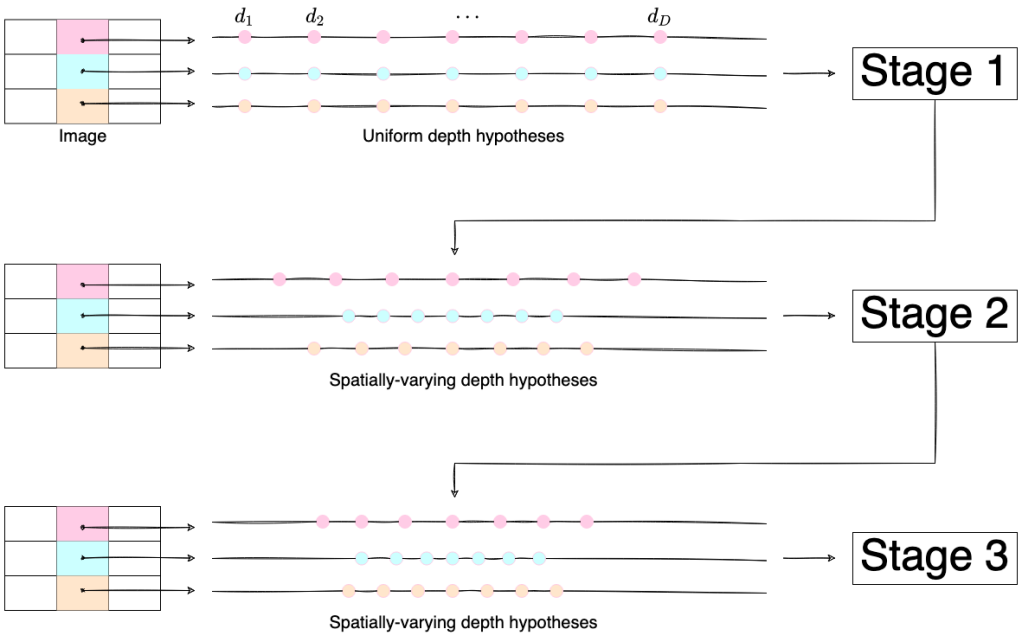

Figure 4.4: The first stage uses a uniform depth hypotheses that have been sampled from a pre-defined depth range. Subsequent stages use spatially-varying depth hypotheses that have been sampled from a confidence-interval produced by their previous stage. The confidence interval becomes more refined as we progress through the stages.

4.1.5 Loss

For each resolution, the mean absolute error is calculated between the estimated depth map and the ground truth, which has been scaled to the same resolution of the relevant estimated depth map, if necessary. The total loss of the network is the linear combination of all three losses:

$$\mathrm{L}_{\text{depth}} = \alpha \mathrm{L}_1 + \beta \mathrm{L}_2 + \gamma \mathrm{L}_3, \tag{4.9}$$

where α, β and γ are the given weighting terms.

4.2 Adapting UCS-Net for Deep Helmholtz Stereopsis

Rather than selecting auxiliary views according to a global selection score, a single reciprocal pair of the reference image is used as the auxiliary images. The reciprocal pair for the reference image will naturally have the strongest visibility association with the reference image in the dataset, so there is no need to calculate a score. This allows the reciprocal pair to be smoothly integrated as auxiliary images.

A single stage of UCS-Net is used for deep Helmholtz Stereopsis (figure 4.5). Each input image is concatenated with their corresponding light source position and fed into the feature extractor network. This enables the network to better leverage Helmholtz reciprocity to infer the depth of a scene. The refine network in MVS-Net is adapted for use in the UCS-Net architecture to refine the estimated depth map of the first stage of UCS-Net. The refine network is used to recover features that may have been lost due to the over-smoothing nature of the cost volume regularizer. To generate the input for the network, the depth map is normalized to range [0, 1] to prevent the network from being biased at a particular scale. Next, the reference image is scaled down to the resolution of the initial depth map and concatenated with the initial depth map. The result of the concatenation is fed as input to the refine network. The network consists of four convolutional layers. Only the first three layers is followed by a batch normalization layer and a ReLU layer. Each of them output a 32-channel feature map. The last layer outputs a 1-channel depth residual map. The depth residual map and initial depth map are added together to generate the refined depth map. To adapt this network for deep Helmholtz stereopsis, the reference image is first concatenated with their corresponding light source position before it is concatenated with the initial depth map, which is then fed as input to the refined network. The other change is that the average pooling operation is used to fuse the initial depth map and the output of the refined network in order to generate the refined depth map.

The loss for each depth map is calculated as the mean absolute difference between the estimated depth map and the ground truth. The loss function of the network is the sum of both losses:

$$\mathrm{L}_{\text{depth}} = \mathrm{L}_{\text{initial depth}} + \mathrm{L}_{\text{refined depth}} \tag{4.10}$$

4.3 PS-FCN

PS-FCN (figure 4.6) predicts the normal of a scene from an arbitrary number of images of a view of a scene that have been captured under different lighting directions. It takes an arbitrary number of images, N, along with their corresponding light direction as its input. The following subsections describe how the network estimates the normal map for a single view.

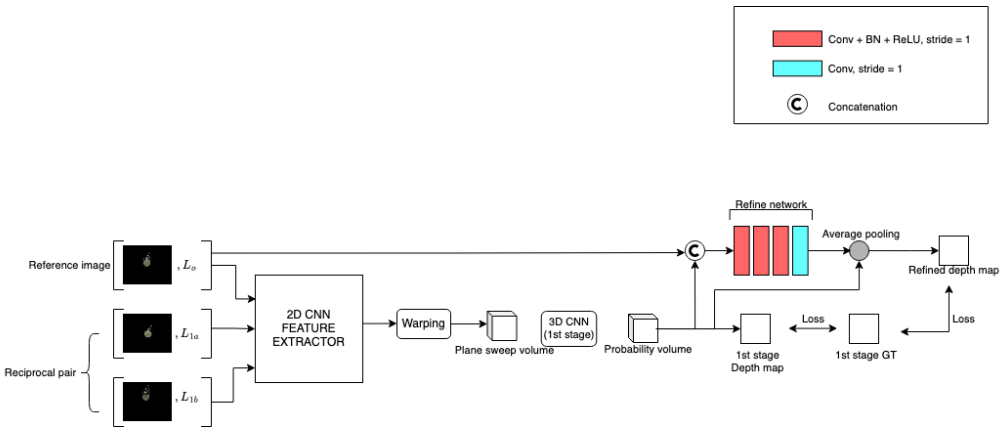

Figure 4.5: Adapted UCS-Net Architecture.

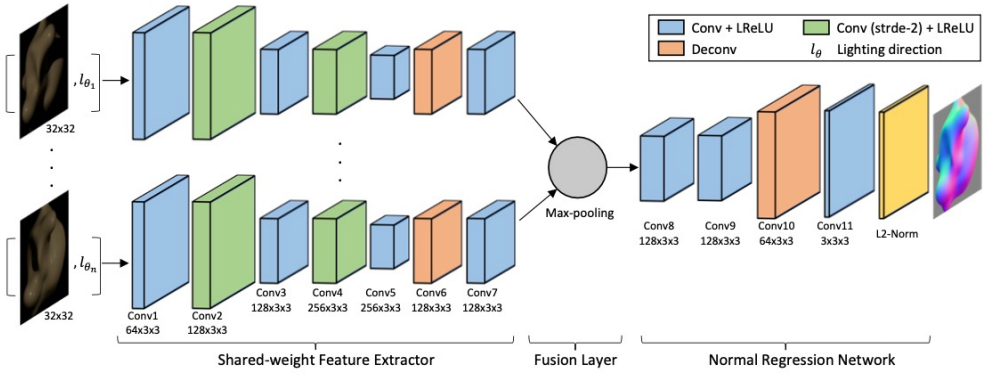

Figure 4.6: The PS-FCN architecture. Extracted from [14].

4.3.1 Extracting Features

Each input image is concatenated with its corresponding light direction and fed into its own feature extractor. The feature extractors share the same weights. A feature extractor consists of seven convolutional layers. Each of the convolutional layers is followed by a Leaky ReLU layer, except the sixth layer. Using a stride of two, the second and fourth layer down-samples the input by a factor of 2 while the sixth layer (deconvolutional layer) up-samples its input by a factor of two. At the end of the feature extractor, the most salient features in the input image are extracted into a 128-channel feature map.

4.3.2 Fusing Extracted Features

Each of the feature extractors generate a 128-channel feature map corresponding to an input image. These feature maps are fused together using the max-pooling operation to extract the global information of the scene. It is fed as input into the normal regression network.

4.3.3 Estimating Initial Normal Map

The normal regression network regresses the surface normal of the scene using the fused feature map. It consists of four convolutional layers. Each of the convolutional layers is followed by a Leaky ReLU layer, except the third layer. The third layer (deconvolutional layer) up-samples its input by a factor of 2. L2 normalization is performed on the output of the last layer of the regression network to generate the estimated normal map.

4.3.4 Loss

The network is regressed using the cosine-similarity loss function. The closer the normal map is to the ground truth map, the smaller the average angle between the normal map and the ground truth, which means that the error between the maps decreases. The loss function is calculated as:

$$\text{Loss} = \frac{1}{hw}\sum_{i}^{hw}(1 - {n_i}^T\tilde{n}_i), \tag{4.11}$$

where for each pixel, i, n_i is the predicted normal and $\tilde{n}_i$ is the ground-truth. h and w are the height and width of the input image, respectively.

4.4 Adapting PS-FCN for Deep Helmholtz Stereopsis

With the exception of the loss function, all other aspects of the PS-FCN network were adapted to make it suitable for deep Helmholtz Stereopsis. The following subsections describe the changes made to the architecture. Figure 4.7 shows the adapted network.

