Cuboid-Maps for Indoor Illumination Modeling and Augmented Reality Rendering

by

Kevin Joseph

B.Sc., York University, 2019

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2021

Declaration of Committee

Name:	**Kevin Joseph**
Degree:	**Master of Science**
Thesis title:	**Cuboid-Maps for Indoor Illumination Modeling and Augmented Reality Rendering**
Committee:	**Chair:** Mo Chen Assistant Professor, Computing Science
	Yasutaka Furukawa Supervisor Associate Professor, Computing Science
	Manolis Savva Committee Member Assistant Professor, Computing Science
	Yagiz Aksoy Examiner Assistant Professor, Computing Science

Abstract

This thesis proposes a novel approach for indoor scene illumination modeling and augmented reality rendering. Our key observation is that an indoor scene is well represented by a set of rectangular spaces, where important illuminants reside on their boundary faces, such as a window on a wall or a ceiling light. Given a perspective image or a panorama and detected rectangular spaces as inputs, we estimate their cuboid shapes, and infer illumination components for each face of the cuboids by a simple convolutional neural architecture. The process turns an image into a set of cuboid environment maps, each of which is a simple extension of a traditional cube-map. For augmented reality rendering, we simply take a linear combination of inferred environment maps and an input image, producing surprisingly realistic illumination effects. This approach is simple and efficient, avoids flickering, and achieves quantitatively more accurate and qualitatively more realistic effects than competing substantially more complicated systems.

Keywords: illumination; virtual reality; augmented reality;

Acknowledgements

Thanks to advisor Prof. Yasutaka Furukawa for his guidance, encouragement and insight throughout the course of this research. I would like to thank my examining committee, Prof. Manolis Saava, Prof. Yagz Aksoy, and Prof. Mo Chen for their valuable feedback and advice. Thank you to my colleagues at the VML and GRuVI labs for your support and stimulating discussions. I would like to thank all the staff at SFU for their time and support in various aspects of this journey. Finally, this would not have been possible without my family.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Illumination Estimation is the process of estimating an incident illumination for either a single 3D point, or for an entire scene. The incident illumination is typically represented as a spherical environment map providing a 360 degree source for lighting. Using this illumination one can render a character or object within a scene in a realistic fashion. In this thesis, we focus on a multi-scene illumination modelling approach. We believe our robust illumination method is capable of producing realistic renderings allowing for various applications in virtual remodelling, augmented reality, and virtual reality. In terms of virtual remodeling our method will allow virtual real estate developers and home owners to accurately preview renovations and furniture before purchases are made. In augmented reality (AR) and virtual reality (VR) applications our method will increase the speed and accuracy of objects rendered as well as provide a much larger area for object insertion.

With the growing quantity of data and the advent of deep neural networks (DNNs), the current research trend is designing data driven illumination estimation models using sophisticated network designs. While these methods have shown great potential, they are not without their weaknesses. The first major weakness of most data driven methods is a lack of temporal consistency resulting in flickering artifacts during video generation. Another major weakness is that most methods are trained using regression. As HDR images are bimodal given the presence of a light source, and regression is unimodal, it is not an ideal loss for illumination estimation. The last weakness is that all current methods only model the immediate information in the image via estimating an environment map for either a single point, or the single given scene.

Considering these weaknesses, this thesis proposes a new illumination modeling approach for multi room/space environments. Instead of devising a complex neural architecture, we propose a simple yet carefully designed algorithm with the following key observations: 1) indoor scenes consist of rectangular spaces (e.g., rooms, corridors, or walk-in closets) and major illuminants reside on their boundary faces ; 2) illumination classification instead of regression provides a less bias inference target compared to regression due to estimating HDR values; and 3) temporal inconsistency can be prevented via a simple linear interpolation between a source and a destination environment.

Concretely, given a perspective or a panorama image, space detection is performed. For each detected indoor space, which is typically rectangular, we use a standard CNN to represent it as a

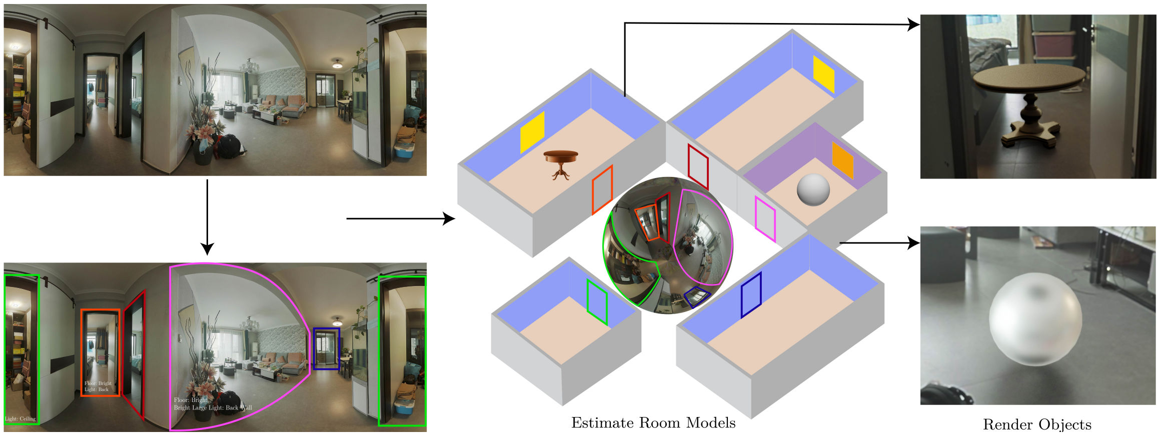


Figure 1.1: Wide angle photography allows for the visibility of multiple indoor spaces. Our method produces cuboid-maps for use in single space or multi-space indoor illumination modeling. These cuboid-maps produce temporally coherent, realistic lighting effects, and once made can be re-projected to any 3D point within the space.

cuboid. For every cuboid we classify their geometric and illumination factors such as its size, the presence of a window on a wall, and ambient intensities. This process turns an image into a collection of "cuboid environment maps".With a scene composed of a collection of cuboid environment maps, and an object of interest to be inserted somewhere in the scene, one requires an incident illumination map to render. In a final step, we project the cuboid map into a spherical illumination image for that location and take the weighted average with the input panorama to obtain an incident illumination. This simple AR rendering algorithm is efficient (i.e., no CNN inference at every frame), consistent without flickering, yet visually plausible.

Our approach is evaluated against state-of-the-art illumination inference algorithms based on a quantitative pixel-wise rendering metric against a pseudo ground-truth as well as a qualitative realism metric via a user study. This simple method outperforms all the existing methods with complex neural architectures, and enables compelling AR experiences in the much wider indoor spaces visible in wide angle or panoramic photography.

In summary, the contribution of this thesis is 3-fold: 1) A unique cuboid environment map for indoor illumination modeling; 2) An efficient, consistent, and compelling rendering algorithm with cuboid maps; and 3) State-of-the-art AR rendering results in multi room/space settings beyond competing methods with complex neural architectures. We believe that our robust illumination inference will have tremendous impacts in virtual remodeling, augmented reality (AR) and virtual reality (VR) applications.

With the introduction and motivation covered, onto the remainder of the thesis. The thesis is structured as follows: in Chapter 2 we introduce all related concepts and existing research on the topic. Then in Chapter 3 we detail the main research on Cuboid-Maps including our data acquisition protocol, the Cuboid-Map modelling method, and a simple neural architecture for cuboid parameter inference. Chapter 4 consists of extensive qualitative and quantitative evaluations of existing baselines and state of the art methods. Then to conclude in chapter 5 we discuss the limitations of this research and possible future directions.

Chapter 2

Related Work

Illumination estimation has been a long-standing problem in both computer vision and graphics. In this section, classic and deep learning approaches for illumination inference, inverse rendering, video tone mapping operators, and finally common panoramic datasets are discussed.

2.1 Illumination Inference

Illumination inference is a long-standing fundamental problem in computer vision and computer graphics. Two methods are common amongst all illumination inference algorithms: one requiring physical access to a scene, and deep learning data driven based methods.

2.1.1 Classic Methods

A classical but effective approach is to place a reflective sphere in a scene to capture an environment map. In terms of classic methods, Debevec et al. [9] rendered virtual objects into real images by using high dynamic range (HDR) environment maps, captured via bracketed exposures of a chrome ball 2.1. Further research explored the use of a diffuse sphere [32] or a metallic/diffuse hybrid sphere [10]. While these approaches are widely adopted in production systems, physical access to scenes is required.

2.1.2 Neural illumination inference

With the emergence of deep neural networks, the current research trend is in designing a passive data-driven approach, inferring a complete scene illumination even from a single image.

Gardner et al. use a CNN to identify the light locations and regress their intensities [14]. They later extend the approach to handle spatially varying illuminations [13]. In their follow up work, they represent illumination using a discrete set of light sources with various geometric and photometric properties. They fit parameters to light sources via an optimization process comparing their renders to the ground truth renders. These parameters are then estimated using deep learning. This method lends itself to complicated issues regarding the quantity and placement of light sources for indoor

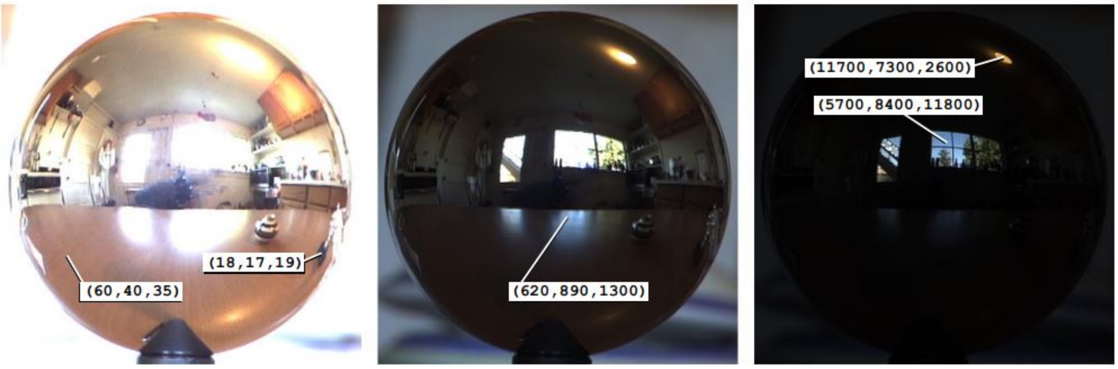


Figure 2.1: This full dynamic range lighting environment was acquired by photographing a mirrored ball balanced on the cap of a pen. (Image Source [9])

scenes. Our method differs as cuboid models provide some geometric assumptions i.e a light source should be on a wall or ceiling, limiting the placement and quantity of light sources.