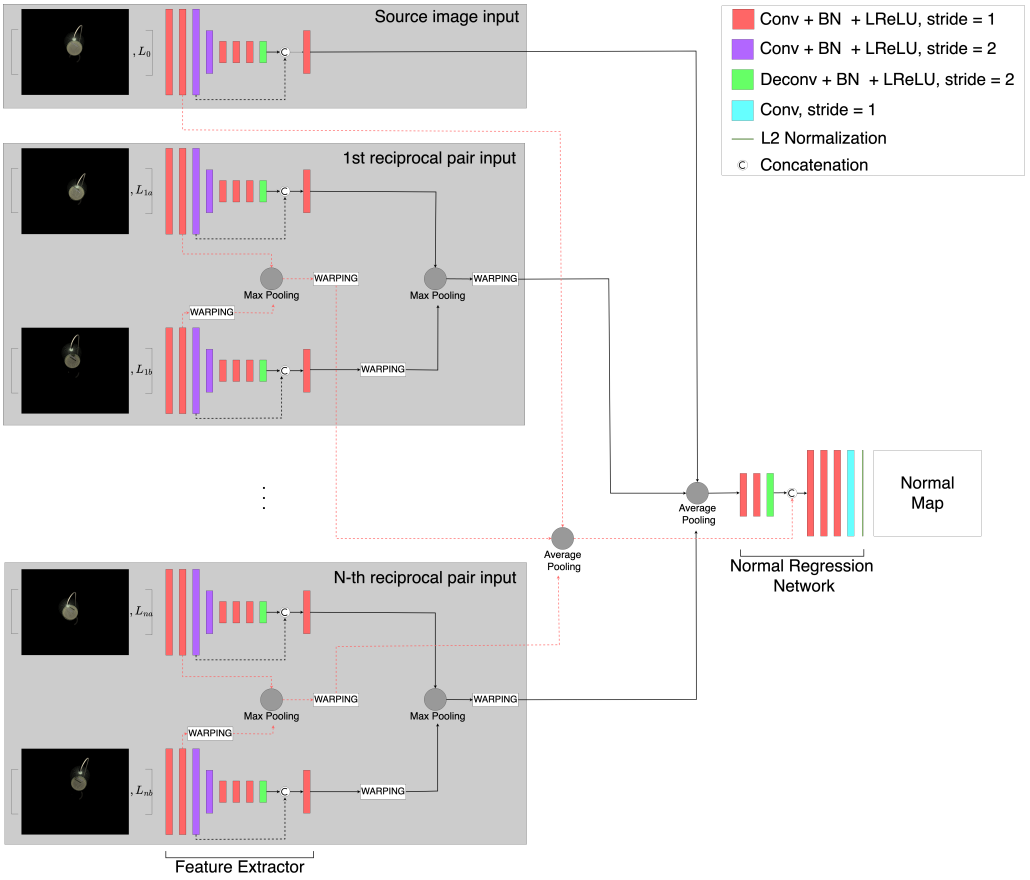

Figure 4.7: The Helmholtz-PS architecture.

4.4.1 Extracting Features

The input images are the reference image and its N reciprocal pairs of images. Each input image is concatenated with its corresponding light source position and fed into its own feature extractor. Similarly to PS-FCN, the feature extractors share the same weights. In this adaptation, a feature extractor is made up of nine convolutional layers with two skip connections in the second and third layers. Each of the convolutional layers is followed by a batch normalization layer and a Leaky ReLU layer, except the third, fourth and eighth layers. Using a stride of 2, the third and fourth layers down-samples their inputs by a factor of 2, while the eighth layer (deconvolutional layer) up-samples its input by a factor of 2. The output of the eighth layer is concatenated with the output of the third layer to recover lost features before it is passed as input to the final layer. The final layer outputs a 64-channel feature map that corresponds to its input image.

4.4.2 Fusing Extracted Features

After the features have been extracted from the input images, they are fused together in stages. In the first stage, for each reciprocal pair, the feature map of one image is warped with respect to its

reciprocal image using the same differentiable homography used in section 4.1.3 for generating the cost volume, except depth hypotheses are not used as parameters. The warping process ensures the feature maps are fronto-parallel before fusion. After warping, the feature maps are fused together using the max-pooling operation to extract the global information for each reciprocal pair. In the second stage, the feature maps of the reciprocal pairs, which now contain the global information for their corresponding reciprocal pair are warped with respect to the reference image. The results are fused together with the feature map of the reference image using the average-pooling operation. The aggregated feature map is fed as input to the normal regression network.

4.4.3 Estimating Initial Normal Map

In this adaptation, the normal regression network consists of seven convolutional layers. Each of the convolutional layers is followed by a batch normalization layer and a Leaky ReLU layer except the third and the last layer. Similarly to the PS-FCN architecture, the third layer (deconvolutional layer) up-samples its input by a factor of 2. To recover some of the features that may have been lost during the feature extraction process, the outputs of the second layer of the feature extractor are fused together in a similar manner as described in the previous sub-section (figure 4.7). The aggregated feature map is concatenated with the output of the deconvolutional layer and fed as input to the fourth layer of the regression network. Similarly to PS-FCN, L2 normalization is performed on the output of the last layer of the regression network to generate the estimated normal map.

5 RESULTS AND DISCUSSIONS

Implementation

Both architectures were implemented in Pytorch. They were trained using Adam [31] optimizer with default beta coefficients (0.9 and 0.999) and a default weight decay of 0. They were trained at an initial learning rate of 0.001, which reduced by a factor of 2 every 5 epochs. The values of the gradients generated by the architectures were clipped such that their L2-norm did not exceed 2.2. Each architecture was trained on 2 GPUs that were automatically selected by HTCondor [3]. The main GPUs used were the Quadro RTX 5000 and the GeForce RTX 3090. The dataset of 4,676 models was split, such that 80% of the dataset were used for training, 10% for validation and the remaining 10%, for testing. Each model has 8 view images. Thus, the architectures were trained on 29,928 images.

This chapter discusses the results of the depth and normal inference of the adapted deep multi-view stereo network and the adapted deep photometric stereo network. Section 5.1 discusses the results of the deep multi-view stereo network. Furthermore, it discusses the performance of the network on different parameter settings. Section 5.2 discusses the results of the deep photometric stereo network. It also discusses the performance of the network on different parameter settings.

5.1 Deep Helmholtz Multi-view Stereo

The deep multi-view stereo network was trained with a batch size of 8. It converged in 59 epochs. Figure 5.1 shows the train/validation loss of the network. It was tested on 3,736 view images of the test models. The performance of the network was measured using an accuracy metric, where higher accuracy means better performance. The accuracy of the network is calculated based on the precision/recall measures in [32]. Let $\mathcal{G}$ be the set of valid points on the ground truth map and $\mathcal{P}$ be the set of corresponding points in the predicted depth map. Let $[d_{\text{min}}, d_{\text{max}}]$ be the depth range of the ground truth and let the number of depth hypotheses be 64, the depth interval, d_{interval} is $\frac{d_{\text{max}} - d_{\text{min}}}{64}$. If the error, e_i for a particular point, $i \in \mathcal{G}$ is $\frac{\mathcal{G}_i - \mathcal{P}_i}{d_{\text{interval}}}$, the accuracy of a predicted depth map, $\mathbf{I}$ can be calculated as:

$$\text{Accuracy}(\mathbf{I}) = \frac{100}{|\mathcal{G}|} \times \sum_{i \in \mathcal{G}} [e_i < d_T], \tag{5.1}$$

where d_{T} is a distance threshold and $[\cdot]$ is the Iverson bracket that evaluates to 1 if '$\cdot$' is true, or 0 otherwise. For the experiments, d_{T} was set to 2.

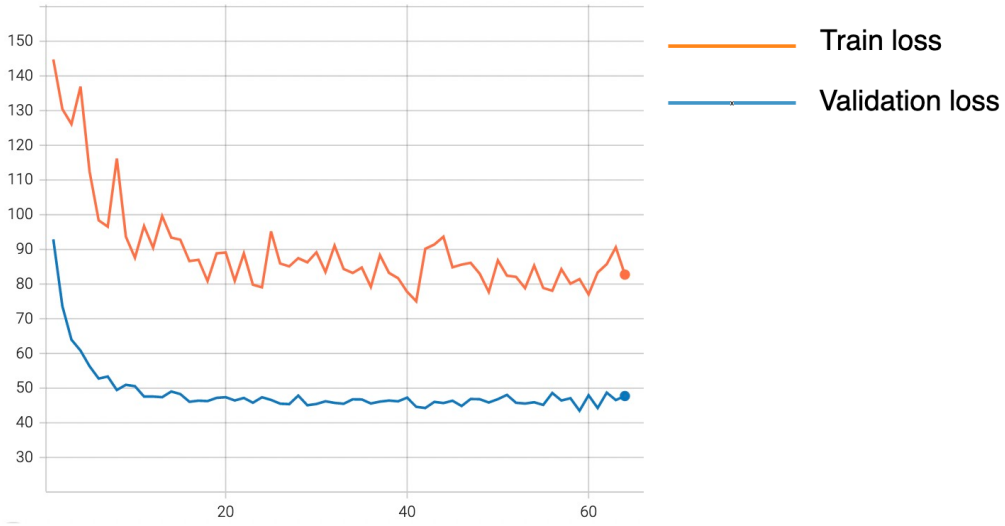

Figure 5.1: Train/Validation loss of the deep multi-view stereo network.

The network predicts two depth maps, the initial depth map and the refined depth map. Table 5.1 shows that each depth map performed better on some of the selected test images than the other depth map. Figure 5.2 shows the results on selected test images. The error map is calculated as the absolute distance between the ground truth and the predicted depth map. The results show that the network is able to produce decent results on different types of models. Furthermore, the results show that although the initial depth map produced better results on some of the test images than the refined depth map, it had rougher surfaces than the refined depth map. This suggests that the initial depth map focuses more on accuracy than global consistency while the refined depth map focuses on both accuracy and global consistency, which may be a possible reason why the refined depth map produced a lower accuracy on average than the initial depth map. Using the 'End table' test image as an example, the initial depth map is more accurate than the refined depth map but on closer inspection (figure 5.3), it has a rougher surface (that is not consistent with the ground truth) than the refined depth map.

Table 5.1: Comparison of the depth maps accuracies. Higher Accuracy means better result.

	Accuracy on all test images ($d_{\mathbf{T}} = 2$)	**Accuracy on selected tests images** ($d_{\mathbf{T}} = 2$)						
		Laptop	**Horse**	**Soda can**	**End table**	**Ceiling lamp**	**Standing person**	**Car**
Initial depth map	**59.11**	87.13	**93.47**	82.73	**90.45**	85.53	**88.96**	82.51
Refined depth map	55.71	**91.21**	76.35	**85.59**	87.30	**93.53**	81.25	**82.83**

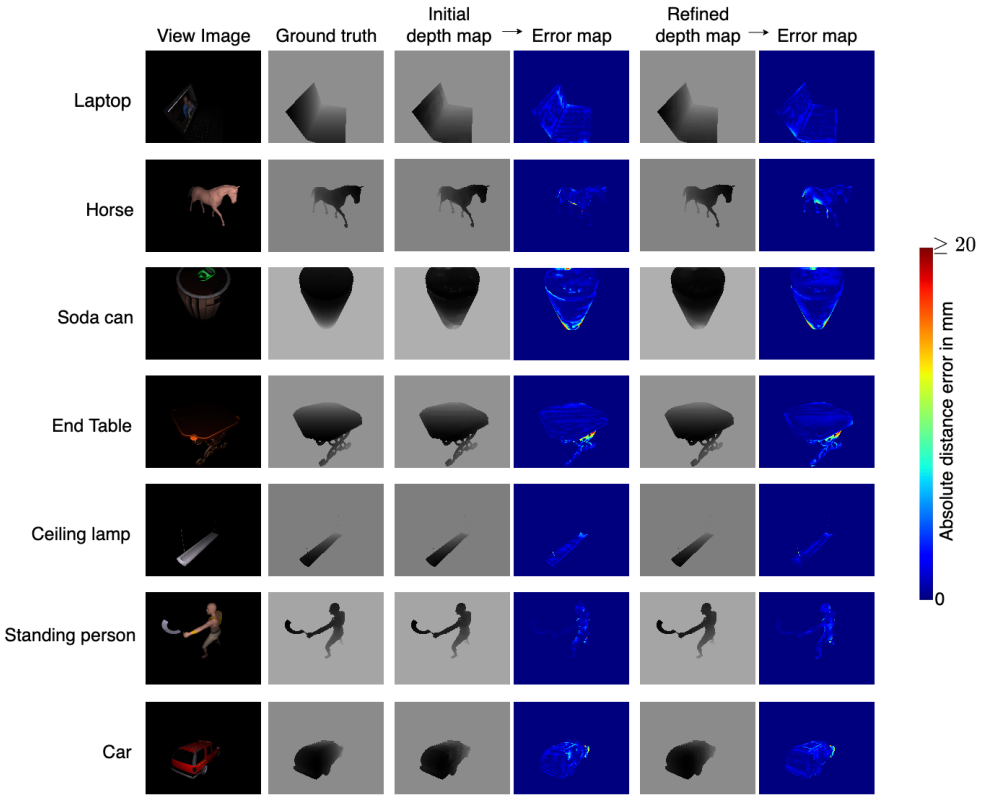

Figure 5.2: Depth Map inference results on selected test view images.

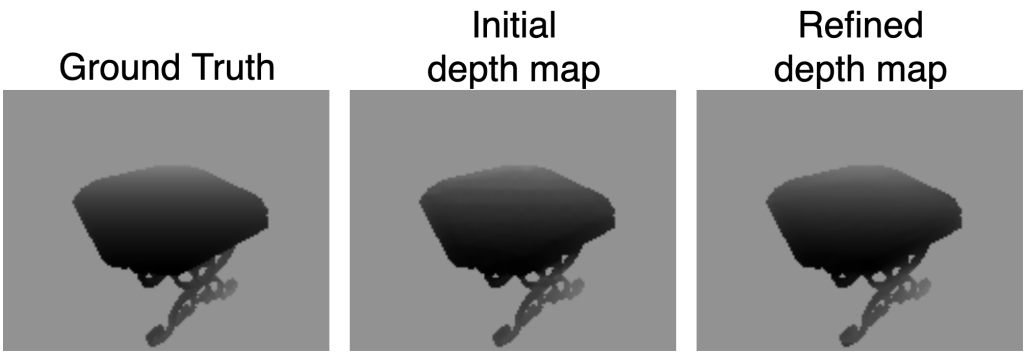

Figure 5.3: The initial depth map is more accurate than the refined depth map but it has a rougher surface than the refined depth map.

5.1.1 Ablation Study

Effects of Refined Network

I compared the performance of the architecture with the refine network with the performance of the architecture without the refine network. Table 5.2 shows that the network without the refine network performed better on average. A possible reason for this could be the same reason as to why the initial depth map performed better than the refined depth map. For instance, on the 'Soda can' test image, the architecture without the refine network performed better but it produced a rougher surface than the refined depth map of the architecture with the refine network (figure 5.4). The initial depth map of the architecture with the refine network performed a little worse than the architecture without a refine network. This may suggest that the initial depth map may be taking the global consistency into account but at a lower rate than the refined depth map.

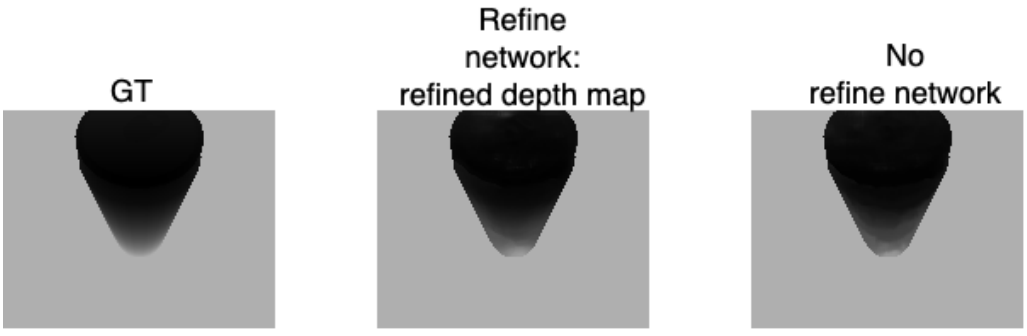

Figure 5.4: The depth map inferred by the architecture with no refine network has a rougher surface than the refined depth map of the architecture with the refine network.

Effects of feeding the light source positions as an input

I compared the effects of passing the light positions as parts of the inputs to the network. I

Table 5.2: Comparison in performance between the architecture with a refine network and the architecture without the refine network. Higher accuracy means better result.

	Accuracy on all test images ($d_{\mathbf{T}} = 2$)	**Accuracy on selected tests images** ($d_{\mathbf{T}} = 2$)						
		Laptop	**Horse**	**Soda can**	**End table**	**Ceiling lamp**	**Standing person**	**Car**
No refine network	**59.89**	86.28	92.7	**86.94**	83.74	88.24	**92.50**	81.24
Refine network: Initial depth map	59.11	87.13	**93.47**	82.73	**90.45**	85.53	88.96	82.51
Refine network: Refined depth map	55.71	**91.21**	76.35	85.59	87.30	**93.53**	81.25	**82.83**

analyzed the effects on the architecture with the refine network and the architecture without the refine network. Table 5.3 and Table 5.3 show that the architectures in which its inputs contain the positions of the light source performed better than the architectures that do not use light source positions as part of its input. This shows the relative importance of using the light source positions. Figures 5.5 and 5.6 show this increase in performance as well, but at a less observable scale.

Table 5.3: Comparison between the architecture that uses light positions as parts of its input and the architecture that does not use light positions as parts of its inputs. Higher accuracy means better result.

	Accuracy on all test images ($d_{\mathbf{T}} = 2$)	**Accuracy on selected test images** ($d_{\mathbf{T}} = 2$)						
		Laptop	**Horse**	**Soda can**	**End table**	**Ceiling lamp**	**Standing person**	**Car**
Refine network, no light: Refined depth map	55.6	86.33	66.1	**87.51**	71.68	93.29	**87.05**	82.47
(default) Refine network and light: Refined depth map	**55.71**	**91.21**	**76.35**	85.59	**87.30**	**93.53**	81.25	**82.83**

Table 5.4: Comparison between the architecture that uses light positions as parts of its input and the architecture that does not use light positions as parts of its inputs. Higher accuracy means better result.

	Accuracy on all test images ($d_{\mathbf{T}} = 2$)	**Accuracy on selected test images** ($d_{\mathbf{T}} = 2$)						
		Laptop	**Horse**	**Soda can**	**End table**	**Ceiling lamp**	**Standing person**	**Car**
No refine network, no light	59.3	84.54	84.54	74.98	**84.11**	86.24	90.09	76.61
No refine network but with light	**59.89**	**86.28**	**92.7**	**86.94**	83.74	88.24	**92.50**	**81.24**

Effects of the number of reciprocals

I analyzed the effects of the number of reciprocal pairs of images on the performance of the network. Table 5.5 suggests that, as the number of reciprocal pairs increased, the average performance of the network improved as well, although the rate of increase in performance decreased accordingly, which suggests that there is a limit to the performance gained by increasing the number of reciprocal pairs. The table also shows that the performance of the network on the selected test images depends on the number of reciprocal pairs used. For instance, a network trained with 4 reciprocal pairs produced a relatively-poor performance when tested on the ”end table” image. This large variation in performance can be further visualized in figure 5.7. Despite this variation, it can be observed in the graph that using 1 reciprocal pair produced the worst performance on average among the tested numbers of reciprocal pairs, since there are more points above the ”1 reciprocal pair” line than under the line. Furthermore, the higher the number of reciprocal pairs, the closer their corresponding points are to the top of the graph, which provides more evidence that increasing

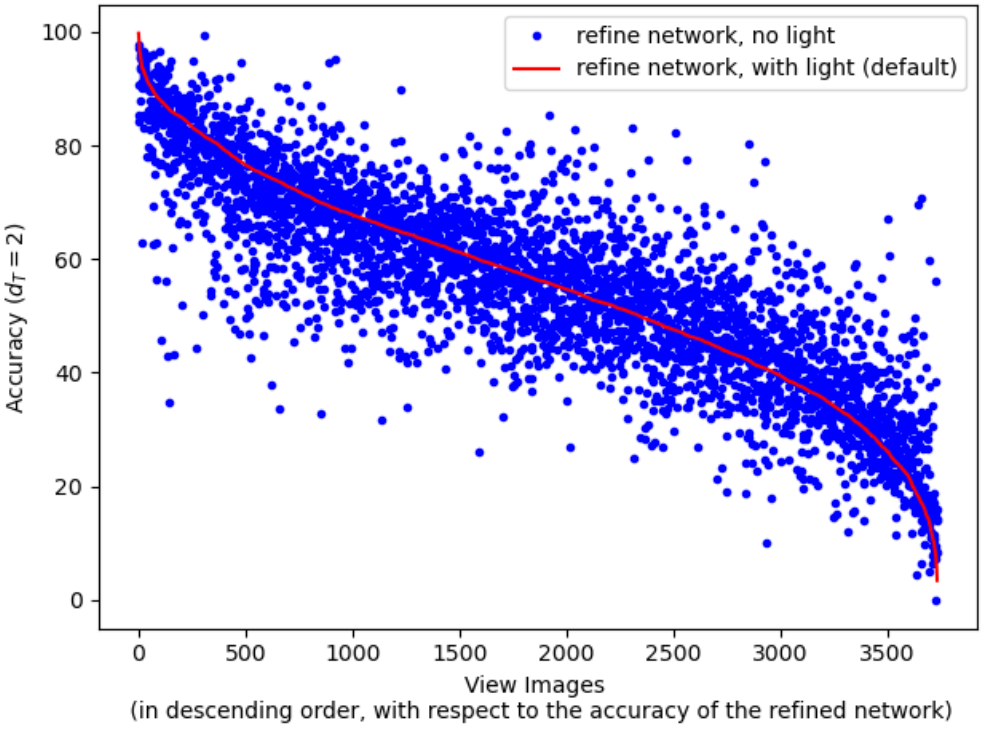

Figure 5.5: Comparison between the architecture that uses light positions as parts of its input and the architecture that does not use light positions as parts of its input

the number of reciprocal pairs may improve the performance of the network, although, this is less obvious when comparing ”3 reciprocal pairs” and ”4 reciprocal pairs” in the graph.