Neural illumination by Song et al. [35] turns a low dynamic range (LDR) perspective image into a high dynamic range (HDR) environment map. Unlike [14], their pipeline is end-to-end and jointly trained. They estimate HDR environment maps by 1) reprojecting the input image into a panoramic canvas at the position of interest; and 2) using CNN to fill-in the missing pixels 2.2. While being simple and effective, the approach requires the CNN inference at every frame, causing flickering and being computational intensive.

Li et al. estimates an illumination sphere at every pixel only from a single image. This ambitious task requires intensive supervised training data, which comes from fitting spherical gaussians to each pixel within synthetic renderings [25]. This approach also does not account for spatially-varying lighting considering depth. Similar to Neural illumination, it also suffers from flickering artifacts in video generation.

Srinivasan et al. learn to generate a volumetric illumination model from a stereo image pair, where differentiable spherical volume rendering is utilized [37] to create a self-reconstruction loss [37]. These state-of-the-art approaches [37, 25] boast technically intriguing neural architectures. However, we found it hard to reproduce compelling results. Our system is simple, trained via simple supervised learning by exploiting the growing indoor panorama collections, and works well in practice.

2.2 Inverse Rendering

Inverse rendering seeks to infer scene illumination, material, and geometric properties from a single image.

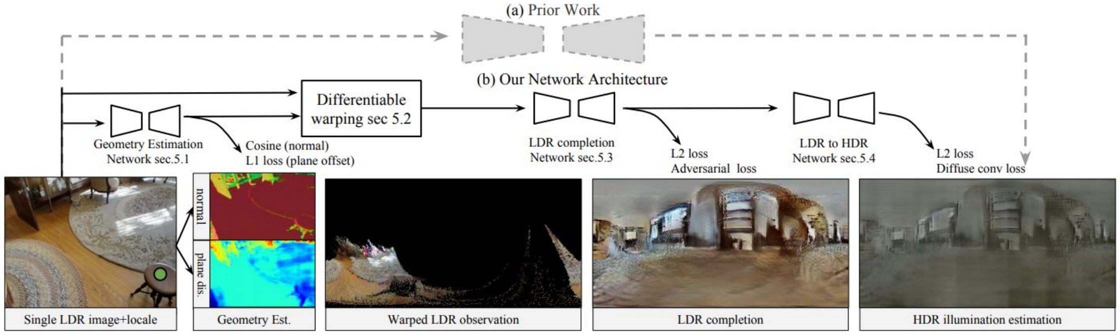


Figure 2.2: Neural Illumination's Pipeline. (Image Source [35])

2.2.1 Classic Methods

Traditional optimization-based approaches require strong statistical prior assumptions and multiple images. Haber et al. [16] and Kim et al. [20] require multi view stereo inputs. Haber et al. attain geometry through multi-view stereo and estimate reflectance and lighting from image collections. Kim et al. also use multi-view stereo to infer geometry, illumination and albedo. In terms of single image inference, Barron et al. [1] propose SIRFS which uses an amalgamation of priors. Jeon et al. [18] note that constraints on classic optimization systems are strict due to rich textures in images. They propose simple constraints for texture-free inputs, which are attained through their texture separate algorithm.

2.2.2 Deep Learning

Deep learning approaches rely on synthetic data and differentiable rendering. Synthetic images are used by [34, 26, 6] to generate ground truth values. Yu et al. [40] use a differentiable renderer to create a self supervised loop via a reconstruction loss. Sengupta et al. . [34] employ a differentiable renderer to perform transfer learning from synthetic to real images. In the proposed cuboid representation, we circumvent the issue of predicting per pixel geometry and illumination by providing a discrete set parameters to classify.

2.3 Intrinsic image decomposition

Intrinsic image decomposition (IID) is the task of inferring shading and reflectance components from an image without explicitly solving for an illumination [2].

Retinex theory and reflectance homogeneity are early promising domain priors [21]. They are based on the observation that sudden changes in intensities come from reflectance and shading. However, the heuristics fail in general scenes. Physics-based rendering on SUNCG dataset [36] was used to train DNN models by Li et al. [24] and Zhou et al. [42]. Liu et al. [27] also use DNN models but take an unsupervised approach by estimating marginal distributions.

One of the key applications of IID is augmented reality rendering, but the quality is inferior to illumination inference approaches, which directly renders 3D object models with the estimated illuminations.

2.4 Video HDR Temporal Consistency

Typically, applying a tone mapping operator individually to each frame of a video leads to temporal artifacts. The two main methodologies that exist to address this issue are preventive and a-priori. Preventive methods focus on loss constraints during training or using a tone mapping operation dependant on the context of the entire image. When rendering an object using an estimated HDR illumination map into a real scene, these methods are not applicable. One would have to plan their training and inference ahead of time to utilize these loss functions to insure temporally consistent predictions. Due to their dependence on the context of the entire image, video tone mapping operators cannot handle the inconsistent predictions that lead to extreme temporal inconsistencies in local sections of the image where the object is present. For these reasons, it is necessary to devise a way to enforce temporal consistency for individual environment map predictions.

2.4.1 Video Tone Mapping

A number of tone mapping operators were modified to contain temporal components that allow them to process HDR video attempting to prevent inconsistencies. Global operators apply the same function to all pixels [12, 17, 38, 28, 5], for example $x^{\frac{1}{\gamma}}$. Local operators [22, 3, 4, 31], apply different tone mapping functions depending on the local neighborhood of each pixel. Both styles of tone mapping are extended over multiple frames in an attempt to prevent temporal inconsistencies.

2.4.2 Loss Function Modification

Deep learning preventive based solutions typically address this issue by modifying the loss. Xu et al. [39] address this issue by training a network to output an entire video at once. By computing a perceptual loss, squared error, and intrinsic loss over the output video, they believe they enforce temporal consistent between predicts across frames. This method is not plausible for rendering pipeline or a video of any reasonable length due to memory constraints. Eilertsten et al. [11] add a regularization term to the loss function. They use a geometric transformation on consecutive frames, which if these are temporally consistent, performing the warping to register the two frames should yield the same result. Given that many illumination estimation methods do not produce videos in advance, these methods are not applicable in this domain. In terms of post processing solutions, Marnerides et al. [29] compose an LDR to HDR pipeline for single images. They note that since the model is not designed for videos, flickering occurs between frames. To alleviate this, they smooth luminance percentile curves. This process is very memory and time intensive. Guthier et al. [15] detect artifacts if the overall brightness difference between successive frames is greater than a threshold. Through an iterative process, the brightness is adjusted until reaching the brightness threshold.

2.5 Panoramic Datasets

There are many existing datasets of indoor panoramas. Unfortunately, some of these datasets are LDR only, while others are composed of synthetic data, which has been shown to not generalize well to real data. Matterport is a panorama RGB-D indoor [7] dataset composed of home scans from a lidar scanner. The dataset includes 90 buildings containing a total of 194,400 RGB-D images, 10,800 panoramas, and 24,727,520 textured triangles. All images are available in HDR. The Laval indoor HDR dataset contains 2100+ high resolution indoor panoramas, captured using a Canon 5D Mark III and a robotic panoramic tripod head [14]. Each capture was multi-exposed (22 f-stops) and is fully HDR, without any saturation [14]. InteriorNet is a massive indoor panoramic dataset composed of 20M images created via moving a camera through synthetic scenes [23]. In this thesis we use a dataset generously provided by Lianjia consisting of scans of 1000 homes. These scans resulted in 16000+ panoramas in 8K HDR. Each panorama was given with it's associated depth, and extrinsic matrix.