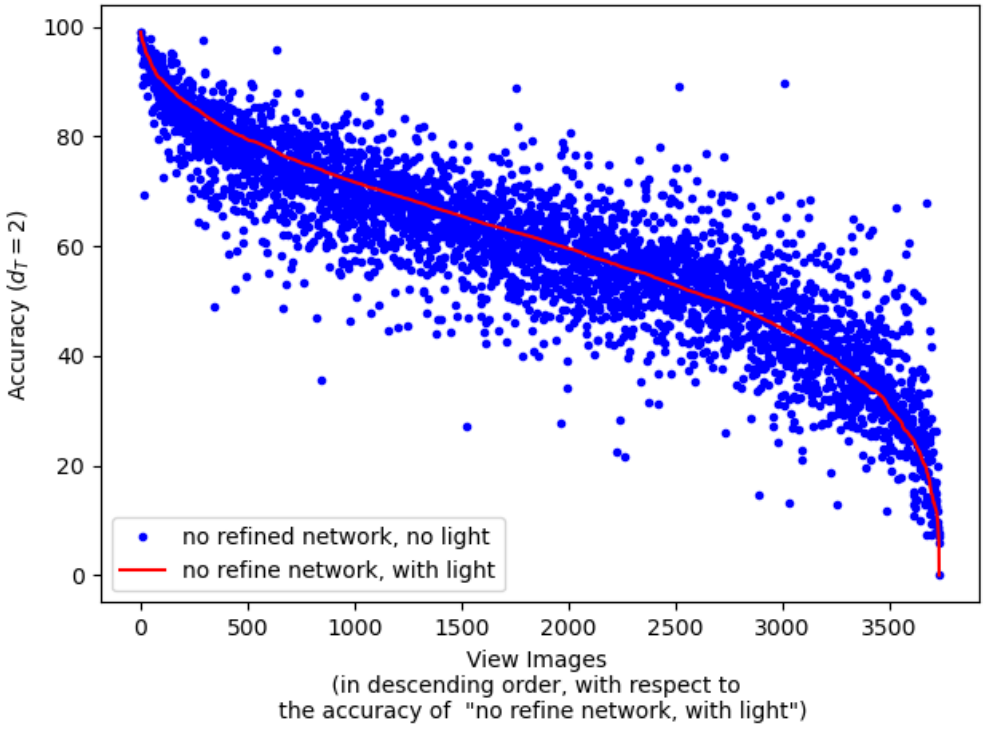

Figure 5.6: Comparison between the architecture that uses light positions as parts of its input and the architecture that does not use light positions as parts of its input

Table 5.5: Comparison of the default network on different numbers of reciprocal pairs. The evaluation is on the refined depth map. Higher accuracy means better result.

# **Reciprocals pairs**	**Accuracy on all test images** ($d_\mathbf{T} = 2$)	**Accuracy on selected test images** ($d_\mathbf{T} = 2$)						
		Laptop	**Horse**	**Soda can**	**End table**	**Ceiling lamp**	**Standing person**	**Car**
1	55.71	91.21	76.35	85.59	**87.30**	**93.53**	81.25	**82.83**
2	57.64	89.05	80.79	91.36	74.31	90.47	87.69	87.17
3	58.5	**95.28**	83.87	89.22	85.91	91.76	83.86	84.94
4	**59.11**	91.00	**84.67**	**92.56**	54.21	92.47	**86.55**	**87.53**

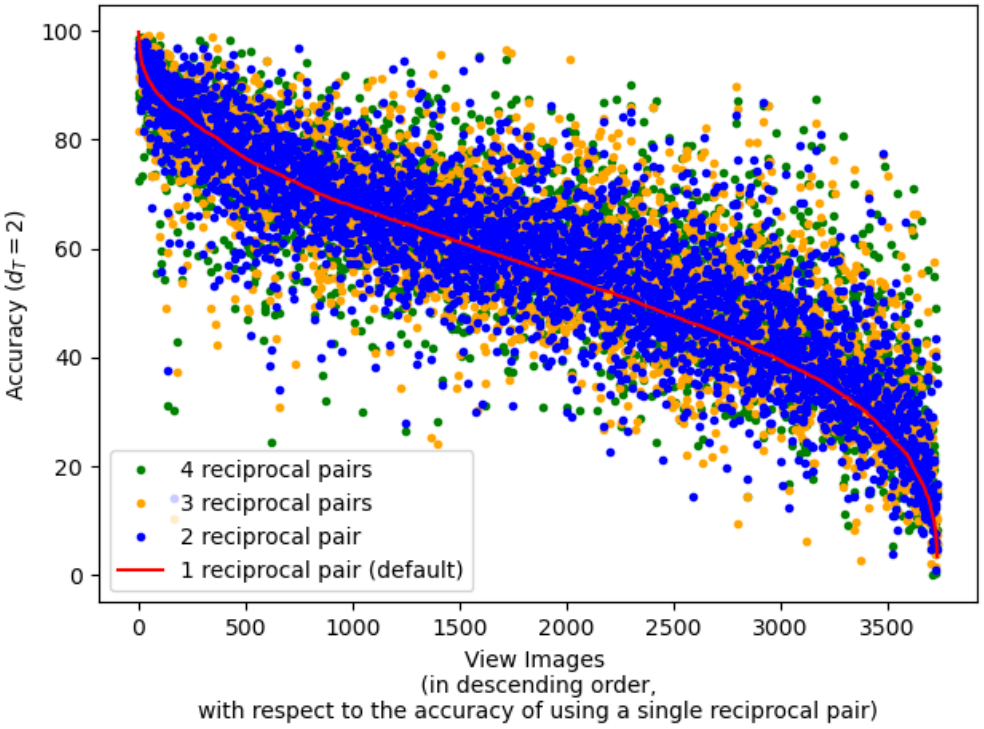

Figure 5.7: Performance of the deep multi-view stereo network network when using different numbers of reciprocal pairs. The evaluation is on the refined depth map. Higher accuracy means better performance. The figure shows that performance of the numbers of reciprocal pairs varies widely. Nevertheless, it can observed that using 1 reciprocal pair produced the worst performance on average, since there are more points above its line than below it. Furthermore, using 4 reciprocal pairs produced the best performance, since its points are the closest to the top of the graph, and the points below the ”1 reciprocal pair” line, which correspond to the numbers of reciprocal pairs, look random. This suggests that the performance of the network increases proportionally to the number of reciprocal pairs.

Using MSE loss function

The Mean Squared Error (MSE) function is a popular loss function that is used for training a variety of deep networks. However, when I used the MSE function to train the mvs network, the network failed to converge.

Limitations

A limitation of the network is that it may overcompensate when trying to learn complex depth information of the scene. Figure 5.8 shows how the network performed decently in recovering the depth information of the occluded regions of the model but overcompensates by losing some of the depth information in the lower part of the handle. Another thing to point out is the potential inconsistency of the network. For instance, when trying to test the effectiveness of the number of

reciprocal pairs, the performance of the networks across the test models varied too widely when using different numbers of reciprocal pairs, even though there was an observable pattern when the average performance of the network was considered.

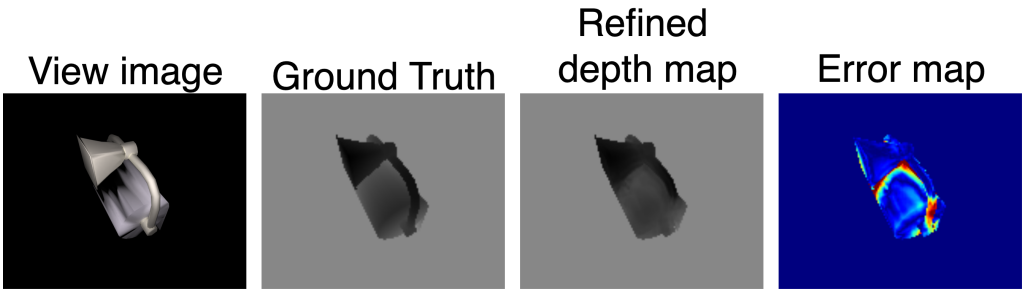

Figure 5.8: Although, the deep multiview stereo network is able to recover some of the depth information in the occluded regions of the model, it fails to correctly recover the depth information of the lower part of the handle.

5.2 Deep Helmholtz Photometric Stereo

The photometric network was trained with a batch size of 8. It converged in 58 epochs. Figure 5.9 shows the train/validation loss of the network. It was tested on 3,736 view images of the test models. To evaluate the performance of the network, I used the mean angular error (MAE) metric, which is measured in degrees. The MAE is the average angle between the ground truth surface normals and the inferred surface normals, which means that the lower the average angle between the ground truth and inferred normal map, the better the result. The mean angular error on the images was 26.75. Figure 5.10 shows the surface normal inference on selected test view images. The error map is calculated as the absolute angular error between the ground truth map and the inferred normal map. The inferred normal maps in the figure shows that the network is able to produce some decent results on different types of models. Furthermore, the estimated normal maps are at a much higher resolution than the estimated normal maps that are generated by the PS-FCN network. In the next subsection, I analyzed the performance of the network on different configurations. With the exception of the fusion configuration networks, the network configurations were trained using a batch size of 4 that was split between two GPUs.

5.2.1 Ablation Study

Effects of fusion type

In the PS-FCN paper [14], the authors found out in their experiments that the max-pooling operation consistently produced better results than the average pooling operation. I decided to analyze the performance of both operations on the photometric architecture. In the photometric architecture

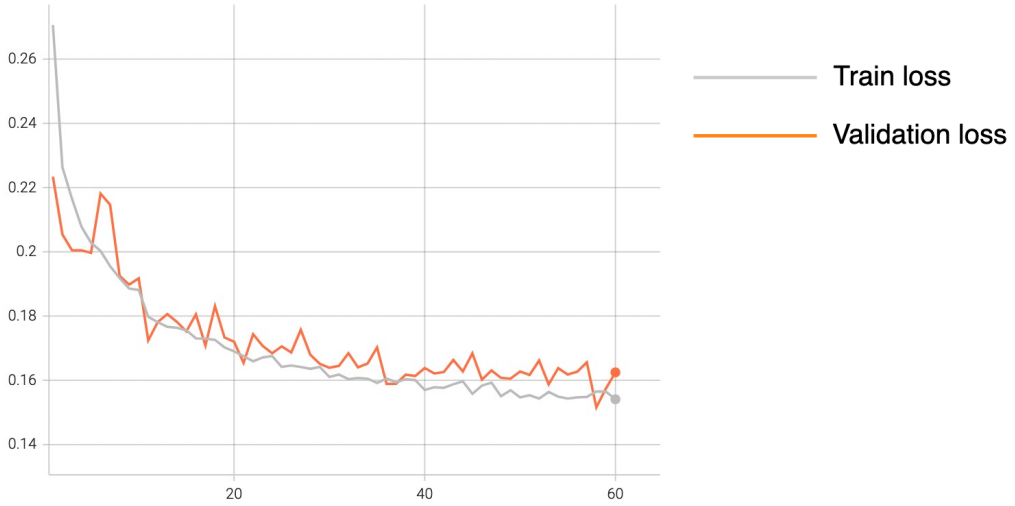

Figure 5.9: Train/Validation loss of the deep photometric stereo network.

(figure 4.7), there are two types of pooling operations that are used to fuse the features together. The first pooling operation is the max-pooling operation, which is applied to each reciprocal pair in order to fuse their features into a single feature map, such that each reciprocal pair has a fused feature map associated with it. The second pooling operation is the average-pooling operation, which is used to fuse the features of the reference image and the fused feature maps of the reciprocal pairs into a single feature map. I refer to this fusion configuration as the Max-Mean fusion type. I compared this configuration with the three other possible configurations in this setup, which are the Max-Max, Mean-Mean and Mean-Max Fusion types. Table 5.6 shows the performance of the fusion types on selected test models. Although, the Mean-Mean fusion type produced the best result, there is little difference in their performance. Figure 5.11 further shows that overall, there is little variation in the performance of the different fusion configurations on the test models when compared with the performance of the default fusion configuration (Max-Mean) on the test models, although, the Max-Max fusion type performs slightly worse than the other fusion types, which is the opposite of what one would expect, based on the experiments in the PS-FCN paper.

Effects of number of base channels

The number of base channels is the number of channels of the feature map that is generated by the first convolutional layer of the network. Table 5.7 shows the network performance on different numbers of base channels. The table suggests that on average, the network performs better as

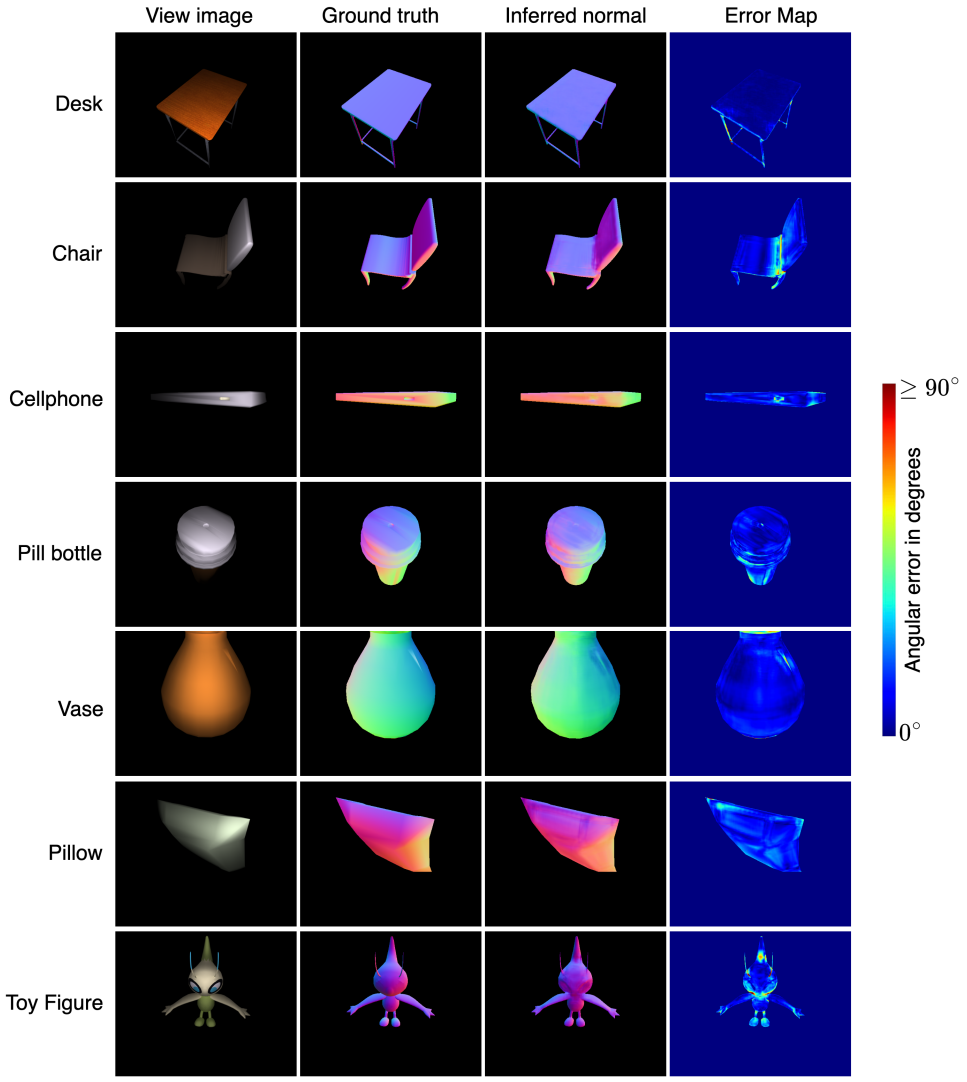

Figure 5.10: Normal Map inference results on the selected test images.

the number of base channels increases. However, the table also shows that the network with 24 base channels produced the best results on non-planar surfaces, which suggests that setting the number of base channels to 24 may be better suited for learning the surface normals of spatially-varying models than using the default number of 32. Figure 5.12 shows that overall, the network with 32 base channels performed better on the test view images than the other tested numbers of base channels, which suggests that the dataset may contain more planar surfaces than non-planar surfaces.

Table 5.6: Comparison of the photometric network on different feature fusion configurations. Lower MAE (in degrees) means better result.

Fusion Type	**Avg MAE on all test images**	**MAE on selected test images**						
		Desk	**Chair**	**Cellphone**	**Pill bottle**	**Vase**	**Pillow**	**Toy figure**
Max-Max	26.82	4.21	13.28	12.37	**10.33**	10.03	14.75	**17.31**
Max-Mean (Default)	26.75	**4.16**	13.55	**12.23**	10.82	9.43	**14.43**	17.97
Mean-Mean	**26.73**	4.68	**13.27**	13.36	10.78	**8.95**	14.96	17.94
Mean-Max	26.78	4.21	13.98	13.23	10.86	9.42	15.72	18.2

Table 5.7: Comparison of the photometric network on different numbers of base channels. Lower MAE (in degrees) means better result.

# base channels	**Avg MAE on all test images**	**MAE on selected test images**						
		Desk	**Chair**	**Cellphone**	**Pill bottle**	**Vase**	**Pillow**	**Toy figure**
8	29.61	5.59	20.12	14.67	13.31	12.31	15.74	20.27
16	28.40	5.43	16.41	13.62	11.76	11.11	16.65	18.73
24	28.07	4.99	15.17	**12.67**	11.85	**9.90**	**15.31**	**17.87**
32 (Default)	**27.70**	**4.51**	**14.80**	13.00	**11.43**	10.42	16.78	18.15

Effects of number of reciprocal pairs

I evaluated how the number of reciprocal pairs affected the performance of the deep photometric stereo network. Table 5.8 show that the performance of the network decreased as the number of reciprocal pairs increased, although, the margins of the performances are quite small. Figure 5.13 shows that overall, there is little variation in the performance of using different numbers of reciprocal pairs.

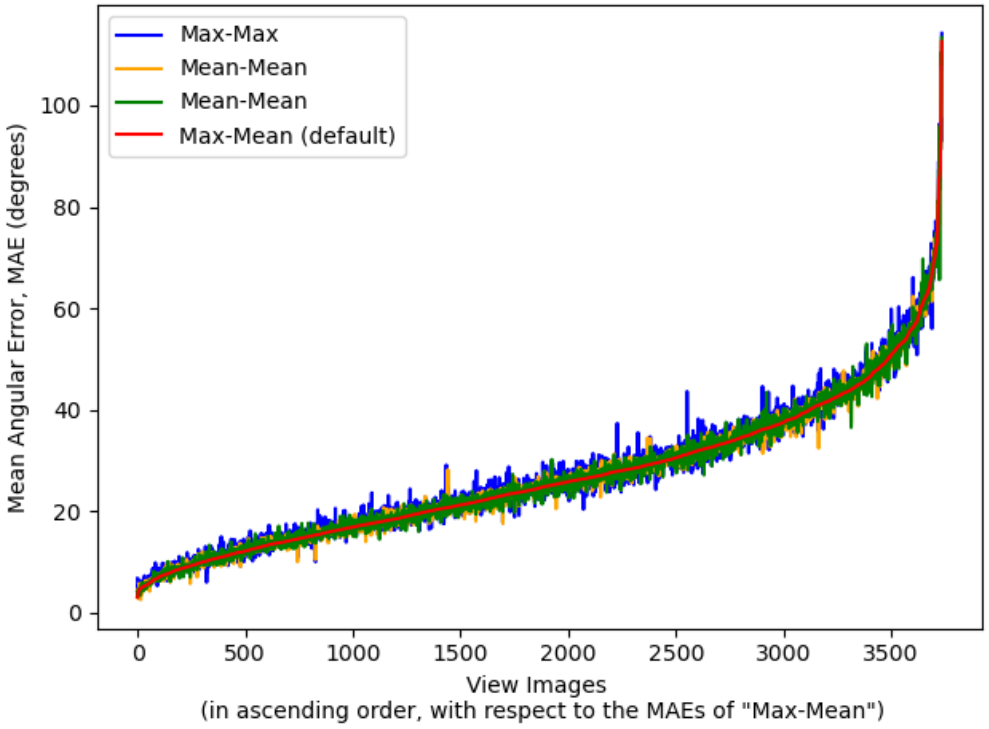

Figure 5.11: Performance of the fusion type configurations on the test images when compared to the performance of the default fusion type configuration, Max-Mean on the test images. Lower MAE means better performance. The figure shows that overall, changing the fusion type does not improve the performance of the network by a noticeable margin.

Table 5.8: Comparison of the photometric network on different numbers of reciprocal pairs. Lower MAE (in degrees) means better result.