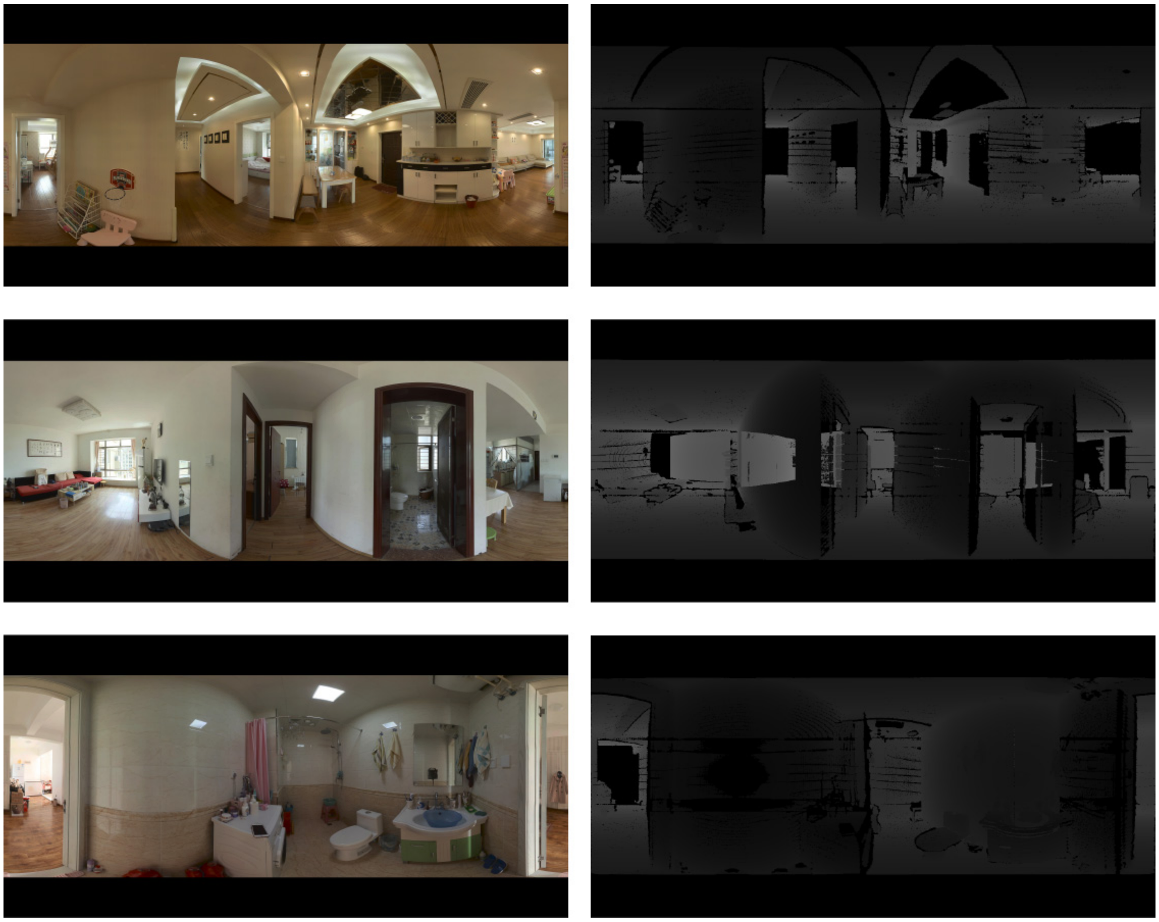


Figure 2.3: Examples of provided panoramas with depth.

Chapter 3

Methods

A panorama of an indoor scene can be viewed as a 360 degree view of that indoor space. Within the scene, one can identity sub spaces. For example, the popular open concept living room and kitchen combo can be divided into a living room, and a kitchen. Each sub space or space itself, is typically a cuboid; each space has four walls, a ceiling, and a floor. From the above, one can conclude that to model the illumination of a space, one needs to define various cuboid parameters, and then fit them using panoramas. With these spaces modeled as cuboids, in this thesis we propose that one can then interpolate between these cuboids to rapidly produce accurate temporally consistent environment maps. The following sections in this chapter detail the cuboid modelling, cuboid fitting, and the rendering processes.

3.1 Cuboid Scene Illumination Modeling

Imagine you are in a living room, looking at a bedroom through a door, for example the right door near the center in Fig. 3.2. It is very difficult to estimate the full surrounding luminance values of that room from this view point. However, even with the partial observation through the doorway, one can tell that the room is small, with a bright window on the left wall, and the ceiling has no light. Following this intuition, we model the geometric and photometric properties of a scene as a set of categorical labels instead of numeric ones. This section explains our categorical cuboid representation and how we collect ground-truth and associate numeric values for the categories.

3.1.1 Categorical Cuboid Representation

A scene is represented by a set of cuboids. Each cuboid (B) consists of the floor, the ceiling, and the four walls ($F_{floor}, F_{ceil}, F_{wall-l}, F_{wall-r}, F_{wall-f}, F_{wall-b}$). We model the following geometric and photometric properties of the geometric components.

$$\begin{aligned}
Size(B) &\leftarrow \{\text{small, medium, large}\} \\
Lpresence(F_{wall-*}, F_{ceil}) &\leftarrow \{\text{yes, no}\} \\
Lsize(F_{wall-*}, F_{ceil}) &\leftarrow \{\text{small, medium, large}\} \\
Lstrength(F_{wall-*}, F_{ceil}) &\leftarrow \{\text{dark, intermed, bright}\} \\
Ambient(F_{*}) &\leftarrow \{\text{dark, intermed, bright}\}
\end{aligned}$$

The cuboid, like any space, has an associated size. The majority of light sources in a space do not come from the floor, therefore we determine the presence of a light source on the walls and ceiling only. Given the presence of a light source, one must determine its strength and size. Ceiling lights and windows exhibit large differences in size. Thus, we model their sizes differently, with the ceiling categories being much smaller A.6. Light strength does not vary between ceilings and walls. As luminance is the light emitted, or reflected by a surface, each face also has an ambient term.

3.1.2 Cuboid Ground-truth collection

Any panorama can be modelled as a cuboid. Simple data statistics are used to convert continuous numeric values into categorical labels for the various parameters of the cuboid.

3.1.2.1 Cuboid Size

Not all spaces are initially cuboids. In this work, we convert any 3-D polygonal shape into a cuboid to determine its width, depth, and height. To determine the width and depth, we utilize floor plans to generate bounding boxes over the extremities of each space. The height component is determined via 2d to 3d reconstruction given ground truth depth. With a set of values in our training dataset, we determine three representative values for classification as follows. We generate a discrete set of candidate values. For every triplet of candidate values, for each training value, we compute the binning error by the L1 distance from its closest candidate value. The average binning error across all training values is the score for the triplet. We simply use the triplet candidate values with the smallest error for the classification.

3.1.2.2 Light Parameters

We attain wall segmentations via projecting their locations into panoramas from floor plans. Using all panoramas on a floor, and their associated depths, we create a representation of that floor in 3D. We orthographically project this 3D representation to create a floor plan. As the ground truth floor plans are aligned with this orthographic projection in the world coordinate frame, one can use backwards projection to segment walls. We iterate over all 4 walls of each room projecting them into the panorama. Given all walls of the room, we segment the ceiling as the remaining upper half of the pixels, and the floor as the lower half. This process can be seen in 3.1. We detect lights in

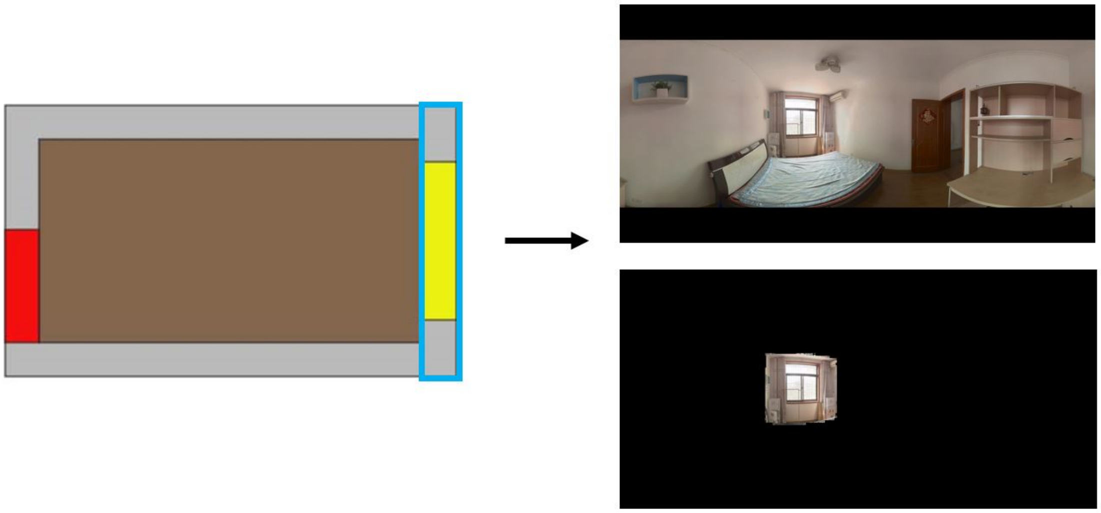


Figure 3.1: As the floor plans are aligned with an orthographic projection of the house models in the world coordinate frame, one can use backwards projection to segment walls.

HDR images using the non-maximal suppression method from [13]. Following [13], we first extract the global maximum value of the panorama. If this max is larger than 15000, we search for a light. To do this, we apply start by analyzing the 3 by 3 patch centered on the global maximum. We first check if at least one value within the patch is at least 80% of the global maximum. If a suitable value exists, the entire patch is set to 0, and the patch grows by 1 in each dimension i.e to 4x4, 5x5, etc... All pixels above the threshold are considered part of the light source. This process is repeated until no value is within 80% of the global maximum. If at least 1.5% of the pixels are selected, we consider that location a light source. We use the average of the detected light values to represent the lights' strength with an HDR value. To determine a scale agnostic representation of size, we place a bounding box around the extremities of the segmented pixels and represent the size of the light source as a percentage of the wall.

3.1.2.3 Ambient Term

Using the same wall segmentations as detailed in the previous section, we removed values over a threshold of 15000 (HDR images range from 0 to 45684) to prevent light sources from contributing to the ambient parameter. The ambient term of each location is then calculated as the average of its pixels.

3.1.2.4 Parameter Bins for Classification

In this section, we detail the optimized discrete parameter values for classification which can be found in 3.1. Note that for the largest light strength and medium light size, we picked parameters that improved the quality of the final renderings by visual inspection as we were not satisfied with the optimal parameters.

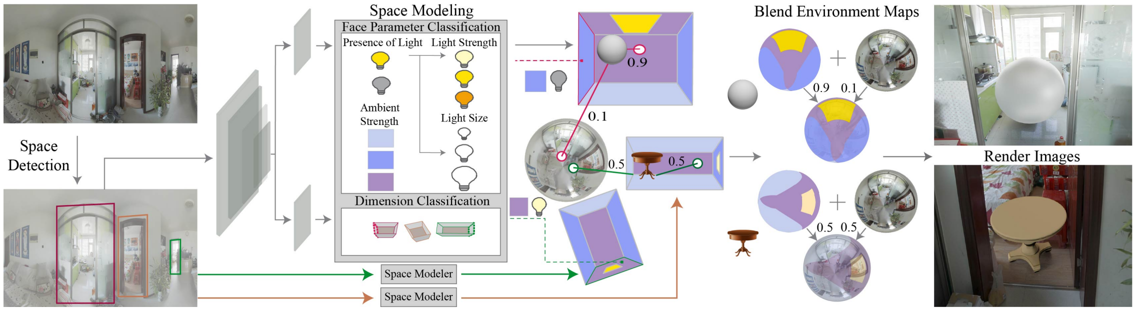


Figure 3.2: Given an input panorama or perspective image, we assume space detection has been performed. Each detection is then sent through a Space Modeller to predict the various parameters for each face of the cuboid, as well as the cuboid's dimension. We orient the cuboids relative to the given image. With an object of interest to be inserted at a designated 3D position, we project the associated cuboid map into a spherical illumination image for this point. We then take the weighted average with the input panorama to obtain a final environment map to be used for rendering.