# Reciprocal pairs	**Avg MAE on all test images**	**MAE on selected test images**						
		Desk	**Chair**	**Cellphone**	**Pill bottle**	**Vase**	**Pillow**	**Toy figure**
1 (Default)	**27.70**	**4.51**	14.80	**13.00**	11.43	10.42	16.78	18.15
2	27.78	5.07	**14.19**	13.68	10.77	10.59	15.26	**17.68**
3	27.84	4.89	15.04	13.47	**10.66**	11.14	15.15	17.83
4	28.06	4.83	15.49	14.02	11.12	**9.07**	**15.12**	18.67

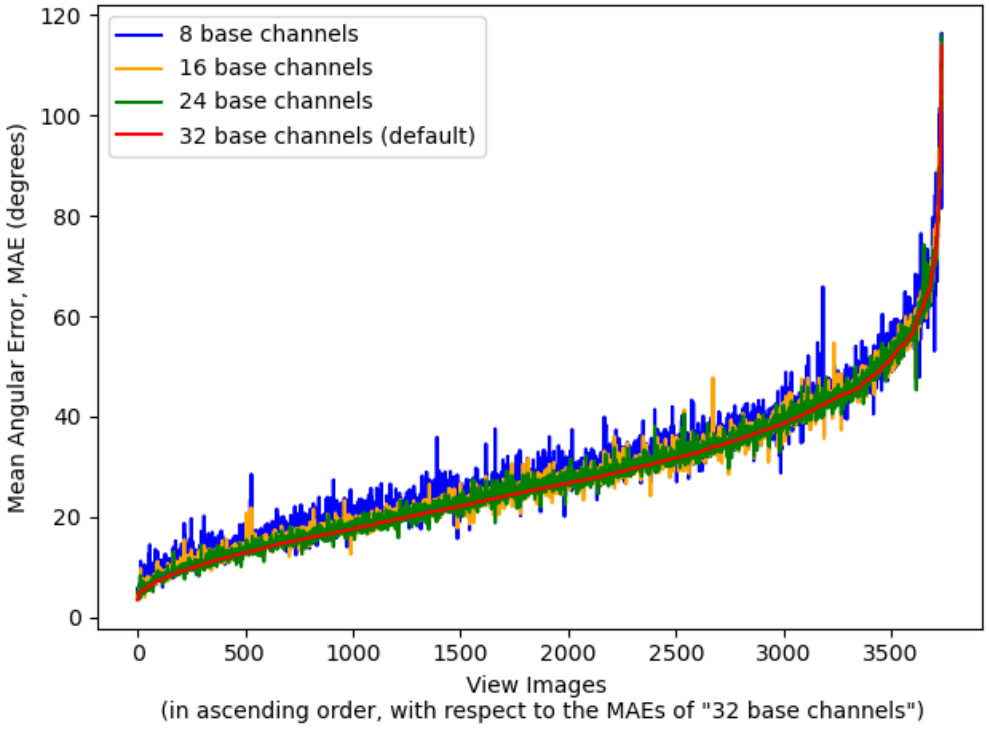

Figure 5.12: Performance of the network on different numbers of base channels when compared with the performance of the network on the default number of base channels (32). Lower MAE means better performance. The figure shows that as the number of base channels increased, the performance increased as well

Cosine similarity vs MSE

I analyzed how the MSE loss function performs in comparison with the cosine similarity loss function on the deep photometric stereo network. Table 5.9 shows that the MSE function performed better than the cosine similarity function, both on average and in the selected test images. However, figure 5.14 shows that the performance difference between the two loss functions on all the test images is not as clear as the table shows.

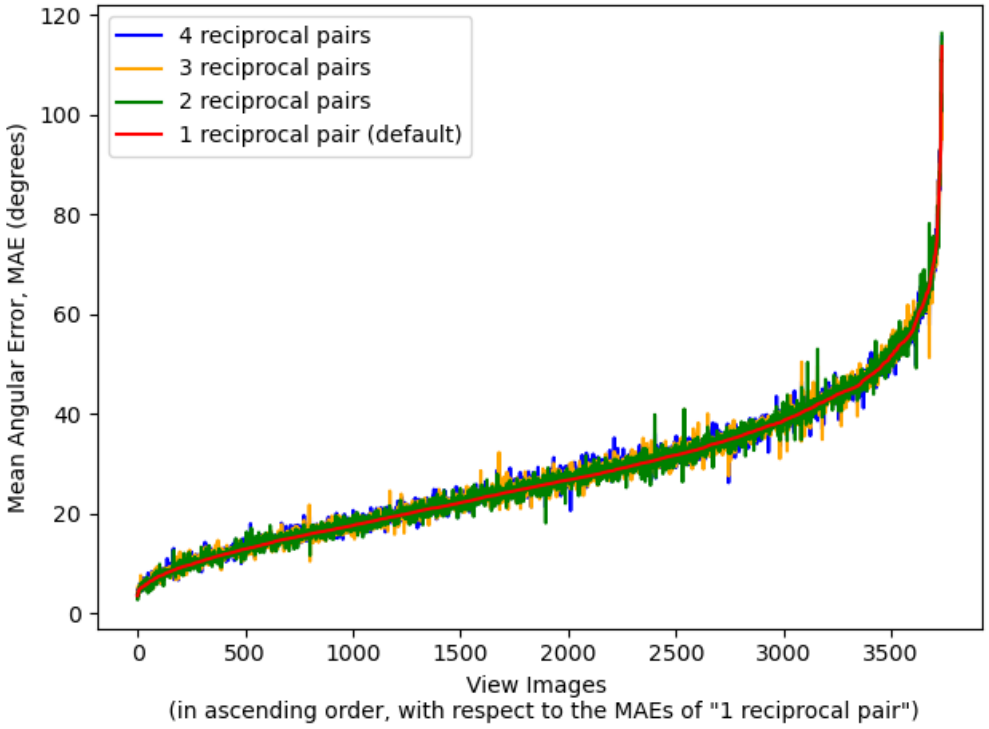

Figure 5.13: Performance of the network on different number of reciprocal pairs when compared with the performance of the network on the default number of pairs (1). Lower MAE means better performance. The figure shows that as the number of reciprocal pairs increased, the performance decreased.

Table 5.9: Cosine similariy loss function vs MSE loss function. Lower MAE (in degrees) means better result.

	Avg MAE on all test images	**MAE on selected test images**						
		Desk	**Chair**	**Cellphone**	**Pill bottle**	**Vase**	**Pillow**	**Toy figure**
Cos Similarity (default)	27.70	4.51	14.80	13.00	11.43	10.42	16.78	18.15
MSE	**27.56**	**4.43**	**14.67**	**12.47**	**11.12**	**9.01**	**14.95**	**18.12**

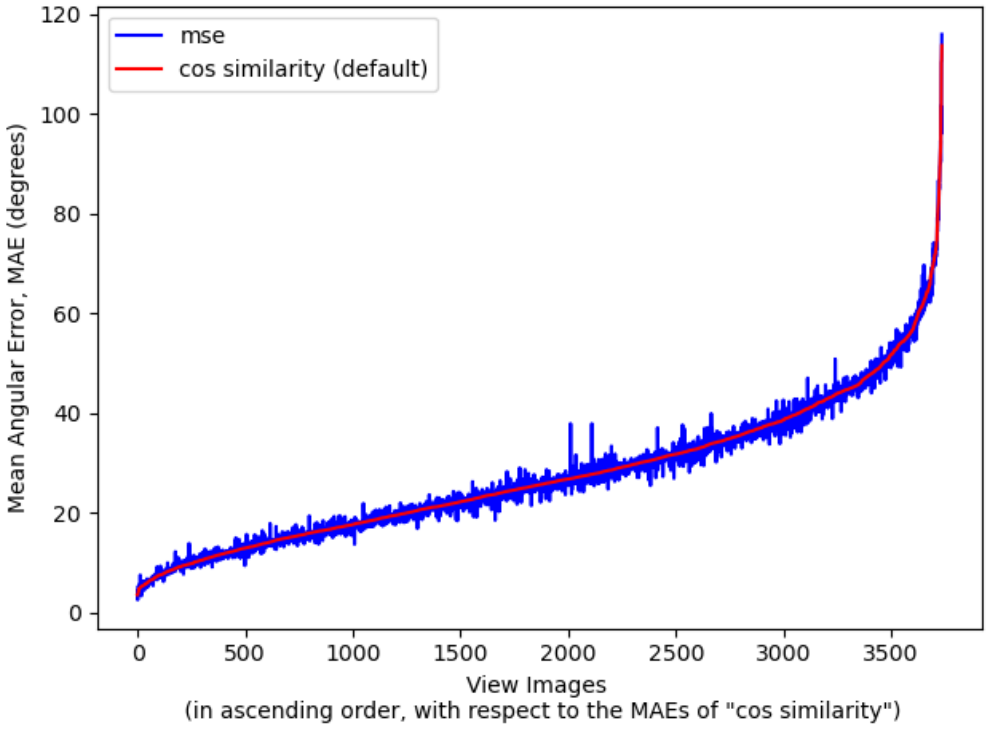

Figure 5.14: Performance of the network when using the cos similarity function as the loss function vs performance of the network when using the MSE function as the loss function. Lower MAE means better performance. The figure shows that as there is no clear performance difference between the two loss functions.

5.2.2 Limitation

The main limitation of the network is that it loses some of the the normal information of a complex surface (figure 5.15). The dataset contains some transparent and translucent objects. The network may infer the normals of surfaces that are behind transparent/translucent surfaces while the ground truth normal map contains only the normals of the surfaces with the nearest depth. Figure 5.16 shows an example of this incorrect normal inference by the network.

The network performs poorly on some of the models in the test images. One possible reason for this is the geometric inconsistencies of the models, which in turn makes the ground truth surface normals inconsistent. Figure 5.17 shows an example of a geometrically-inconsistent model and how the network attempts to correct the inconsistencies. The training dataset possibly contains some geometrically-inconsistent models. These models impact the network's ability to better learn the non-linearity relationship between the surface normals of a scene and its captured images. As a result of the inconsistent normals, the network may not get closer to the true normal, since any

attempt at approaching the true normal is calculated as a loss. However, the inconsistent models are in the minority of the dataset, which makes the network robust enough to be able to learn some of that relationship.

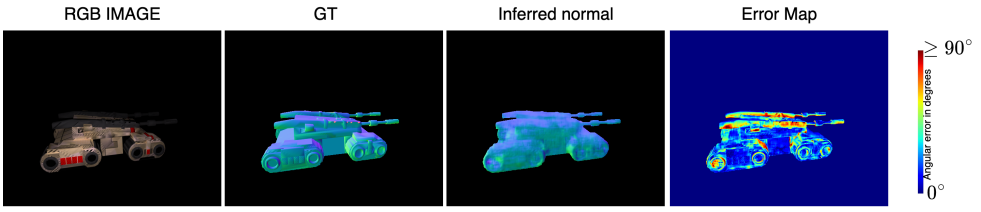

Figure 5.15: The network struggles to infer complex normal information.

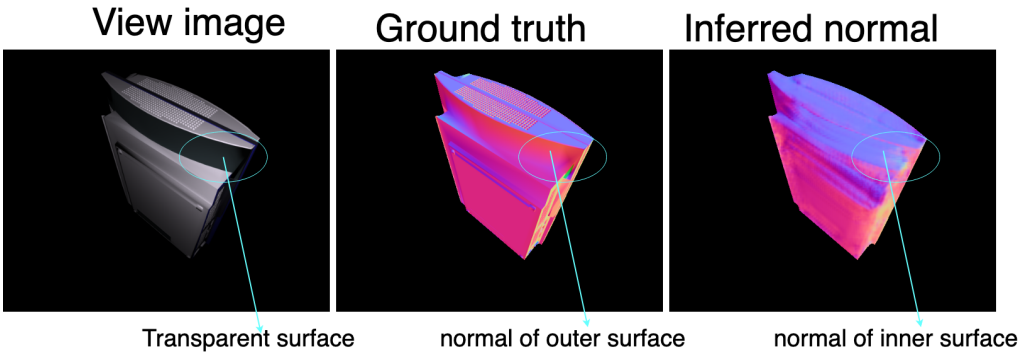

Figure 5.16: The network struggles to infer the correct normal of a transparent surface.