Table 3.1: Parameter Bins for Classification

	Min	Mid	Max
Cuboid Dimension	(2.5, 2)	(4.5, 3)	(9, 5)
Ambient	250	1250	3500
Light Strength	14 000	22 000	42 500
Wall Light Size	5%	50%	92.5%
Ceiling Light Size	2.5%	17.5%	50%

With the cuboid model defined and the fitting processed detailed, we now require data to fit to. The following section details the data generation protocol.

3.2 Dataset Generation

In this work Lianjia generously provided scans of 1000 homes. These scans resulted in 16000+ panoramas in 8K HDR. Each panorama was given with it's associated depth, and extrinsic matrix. Depth annotations were generated through the use of a LiDAR camera. Typically reflective surfaces do not produce depth readings. In the provided dataset, surfaces which produce depth readings are placed in front of those that do not. This provides a dense depth map for each panorama. Each home also comes with annotated floorplans, in which the annotations include any passageway between spaces, space labels, and window locations. Using a floorplan, we detect visible pairs of panoramas spanning different spaces, as well as annotate the location of the passageway between spaces in the source panorama. Visibility testing is performed using depth values as follows: given a destination pixel in the source image, P_{Dest}, and the destination panorama location, D_{Loc}, in the world coordinate frame, depth values are used to project the designated pixel to world $P_{Dest} \longrightarrow P_{World}$. If

$(P_{World} - D_{Loc})^2 \leq 0.05$ meters these panoramas can see each other, and thus are paired. Each visible panorama is considered an individual space and modelled as a cuboid. This process is detailed in 3.3. Examples of detected spaces can be seen in 3.4. This creates 13058 pairs. This is split this into 11750 pairs for training and 1308 pairs for testing.

With sufficient training data, and the cuboid model, one must now be able to infer the model for unseen examples. The next section covers cuboid model inference.

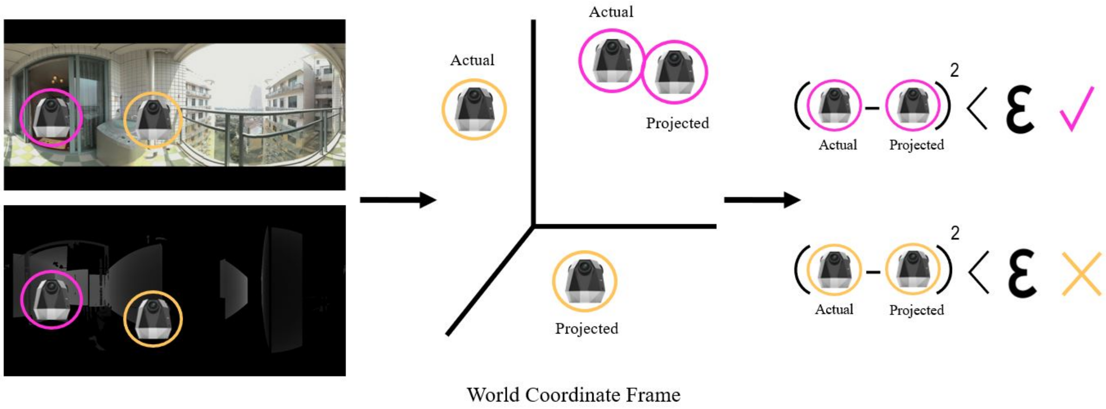


Figure 3.3: Using floor plans panos in adjacent spaces are paired. The passage way between the spaces are annotated. Visibility testing via depth values is performed to insure at least one panorama is visible in the destination room.

3.3 Cuboid Model Inference

In order to build a space composed of cuboid environment maps, we use ground truth bounding boxes of cuboid spaces, then classify their respective geometric and photometric properties. Space detection is reserved for future work.

3.3.1 Space Detection During Inference

Space detection is performed by projecting any labelled passageway between spaces on the given floor plans into the panorama. Panoramas are paired based on their visibility through these passageways accomplished through depth testing. We did not have time to train a space detection network, therefore we use floor plans to generate all space detections. Since various object detection frameworks exist [33, 30] and can be generalized to detect spaces given sufficient data, we do not cover their performance in this work and leave this for future work.

3.3.2 Cuboid Inference

We use Resnet-18 as a backbone network. Resnet is composed of four stages, then an average pooling operation, then a fully connected layer to perform classification. After the average pooling, we replace the single fully connected layer with twelve separate fully connected layers. Each of these layers accepts a 2048 feature vector. Six of these layers make predictions for the ambient value for each face in the cuboid. The remaining five predict the presence of a light for all faces but the one representing the floor. Any positive prediction for the presence of a light is then passed into another unique two fully connected layers for said face. These predict the strength and size of this light source. This set of layers can be seen in 3.2 as the face parameter classification subset of the space modeller. The twelfth and final layer predicts the space dimensions, seen in 3.2 as the dimension classification subsection of the space modeller. This work was implemented in Pytorch. We utilized a work-station with dual Xeon CPUs and dual NVIDIA Titan RTX GPUs. The network is trained with a batch size of 20, for 50 epochs. The ADAM optimizer is utilized with a standard cross-entropy loss. The initial learning rate is 0.0002 and decays by 50% every 15 epochs.

3.4 Rendering and Illumination Interpolation

After attaining all cuboids present in the given image, one must formulate the space of cuboid environment maps to be used for illumination estimation.

3.4.1 From Cuboid to Environment Map

During inference one does not have access to floor plans. As the input is an image centered on the passageway into another room, we use the depth value in the center of the image, at the base of the passageway. Given extrinsic camera parameters and this depth value, one can orient a cuboid in the world coordinate frame. The $(width, height, depth)$ dimensions are then used to create a cuboid relative to the determined position. Each face is meshed to a certain resolution and is given it's respective ambient parameter as a constant. If a light has been determined present, it is placed in the center of the plane consuming the percentage of the wall it has been assigned using the average aspect ratio of lights over the dataset, 1.14. The entire area is set to the predetermined or predicted strength value. Our representation can now be projected to anywhere in the destination room using the proper extrinsic matrix to produce an environment map at that 3D location. Examples of spaces composed of cuboids can be seen in 1.1.

3.4.2 Illumination Interpolation

Individual predictions suffer from spatial inconsistencies producing wildly varying light values for small movements. This results in flickering in videos between frames producing a severe loss of realism. With a source panorama, an object, a final destination for said object, and a prediction environment map for that location, we propose that one linearly interpolates between the source panorama and the final destination to completely remove temporal inconsistencies. The interpolation blending weight is derived via computing the object's euclidean distance to the source panorama and the final destination. These distances are inverted, summed, and the percentage of the total inverted distance is the linear blending parameter assigned to each panorama 3.2. This system is highly effective and efficient preventing a user from having to make per frame predictions.

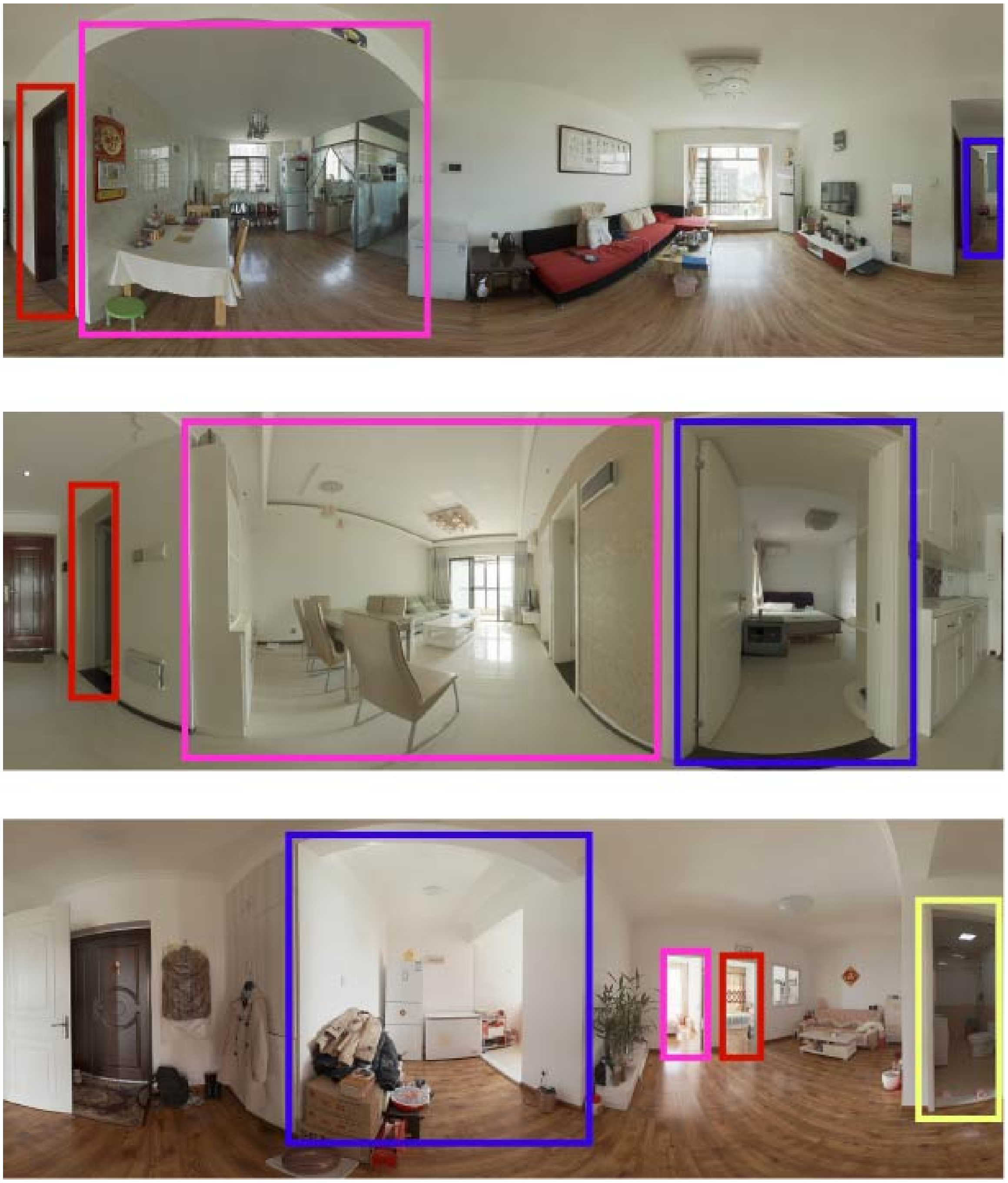


Figure 3.4: Space Detection Results Pre-Visibility Testing.