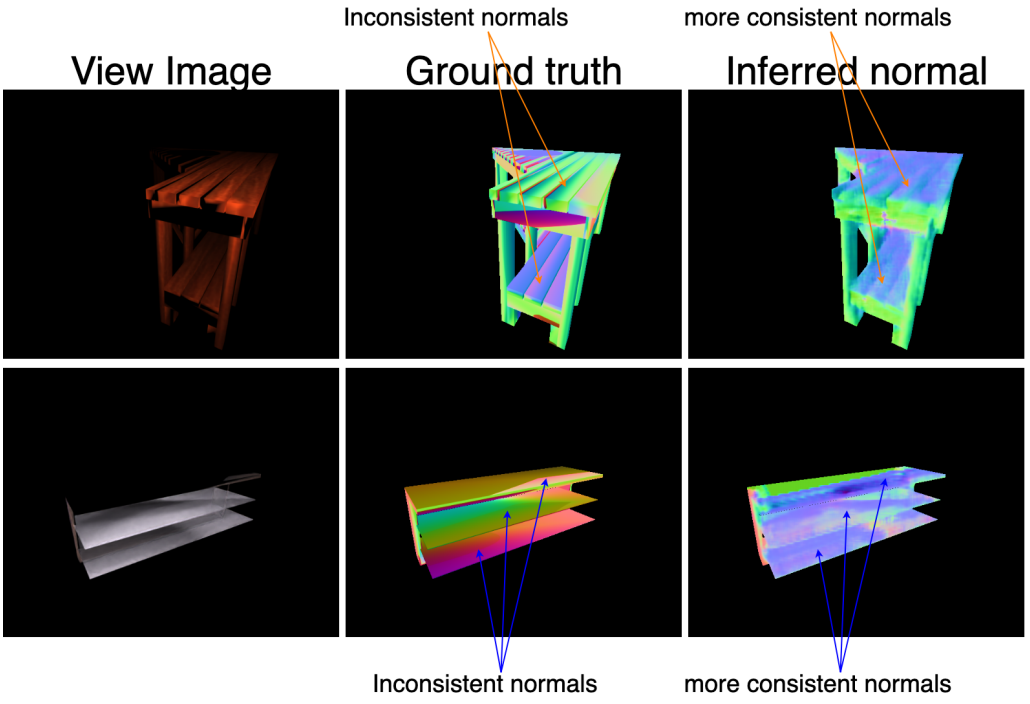

Figure 5.17: Geometeric inconsistencies of the ground truth normal and the network's best attempts to correct this.

6 CONCLUSIONS

Helmholtz Stereopsis is a very useful method to for reconstructing scenes with unknown and spatially-varying reflectance. However, it requires a cumbersome process to set up the Helmholtz environment. Furthermore, there is no deep learning architecture for Helmholtz Stereopsis, despite the prevalence of deep learning in 3D reconstruction. To address these problems, firstly, I designed a method to automatically generate reciprocal pairs of images, which eases the set-up process. Secondly, I used this method to generate a Helmholtz dataset. Finally, I adapted a deep multi-view stereo network and a deep photometric stereo network for deep Helmholtz Stereopsis. The adapted networks were trained using the generated Helmholtz dataset. After training, the adapted multiview stereo network was able to infer the the depth of scenes using deep Helmholtz Stereopis, and the adapted photometric stereo network was able to to infer the surface normals of scenes using deep Helmholtz Stereopsis. The estimated depth and normal maps of the scenes can be used for the reconstruction of the scenes.

Chapter 2 of the dissertation described existing methods of 3D reconstruction. It discussed the classical methods of 3D reconstruction, before discussing deep learning methods in the literature, as well as its corresponding datasets. The chapter ends with a justification of the need for a deep Helmholtz network. Chapter 3 discussed the dataset that was modified to a Helmholtz dataset. Furthermore, it discussed the process of modification. The chapter ends with an evaluation of the generated Helmholtz dataset. Chapter 4 described the deep multi-view stereo network and the deep photometric network that were adapted for Helmholtz Stereopsis. Furthermore, it discussed how the architectures were adapted for Helmholtz Stereopsis. Chapter 5 discussed the inference results generated by the architectures. It showed that the networks had promising results on simple to moderately-complex models. It also showed how they performed worse as the complexity of the models increased. Furthermore, it discussed the performance of the architectures on different configuration settings. It showed how some of the configurations settings, such as the fusion type setting (for the deep photometric stereo network) made little to no improvement on the default architecture. On the other hand, it showed how certain configuration settings, such as the reciprocal pairs setting (for the deep multi-view stereo network) can improve the performance of the default network. Finally, the chapter discussed the limitations of the architectures. For the deep multi-view stereo network, it discussed how the network's attempt to solve a difficult problem may lead

to even more problems. It also discussed the potential inconsistency of the network. For the deep photometric stereo network, it discussed the network's limitation in inferring the surface normals of complex models. It discussed how the network struggled to handle transparent/translucent models. Finally, it discussed the geometric inconsistencies that may exist in the Helmholtz dataset, and how they can negatively affect the performance of the network.

6.1 Evaluation

The objectives of the project were to generate a Helmholtz dataset, adapt existing deep neural networks to infer the depths and surface normals of scenes using deep Helmholtz Stereopsis and to write a dissertation report on the project. I have accomplished these goals and I have showed that the networks produced promising results, especially on simple models. Despite the limitations, I believe that I have filled an existing gap in the literature, and I hope that the project can serve as a starting point for designing deep neural networks that can better utilize Helmholtz reciprocity to infer the depths and surface normals of complex scenes.

6.2 Future Work

6.2.1 Helmholtz Dataset

The Helmholtz dataset was generated using the ShapeNetSem dataset. Despite some complexity in the dataset, it is dominated by models with planar surfaces, which may make the networks biased towards those surfaces, as shown in chapter 5. Multiple datasets from different distributions can be used to generate a Helmholtz dataset, which will make the deep Helmholtz networks more robust to spatially-varying scenes. To better handle occluded areas, the baseline (which is the gap between the reference camera and reciprocal camera) can be reduced but the gap has to be big enough in order to enable a potential deep Helmholtz network to fully leverage Helmholtz Reciprocity. Finding the right baseline for the Helmholtz network is a difficult problem which I believe will require a lot of experimentation.

6.2.2 Deep Helmholtz Network

The proposed deep multi-view network directly used a single reciprocal pair as the auxiliary images. A better approach is to carefully-select a reciprocal pair from multiple reciprocal pairs, perhaps, using a similar view-selection strategy presented in Chapter 4. Multiple reciprocal pairs can be selected but there should be more reciprocal pairs available than the number of selected reciprocal pairs.

One of the potential reasons that was provided in Chapter 4 to explain the poor performance of the deep photometric network on some models is the geometric inconsistency of the models in the dataset. A naive solution is to check every model in the dataset to ensure that they are correct but this is not feasible for a large dataset. The adapted deep multi-view stereo network showed better results than the adapted deep photometric stereo network. An approach to dealing with geometric inconsistency is to leverage the deep multi-view stereo network to train the deep photometric stereo network. By regressing the depth and surface normals of a scene simultaneously, the network learns to generate a globally-consistent view of the scene. A potential limitation is that the hybrid network may not be able to predict a more accurate and complete depth map than a deep Helmholtz multi-view stereo network.

BIBLIOGRAPHY

[1] Altizure. URL https://www.altizure.com/.

[2] Cycles. URL https://www.cycles-renderer.org/.

[3] Htcondor. URL https://htcondor.org/.

[4] Sketchfab. URL https://sketchfab.com/.

[5] Ackermann, J. and Michael, G. . 2015.

[6] Addari, G. and Guillemaut, J.-Y. . Towards Globally Optimal Full 3D Reconstruction of Scenes with Complex Reflectance Using Helmholtz Stereopsis. In *European Conference on Visual Media Production*, CVMP '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450370035. doi: 10.1145/3359998.3369410. URL https://doi.org/10.1145/3359998.3369410.

[7] Aittala, M. , Weyrich, T. , and Lehtinen, J. . Practical SVBRDF capture in the frequency domain. *ACM Trans. Graph.*, 32(4):110–1, 2013.

[8] Barsky, S. and Petrou, M. . The 4-Source Photometric Stereo Technique for Three-Dimensional Surfaces in the Presence of Highlights and Shadows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25:1239– 1252, 11 2003. doi: 10.1109/TPAMI.2003.1233898.

[9] Baumgart, B. G. . Geometric modeling for computer vision. 1974.

[10] Bell, S. , Upchurch, P. , Snavely, N. , and Bala, K. . OpenSurfaces: A Richly Annotated Catalog of Surface Appearance. *ACM Trans. Graph.*, 32(4), jul 2013. ISSN 0730-0301. doi: 10.1145/2461912.2462002. URL https://doi.org/10.1145/2461912.2462002.

[11] Bradley, D. , Boubekeur, T. , and Heidrich, W. . Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. doi: 10.1109/CVPR.2008.4587792.

[12] Campbell, N. D. F. , Vogiatzis, G. , Hernández, C. , and Cipolla, R. . Using Multiple Hypotheses to Improve Depth-Maps for Multi-View Stereo. In *ECCV*, 2008.

[13] Chang, A. X. , Funkhouser, T. , Guibas, L. , Hanrahan, P. , Huang, Q. , Li, Z. , Savarese, S. , Savva, M. , Song, S. , Su, H. , Xiao, J. , Yi, L. , and Yu, F. . ShapeNet: An Information-Rich 3D Model Repository, 2015. URL `https://arxiv.org/abs/1512.03012`.

[14] Chen, G. , Han, K. , and Wong, K.-Y. K. . PS-FCN: A Flexible Learning Framework for Photometric Stereo, 2018. URL `https://arxiv.org/abs/1807.08696`.

[15] Chen, G. , Han, K. , Shi, B. , Matsushita, Y. , and Wong, K.-Y. K. K. . Self-Calibrating Deep Photometric Stereo Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8731–8739, 2019. doi: 10.1109/CVPR.2019.00894.

[16] Chen, G. , Han, K. , Shi, B. , Matsushita, Y. , and Wong, K.-Y. K. . Deep Photometric Stereo for Non-Lambertian Surfaces, 2020. URL `https://arxiv.org/abs/2007.13145`.

[17] Cheng, S. , Xu, Z. , Zhu, S. , Li, Z. , Li, L. E. , Ramamoorthi, R. , and Su, H. . Deep Stereo using Adaptive Thin Volume Representation with Uncertainty Awareness, 2019. URL `https://arxiv.org/abs/1911.12012`.