Chapter 4

Experiments

Our Cuboid-Maps are evaluated against three baselines on the generated dataset. This chapter outlines the baseline models, qualitative, and quantitative evaluations.

4.1 Baseline Models

Our method is compared to Srinivasan et al. [37], Li et al. [25], and Song et al. [35]. For Neural Illumination [35] we provide ground-truth geometry. In addition, 3-D to 2-D rasterization is used to insure the projection to the destination location is as dense and accurate as possible. To do this we first project the 3D triangular faces into the panorama using perspective projection. Then, we loop over all pixels in the panorama and test whether they lie within the resulting 2D triangle. If they do, we fill the pixel with the triangle's color. Our final modification to Neural Illumination is that we replace their in-painting U-Net with the more recent PEN-Net [41] due to it's proven improved performance in in-painting versus U-Net. This method will be referred to as Neural Illumination ++ in the results. Results from rasterization versus the naive projection can be seen in 4.1.

Neural Illumination ++ [35] and Lighthouse [37] were both retrained on our data. We were incapable of retraining Li et al. [25] due to it's dependency on synthetic data to produce ground truth spherical gaussians per pixel. As UCSD and lighthouse were trained with perspective LDR image inputs, the inputs for all methods are perspective LDRs. All of the quantitative metrics are computed on HDR images. In the event a method produces LDR output, we used a known inverse-tone mapping operator, $x^2.1$, to produce a proper HDR output. It should be noted that our method can be trained with LDR or HDR inputs, as well as panoramic or perspective inputs.

4.2 Quantitative Evaluation

4.2.1 Network Inference

The results of the network inference can be found in table 4.1. The network is highly capable of predicting room size achieving an accuracy of 97%. Presence of light detection accuracy is consistent across all faces. Light strength predictions are highest for back walls, which when windows

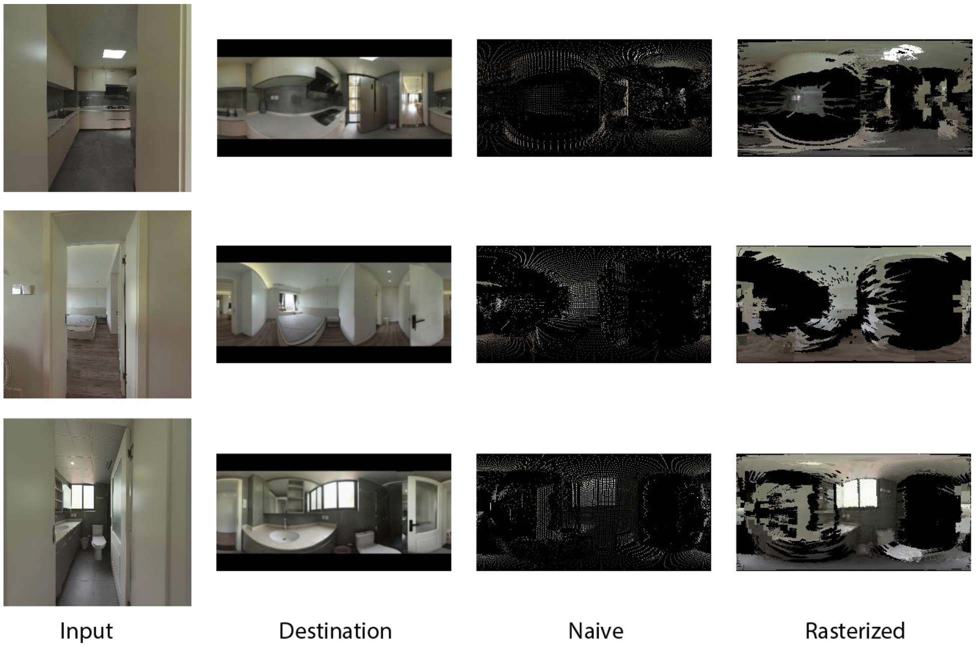


Figure 4.1: Results from rasterizing instead of using naive forward projection for Neural Illumination ++.

are present, are usually visible through a spatial passageway. Ceiling light strength values are the lowest, while the size detection results are the highest, both cases are likely due to their frequent small size. Ambient value detection is consistent across all faces except the floor. We believe this is because there is larger variability in floor values due to glare / strong highlights from windows and ceiling lights on the floor.

4.2.2 Comparison to Ground Truth Renders

As the environment maps produced via cuboids are not designed to be an exact replica of the ground truth environment map, we have decided to compare against other methods via computing the MSE between each method's render and the ground truth's render. Table 4.2 shows that the cuboid model outperforms all other baselines when rendering only the object by a significant margin. When considering the entire rendered scene, our method also outperforms all other baselines. We believe this is because of two reasons. First, when given partial information of the desired space, it is much simpler to predict per face values for illumination rather than per pixel. Second, we choose classification instead of regression. This decreases the model's risk of predicting large HDR values for a light's strength common with regression. Since the model is an approximation, in 4.2 one can find the lowest error achievable. This error, while not capable of becoming 0 as other methods that directly predict environment maps, is rather low. In defense of cuboid modelling, any method which predicts a re-usable approximation, or an approximation of illumination for a space, will always have some error versus a method that predicts per point illumination.

Table 4.1: Cuboid Parameter Classification Accuracy out of 100.

Position Parameter	Front	Back	Left	Right	Ceiling	Floor	Size
Cuboid Size	-	-	-	-	-	-	97
Presence of Light	88	86	88	87	86	-	-
Light Strength	70	76	71	70	60	-	-
Light Size	86	78	79	81	91	-	-
Ambient	87	85	87	86	85	74	-

Table 4.2: MSE computed on the renders VS the ground truth render

	Lighthouse	UCSD	Neural Illum ++	Ours	Ours GT
Object Only	1.98	1.64	0.58	0.31	0.29
Entire Image	0.078	0.053	0.025	0.017	0.014

4.2.3 User Study

Given a real image of an indoor scene, we insert an object at two points in the scene that should display a large difference in lighting. We render the object using predictions for each baseline method. Users are shown three sets of results: two methods, and the ground truth render. Users are asked to pick which is more realistic, or whether the methods produce similar results, with the ground truth as a reference. An example question can be seen in 4.2.

The scenes were categorized into either *Easy* or *Hard* cases. An *Easy* case, is a case in which a single, or multiple light sources is visible in the input image. Example *Easy* cases can be seen in the first two scenes in 4.4. In *Hard* cases the perspective input does not directly show any light source, but does display hints via specular highlights available in the scene. An example *Hard* case can be seen as the third scene in 4.4. Scores for each method were computed as +1 if preferred, 0 if similar, and -1 if not preferred. The average of all scores was taken. A score of 1.0 indicates that method was preferred 100% of the time, 0 both methods were similar in all cases, and -1.0 indicates the method was never preferred. In 4.3 one can see that in both categories of comparisons against all methods, our method is preferred. Neural Illumination ++ takes second place outperforming UCSD and Lighthouse. UCSD performs the worst in all comparisons. This is likely because of their inflexible model requiring extensive synthetic data for training not allowing for any form of retraining on our real dataset. For the easy category our preference rates are high against all baselines. This indicates that, given the visibility of a light source in an input image, our method is more consistent in properly modeling it. We believe this is because we choose classification instead of regression, decreasing the model's risk of predicting large HDR values for a light's strength common with re-

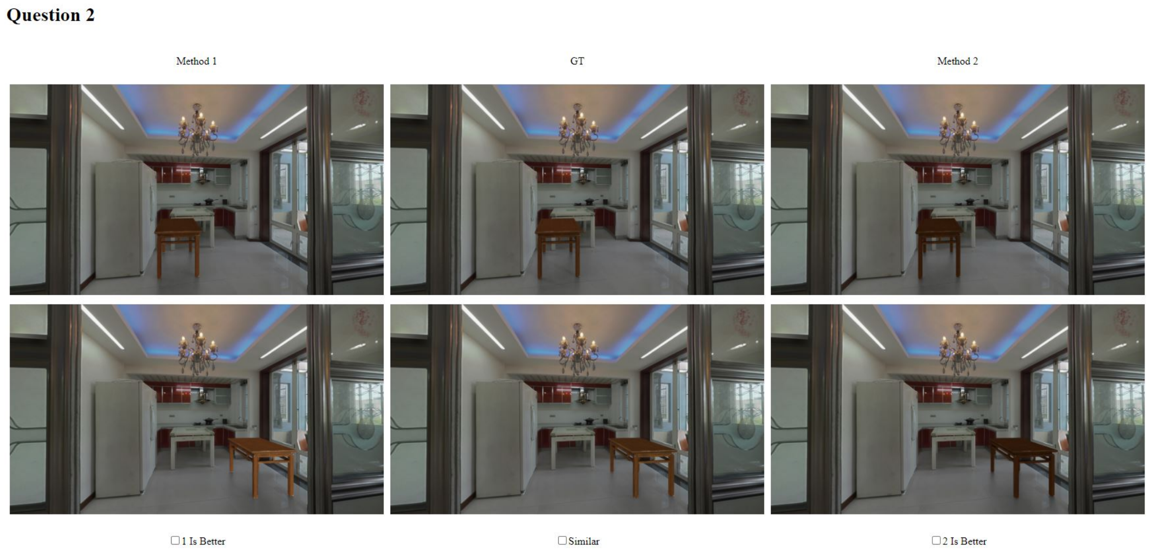


Figure 4.2: User study example question.

gression. The results in the *Hard* category suggest our illumination modeling strategy considering more environment cues to predict the presence of unseen light sources.

4.3 Qualitative Evaluation

4.3.1 Static Renders

In 4.4 We insert an object at two points in the scene that should display a large difference in lighting. We render the object using predictions for each baseline method. For the first pair of positions one can see that all baseline methods are darker closer to the light source, while mine, like the ground truth, is brighter. The second pair of positions shows that both UCSD and Lighthouse produce predictions that are composed of bright ambient values only. While Neural Illum ++ does produce specular highlights on the object, they are to a much lesser degree than those produced by ours and the ground truth. In the third example one can see that our method predicts the presence of an unseen light.

Investigating produced environment maps is a pivotal part of illumination estimation. In 4.5 one can see that our environment maps do an excellent job of spatially positioning light sources in a space. All examples display that our method produces more intense specular highlights versus all other methods, indicating that not only does it position lights properly, it models their strength and size accurately as well. The first, second, and third examples show that our model also considers that the ambient contribution of a plane with a sufficiently strong light source is higher. In terms of baselines, the figure shows that UCSD predicts a very low resolution environment map. These lower resolution maps produce less sharp shadows and lighting on objects. When light sources are

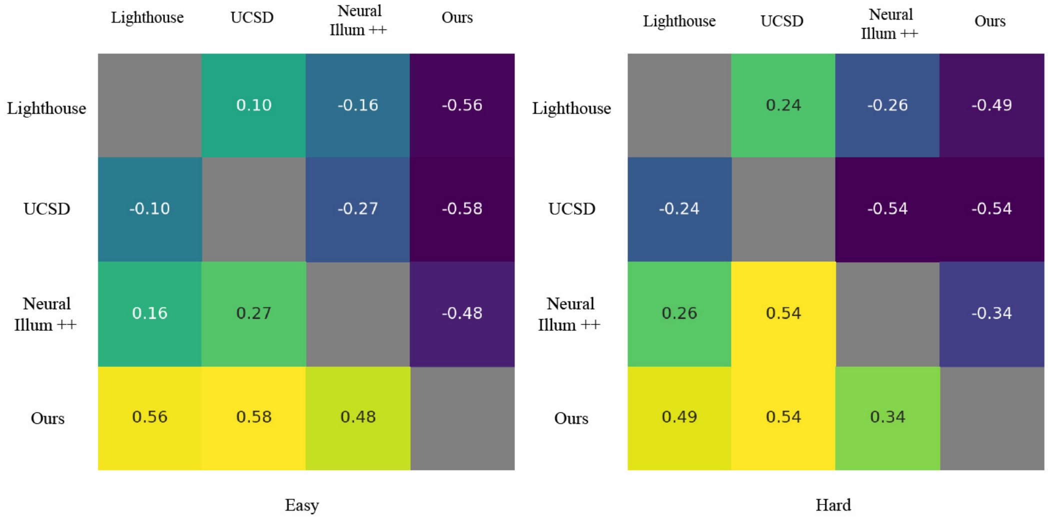


Figure 4.3: The user study scores preferences for each pair of methods in Easy situations in which a single or multiple light sources are visible, and Hard situations in which no light sources are immediately apparent. The tables should be read row-by-row. For example, the bottom row shows the results of Our method against the other methods. A score of 1.0 indicates that method was preferred 100% of the time, 0 the methods are completely similar, and -1.0 the method was never preferred.

visible, all methods place them reasonably well. In the fourth example in which the light source is not visible, it's existence and localization in the environment map suffers in all baselines.

4.3.2 Illumination Interpolation

Per pixel predictions are typically temporally incoherent producing flickering artifacts when creating videos. As seen in 4.6 one can see the effectiveness of our illumination interpolation to avoid such artifacts. In all videos beyond this example, the flickering is completely eliminated producing a much higher quality product, while simultaneously being faster than per pixel predictions.

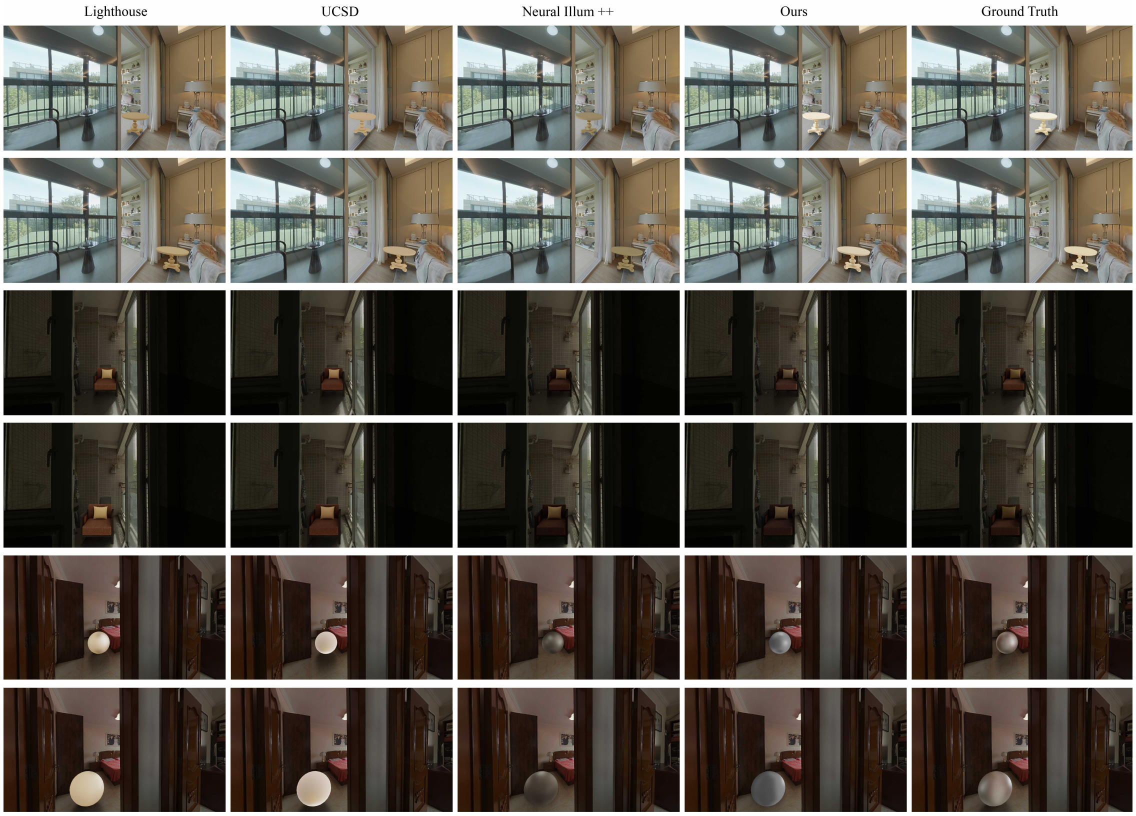


Figure 4.4: Renders given two points of object insertion. In all cases you can see that our method produces more specular highlights properly representing the spatial position of the object relative to the light source.