[18] Choy, C. B. , Xu, D. , Gwak, J. , Chen, K. , and Savarese, S. . 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction, 2016. URL `https://arxiv.org/abs/1604.00449`.

[19] Dong, B. , Moore, K. D. , Zhang, W. , and Peers, P. . Scattering parameters and surface normals from homogeneous translucent materials using photometric stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2291–2298, 2014.

[20] Gallup, D. , Frahm, J.-M. , Mordohai, P. , Yang, Q. , and Pollefeys, M. . Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. doi: 10.1109/CVPR.2007.383245.

[21] Goldman, D. B. , Curless, B. , Hertzmann, A. , and Seitz, S. M. . Shape and spatially-varying brdfs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1060–1071, 2009.

[22] Griffiths, D. and Boehm, J. . A Review on Deep Learning Techniques for 3D Sensed Data Classification. *Remote Sensing*, 11(12):1499, jun 2019. doi: 10.3390/rs11121499. URL https://doi.org/10.3390%2Frs11121499.

[23] Hartt, K. and Carlotto, M. . A method for shape-from-shading using multiple images acquired under different viewing and lighting conditions. In *Proceedings CVPR '89: IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 53–60. doi: 10.1109/CVPR.1989.37828.

[24] Hayakawa, H. . Photometric stereo under a light source with arbitrary motion. *Journal of The Optical Society of America A-optics Image Science and Vision*, 11:3079–3089, 1994.

[25] Ikehata, S. . CNN-PS: CNN-based Photometric Stereo for General Non-Convex Surfaces. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[26] Ikehata, S. and Aizawa, K. . Photometric stereo using constrained bivariate regression for general isotropic surfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2179–2186, 2014.

[27] Jensen, R. , Dahl, A. , Vogiatzis, G. , Tola, E. , and Aanæs, H. . Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.

[28] Jin, H. , Soatto, S. , and Yezzi, A. . Multi-view stereo beyond Lambert. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I, 2003. doi: 10.1109/CVPR.2003.1211351.

[29] Jin, H. , Soatto, S. , and Yezzi, A. . Multi-View Stereo Reconstruction of Dense Shape and Complex Appearance. *International Journal of Computer Vision*, 63:175–189, 07 2005. doi: 10.1007/s11263-005-6876-7.

[30] Johnson, M. K. and Adelson, E. H. . Shape estimation in natural illumination. In *CVPR 2011*, pages 2553–2560, 2011. doi: 10.1109/CVPR.2011.5995510.

[31] Kingma, D. P. and Ba, J. . Adam: A Method for Stochastic Optimization, 2014. URL https://arxiv.org/abs/1412.6980.

[32] Knapitsch, A. , Park, J. , Zhou, Q.-Y. , and Koltun, V. . Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Trans. Graph.*, 36(4), jul 2017. ISSN 0730-0301.

doi: 10.1145/3072959.3073599. URL https://doi.org/10.1145/3072959.3073599.

[33] Krizhevsky, A. , Sutskever, I. , and Hinton, G. E. . ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F. , Burges, C. , Bottou, L. , and Weinberger, K. , editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[34] Laurentini, A. . The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994. doi: 10.1109/34.273735.

[35] Lazebnik, S. , Furukawa, Y. , and Ponce, J. . Projective visual hulls. *International Journal of Computer Vision*, 74(2):137–165, 2007.

[36] Liang, C. and Wong, K.-Y. K. . 3D reconstruction using silhouettes from unordered viewpoints. *Image and Vision Computing*, 28(4):579–589, 2010.

[37] Liu, Y. , Cao, X. , Dai, Q. , and Xu, W. . Continuous depth estimation for multi-view stereo. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2121–2128, 2009. doi: 10.1109/CVPR.2009.5206712.

[38] Mallick, S. P. , Zickler, T. E. , Kriegman, D. J. , and Belhumeur, P. N. . Beyond lambert: Reconstructing specular surfaces using color. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 619–626. Ieee, 2005.

[39] Matusik, W. . *A data-driven reflectance model*. PhD thesis, Massachusetts Institute of Technology, 2003.

[40] Matusik, W. , Buehler, C. , Raskar, R. , Gortler, S. J. , and McMillan, L. . Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 369–374, 2000.

[41] Mersch, S. H. . *Polarized lighting for machine vision applications*. Society of Manufacturing Engineers, 1984.

[42] Mescheder, L. , Oechsle, M. , Niemeyer, M. , Nowozin, S. , and Geiger, A. . Occupancy Networks: Learning 3D Reconstruction in Function Space, 2018. URL https://arxiv.org/abs/1812.03828.

[43] Nayar, S. , Ikeuchi, K. , and Kanade, T. . Determining shape and reflectance of Lambertian, specular, and hybrid surfaces using extended sources. In *International Workshop on Industrial Applications of Machine Intelligence and Vision,*, pages 169–175, 1989. doi: 10.1109/MIV.1989.40544.

[44] Nayar, S. K. , Fang, X.-S. , and Boult, T. . Removal of specularities using color and polarization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 583–590. IEEE, 1993.

[45] Oxholm, G. and Nishino, K. . Multiview Shape and Reflectance from Natural Illumination. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2163–2170, 2014. doi: 10.1109/CVPR.2014.277.

[46] Papadhimitri, T. and Favaro, P. . A New Perspective on Uncalibrated Photometric Stereo. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1474–1481, 2013. doi: 10.1109/CVPR.2013.194.

[47] Pons, J.-P. , Keriven, R. , and Faugeras, O. . Multi-View Stereo Reconstruction and Scene Flow Estimation with a Global Image-Based Matching Score. *International Journal of Computer Vision*, 72:179–193, 03 2007. doi: 10.1007/s11263-006-8671-5.

[48] P.P., R. N. and Jabbar, S. . Efficient 3D visual hull reconstruction based on marching cube algorithm. In *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pages 1–6, 2015. doi: 10.1109/ICIIECS.2015.7193189.

[49] Ronneberger, O. , Fischer, P. , and Brox, T. . U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

[50] Roubtsova, N. and Guillemaut, J.-Y. . Colour Helmholtz Stereopsis for Reconstruction of Dynamic Scenes with Arbitrary Unknown Reflectance. *International Journal of Computer Vision*, 124, 08 2017. doi: 10.1007/s11263-016-0951-0.

[51] Roubtsova, N. and Guillemaut, J.-Y. . Bayesian Helmholtz Stereopsis with Integrability Prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(9):2265–2272, 2018. doi: 10.1109/TPAMI.2017.2749373.

[52] Santo, H. , Samejima, M. , Sugano, Y. , Shi, B. , and Matsushita, Y. . Deep Photometric Stereo Network. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 501–509, 2017. doi: 10.1109/ICCVW.2017.66.

[53] Sato, Y. and Ikeuchi, K. . Temporal-color space analysis of reflection. *JOSA A*, 11(11): 2990–3002, 1994.

[54] Schöps, T. , Schönberger, J. L. , Galliani, S. , Sattler, T. , Schindler, K. , Pollefeys, M. , and Geiger, A. . A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[55] Seitz, S. , Curless, B. , Diebel, J. , Scharstein, D. , and Szeliski, R. . A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528, 2006. doi: 10.1109/CVPR.2006.19.

[56] Shi, B. , Matsushita, Y. , Wei, Y. , Xu, C. , and Tan, P. . Self-calibrating photometric stereo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1118–1125, 2010. doi: 10.1109/CVPR.2010.5540091.

[57] Shi, B. , Tan, P. , Matsushita, Y. , and Ikeuchi, K. . A biquadratic reflectance model for radiometric image analysis. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 230–237. IEEE, 2012.

[58] Vogiatzis, G. and Hernández, C. . Self-Calibrated, Multi-Spectral Photometric Stereo for 3D Face Capture. *Int. J. Comput. Vision*, 97(1):91–103, mar 2012. ISSN 0920-5691. doi: 10.1007/s11263-011-0482-7. URL https://doi.org/10.1007/s11263-011-0482-7.

[59] Wang, J. , Zhang, C. , Zhu, W. , Zhang, Z. , Xiong, Z. , and Chou, P. A. . 3D scene reconstruction by multiple structured-light based commodity depth cameras. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5429–5432, 2012. doi: 10.1109/ICASSP.2012.6289149.

[60] Wang, X. , Wang, C. , Liu, B. , Zhou, X. , Zhang, L. , Zheng, J. , and Bai, X. . Multi-view stereo in the deep learning era: A comprehensive review. *Displays*, 70:102102, 2021. ISSN 0141-9382. doi: https://doi.org/10.1016/j.displa.2021.102102. URL https://www.sciencedirect.com/science/article/pii/S0141938221001062.

[61] Wiles, O. and Zisserman, A. . SilNet : Single- and Multi-View Reconstruction by Learning from Silhouettes, 2017. URL https://arxiv.org/abs/1711.07888.

[62] Woodham, R. . Photometric Method for Determining Surface Orientation from Multiple Images. *Optical Engineering*, 19, 01 1992. doi: 10.1117/12.7972479.

[63] Wu, Z. , Song, S. , Khosla, A. , Yu, F. , Zhang, L. , Tang, X. , and Xiao, J. . 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[64] Xiang, Y. , Mottaghi, R. , and Savarese, S. . Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[65] Xiang, Y. , Kim, W. , Chen, W. , Ji, J. , Choy, C. , Su, H. , Mottaghi, R. , Guibas, L. , and Savarese, S. . ObjectNet3D: A Large Scale Database for 3D Object Recognition. In *European Conference Computer Vision (ECCV)*, 2016.

[66] Yao, Y. , Luo, Z. , Li, S. , Fang, T. , and Quan, L. . MVSNet: Depth Inference for Unstructured Multi-view Stereo, 2018. URL https://arxiv.org/abs/1804.02505.

[67] Yao, Y. , Luo, Z. , Li, S. , Zhang, J. , Ren, Y. , Zhou, L. , Fang, T. , and Quan, L. . BlendedMVS: A Large-scale Dataset for Generalized Multi-view Stereo Networks, 2019. URL https://arxiv.org/abs/1911.10127.

[68] Zheng, Q. , Shi, B. , and Pan, G. . Summary study of data-driven photometric stereo methods. *Virtual Reality & Intelligent Hardware*, 2(3):213–221, 2020. ISSN 2096-5796. doi: https://doi.org/10.1016/j.vrih.2020.03.001. URL https://www.sciencedirect.com/science/article/pii/S2096579620300425. 3D Visual Processing and Reconstruction Special Issue.

[69] Zickler, T. , Belhumeur, P. N. , and Kriegman, D. J. . Helmholtz Stereopsis: Exploiting Reciprocity for Surface Reconstruction. In *Proceedings of the 7th European Conference on Computer Vision-Part III*, ECCV '02, page 869–884, Berlin, Heidelberg, 2002. Springer-Verlag. ISBN 3540437460.