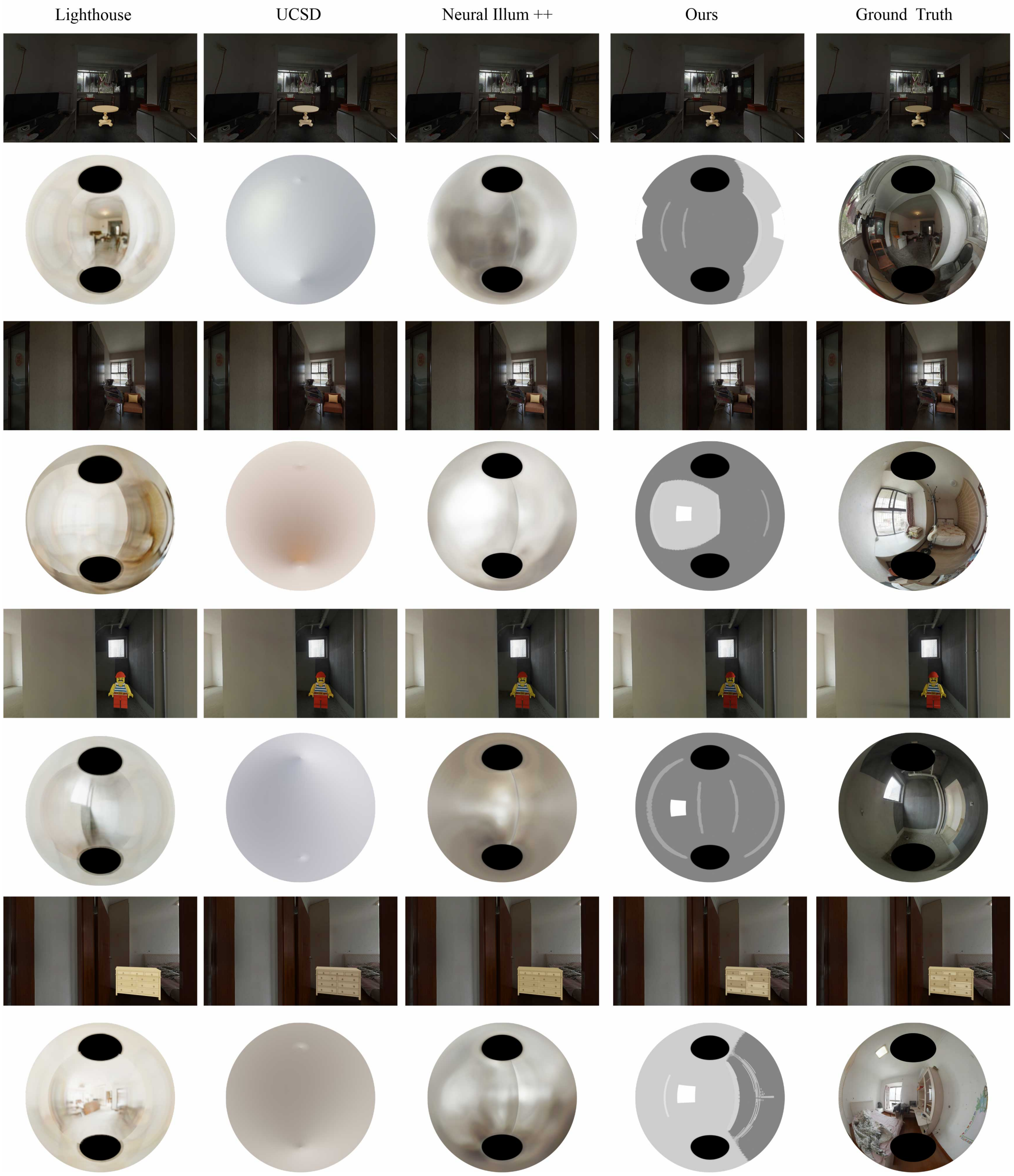


Figure 4.5: Renders and environment maps produced by each method. Our method clearly displays better spatial awareness placing light sources at the correct position within the scene. In cases other methods do get the correct light placement, the regressed values are not high enough to create the same illumination effects produced by the ground truth or our model.

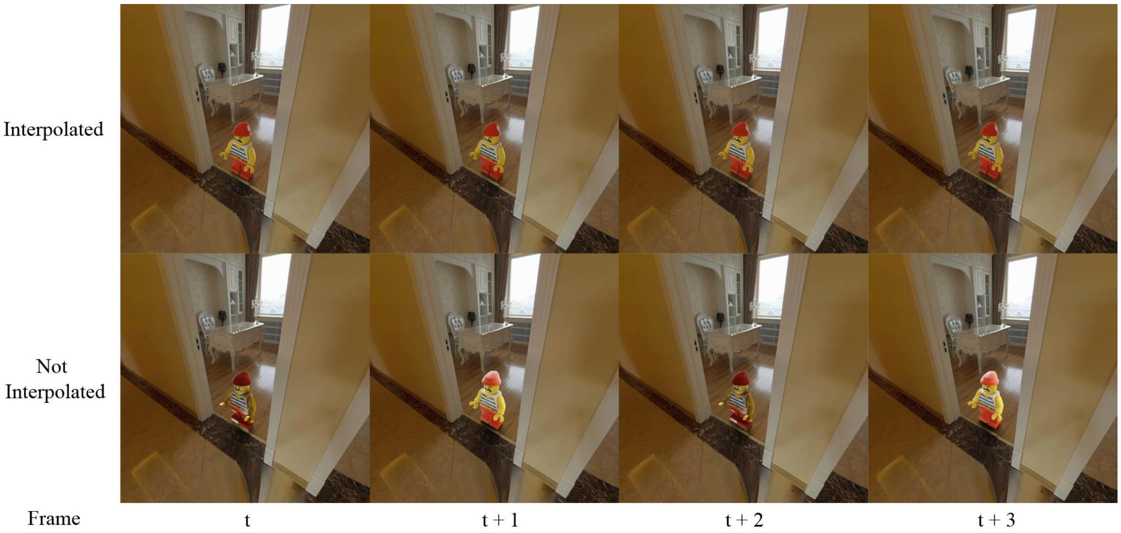


Figure 4.6: Interpolating results when using per pixel eliminates the production of any flickering artifacts

Chapter 5

Conclusion and Future Work

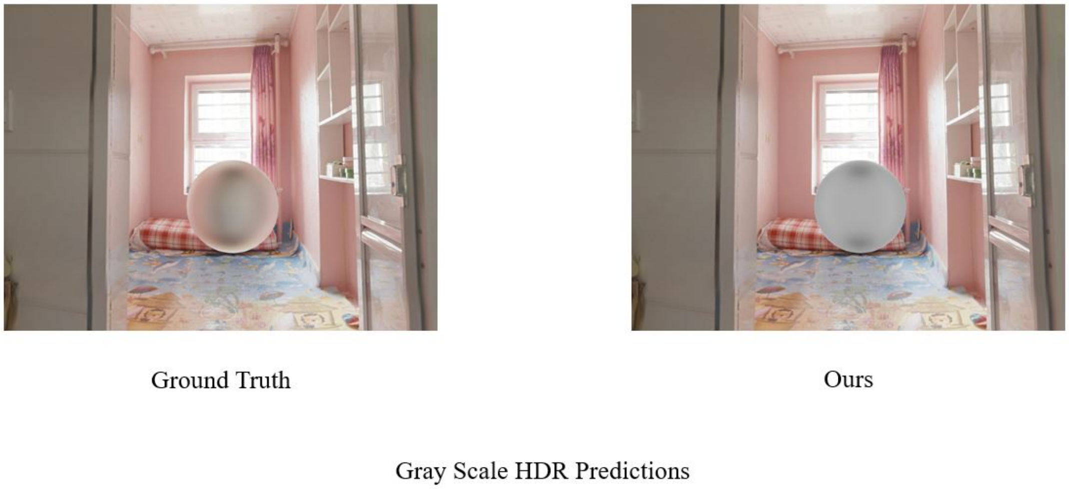
This thesis introduces a cuboid structure for illumination estimation, and proposes a simple method of using linear interpolation to prevent flickering for any illumination inference system. In this section we will present the limitations of this work, suggest future work, and finally, conclude the thesis.

5.1 Limitations

No method is perfect. One of the limitations of the current approach is that it produces gray scale intensity values, resulting in unnatural looking renders under heavily colored light sources 5.1. Our environment maps also do not replicate the ground truth environment maps resulting in a loss of realism when rendering highly specular or metallic objects. In terms of cuboid construction, we position the light sources in the center of the cuboid faces, at a fixed aspect ratio. Thus, when light sources are higher on the wall, our renders may look unnatural depending on the strength of the light. While we claim classification loss is much better than regression for the task of illumination estimation, it is not without it's weaknesses. One weakness of using a classification approach is that outliers have no chance of being modelled properly, especially in terms of their spatial dimensions. Lastly, we do not perform space detection.

5.2 Future Work

The backbone network used was very simple. Future work could explore using either an entirely different backbone network or facial based attention to increase our classification results. Extensions could also include positioning light sources vertically and horizontally on a face, rather than in the center. Additionally, one could model 3D objects in the cuboid to insure light is properly occluded. Lastly, implementing an automated space detection system would make a massive difference: one could model an entire area with a single panorama input. This will be the main focus of future work.



Ground Truth

Ours

Gray Scale HDR Predictions

Figure 5.1: Our method only outputs gray scale HDR values. This creates unrealistic renders when in a heavily colored area.

5.3 Conclusion

This thesis introduces a cuboid structure for illumination estimation, and proposes a simple method of using linear interpolation to prevent flickering for any illumination inference system. The proposed cuboid map illumination modelling is classification based avoiding pitfalls associated with L2 loss. Our method also uses only a single network to estimate illumination while all other's methods covered use at least 3. Quantitatively, the inference model effectively classifies faces of the cuboid allowing for our qualitative and quantitative evaluations to demonstrate consistent improvements over the existing techniques. In both Easy and Hard cases our method outperforms all other baselines according to our user study. Our performance in the hard category indicates that when given partial information of the desired space, it is much simpler to predict per face values for illumination rather than per pixel. In terms of matching the pseudo ground truth renderings, our method also significantly out performs all baselines. To conclude, we believe the cuboid structure often produces feasible illumination maps whether or not lights are observed directly in the input that can be rapidly reprojected to produce a variety of high quality renders.

Bibliography

[1] Jonathan T. Barron and Jitendra Malik. Shape, illumination, and reflectance from shading, 2020.

[2] H.G. Barrow. Recovering intrinsic scene characteristics from images. *Computer Vision Systems*, pages 3–26, 1978. cited By (since 1996) 143.

[3] Eric P. Bennett and Leonard McMillan. Video enhancement using per-pixel virtual exposures. *ACM Trans. Graph.*, 24(3):845–852, July 2005.

[4] Alexandre Benoit, David Alleysson, Jeanny Herault, and Patrick Callet. *Spatio-Temporal Tone Mapping Operator Based on a Retina Model*, page 12–22. Springer-Verlag, Berlin, Heidelberg, 2009.

[5] Ronan Boitard, Kadi Bouatouch, Remi Cozot, Dominique Thoreau, and Adrien Gruson. Temporal coherency for video tone mapping. In Andrew G. Tescher, editor, *Applications of Digital Image Processing XXXV*, volume 8499, pages 113 – 122. International Society for Optics and Photonics, SPIE, 2012.

[6] Mark Boss, Varun Jampani, Kihwan Kim, Hendrik P. A. Lensch, and Jan Kautz. Two-shot spatially-varying brdf and shape estimation, 2020.

[7] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[8] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[9] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, page 189–198, New York, NY, USA, 1998. Association for Computing Machinery.

[10] Paul Debevec, Paul Graham, Jay Busch, and Mark Bolas. A single-shot light probe. In *ACM SIGGRAPH 2012 Talks*, SIGGRAPH '12, New York, NY, USA, 2012. Association for Computing Machinery.

[11] Gabriel Eilertsen, Rafal K. Mantiuk, and Jonas Unger. Single-frame regularization for temporally stable cnns. *CoRR*, abs/1902.10424, 2019.

[12] James A. Ferwerda, Sumanta N. Pattanaik, Peter Shirley, and Donald P. Greenberg. A model of visual adaptation for realistic image synthesis. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 249–258, New York, NY, USA, 1996. Association for Computing Machinery.

[13] Marc-Andre Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagne, and Jean-Francois Lalonde. Deep parametric indoor lighting estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[14] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *CoRR*, abs/1704.00090, 2017.

[15] Benjamin Guthier, Stephan Kopf, Marc Eble, and Wolfgang Effelsberg. Flicker reduction in tone mapped high dynamic range video. volume 7866, pages 78660C:01 – 78660C:15, 01 2011.

[16] T. Haber, C. Fuchs, P. Bekaer, H. Seidel, M. Goesele, and H. P. A. Lensch. Relighting objects from image collections. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 627–634, 2009.

[17] Piti Irawan, James A. Ferwerda, and Stephen R. Marschner. Perceptually based tone mapping of high dynamic range image streams. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, EGSR '05, page 231–242, Goslar, DEU, 2005. Eurographics Association.

[18] Junho Jeon, Sunghyun Cho, Xin Tong, and Seungyong Lee. Intrinsic image decomposition using structure-texture separation and surface normals. pages 218–233, 09 2014.

[19] Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig, and David Forsyth. Automatic scene inference for 3d object compositing. *ACM Trans. Graph.*, 33(3), June 2014.

[20] Kichang Kim, A. Torii, and M. Okutomi. Multi-view inverse rendering under arbitrary illumination and albedo. In *ECCV*, 2016.

[21] Edwin Land and John McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61:1–11, 02 1971.

[22] Patrick Ledda, Luis Paulo Santos, and Alan Chalmers. A local model of eye adaptation for high dynamic range images. In *Proceedings of the 3rd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, AFRIGRAPH '04, page 151–160, New York, NY, USA, 2004. Association for Computing Machinery.

[23] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *British Machine Vision Conference (BMVC)*, 2018.

[24] Zhengqi Li and Noah Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–387, 2018.

[25] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2475–2484, 2020.

[26] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Trans. Graph.*, 37(6), December 2018.

[27] Yunfei Liu, Yu Li, Shaodi You, and Feng Lu. Unsupervised learning for intrinsic image decomposition from a single image, 2020.

[28] Rafał Mantiuk, Scott Daly, and Louis Kerofsky. Display adaptive tone mapping. *ACM Trans. Graph.*, 27(3):1–10, August 2008.

[29] Demetris Marnerides, Thomas Bashford-Rogers, Jonathan Hatchett, and Kurt Debattista. Expandnet: A deep convolutional neural network for high dynamic range expansion from low dynamic range content. *CoRR*, abs/1803.02266, 2018.

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[31] Erik Reinhard, Tania Pouli, Timo Kunkel, Ben Long, Anders Ballestad, and Gerwin Damberg. Calibrated image appearance reproduction. *ACM Trans. Graph.*, 31(6), November 2012.

[32] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[34] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W. Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image, 2019.

[35] Shuran Song and Thomas A. Funkhouser. Neural illumination: Lighting prediction for indoor environments. *CoRR*, abs/1906.07370, 2019.

[36] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1754, 2017.

[37] Pratul P. Srinivasan, Ben Mildenhall, Matthew Tancik, Jonathan T. Barron, Richard Tucker, and Noah Snavely. Lighthouse: Predicting lighting volumes for spatially-coherent illumination, 2020.

[38] J. H. Van Hateren. Encoding of high dynamic range video with a model of human cones. *ACM Trans. Graph.*, 25(4):1380–1399, October 2006.

[39] Y. Xu, L. Song, R. Xie, and W. Zhang. Deep video inverse tone mapping. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, pages 142–147, 2019.

[40] Ye Yu and William A. P. Smith. Inverserendernet: Learning single image inverse rendering, 2018.

[41] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. *CoRR*, abs/1904.07475, 2019.

[42] Hao Zhou, Xiang Yu, and David W Jacobs. Glosh: Global-local spherical harmonics for intrinsic image decomposition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7820–7829, 2019.

Appendix A

Appendix

A.1 Rendering Software: Blender

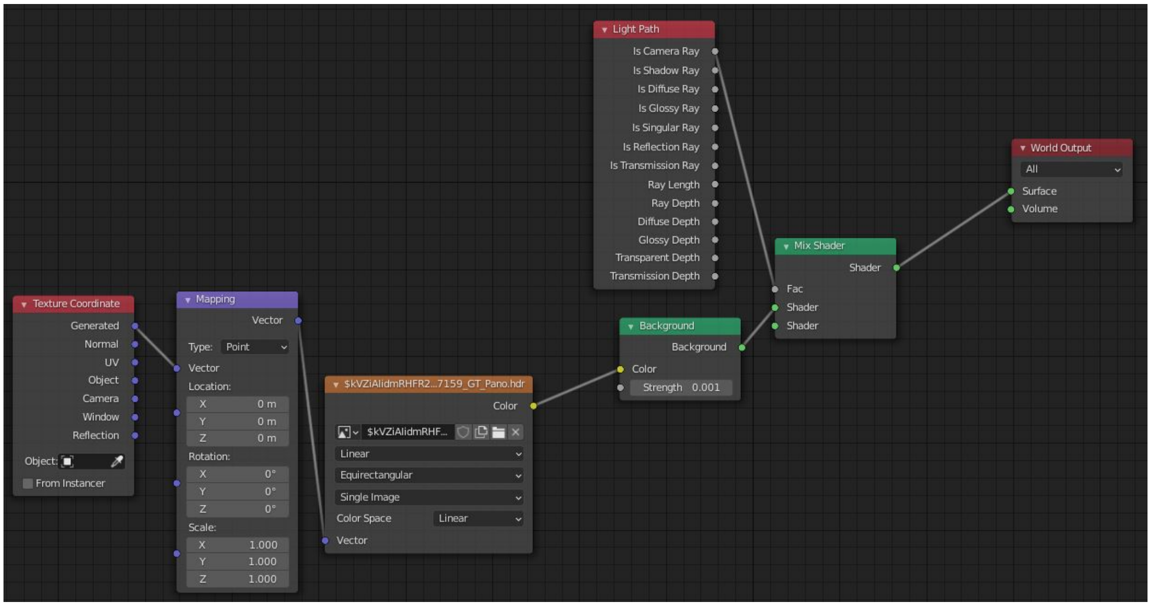
In this thesis all renderings were created with Blender 2.90a. Blender provides two methods of rendering, a real time engine, Eevee, and a slower more advanced engine, Cycles. Cycles averages randomized rays from the camera into the scene using Monte-Carlo simulation [8]. The rays are continually reflected or refracted until they either get absorbed by objects, hit a light source or reach their bounce limit. This process, when applied to paths of light is called path tracing. Eevee uses rasterization via OpenGL 3.3 [8]. Rasterization works projects the faces of a model onto the pixels that make up the 2D image. The pixels have their RGB values adjsuted according to an objects various BRDFs and location in the scene i.e is it in a shadow. Eevee provides real time rendering [8]. As always, there is no free lunch, and this speed comes at the cost of accuracy. In this thesis we exclusively use Cycles.

A.1.1 HDR Environment Lighting Setup

HDR lighting provides realistic lighting and shadows due to the images higher range of luminance levels. They improve reflections from metallic and glossy materials created in 3D as panoramic HDR environment maps, give these materials a world around them to reflect. In Blender one must set up their HDR environment map as either 1) a background that casts no light, 2) a background that casts light or 3) is invisible, but casts light. Regardless of the setup, it's light contribution value must be set. This is widely dependant on the HDR's range of values. For all renderings and videos in this thesis we set a light contribution value of 0.0008. As our environment maps are estimated, we use the third setup for all renderings (i.e HDR environment map is invisible and only contributes light). This setup can be seen in A.1.

A.1.2 Enhancing Shadows to Remove Glare

High quality shadows make a rendered object look much more realistic. We perform two optimizations to provide higher quality shadows. First, we add a glossy BRDF to shadow catcher and set roughness to 0.4. This produces a more reflective material versus the default solid color plane that



World Output: All

Mapping — Type: Point

Location: X 0 m, Y 0 m, Z 0 m

Rotation: X 0°, Y 0°, Z 0°

Scale: X 1.000, Y 1.000, Z 1.000

$kVZiAlidmRHF...

Linear

Equirectangular

Single Image

Color Space: Linear

Strength 0.001

Figure A.1: Blender uses a shader node system. In this figure one can see the background is loaded in the node with the orange tab. To insure it only lights objects in the scene and is not the background, we use a light path node specifying that the HDR environment should only light objects in the camera's view.

catches shadows in blender. The second optimization to improve shadow quality is from [19]. The object O, shadow S, and background B, are rendered separately. The shadow is initially composited on a background of all white, 1.0 values. With these three renders we perform the following:

$$S^{'} = S^2$$
$$Out = (S^{'} * B) * O_{Mask} + O$$

$S^{'}$ is a darker shadow. This shadow, with dark RGB values close to 0, weights the image values lower. Then the object is then composited on top.

A.2 Cuboid Parameter Distribution and Confusion Matrices

In the following figures one can find a in-depth analysis of errors, as well as the exact test data distributions for all variables, for each face. Starting with the light presence, the back face of the cube is the only face in which the data distribution is not skewed towards "No Light". Light strength value distributions show that in all cases, the medium strength is the most frequent. Our predictions indicate they are done proportionally to the dataset distribution for all faces but the ceiling. Strength predictions for the ceiling are skewed towards stronger values. Light sizes for all faces are heavily skewed towards smaller lights. Despite this, our system predicts a good spread of light sizes. In terms of the ambient parameter, the floor's ambient values tend to be much brighter than the rest of the faces. This is likely due to it's constant exposure to both wall and ceiling light sources.

A.3 Cuboid Parameters Binned

In A.6 one can see the binning results of Light Size for walls and ceilings. Note that the ceiling lights are much smaller and have virtually no large lights. Therefore we decided to separate ceiling light size from wall light size to prevent a extremely skewed distribution. A.7 looks bell shaped. While we did try to regress this value due to it's distribution shape, ultimately classification provided better results. Lastly in A.8 one can see that the large majority of ambient values are very small.

A.4 More Qualitative Results

Below one can find more qualitative results of our method.

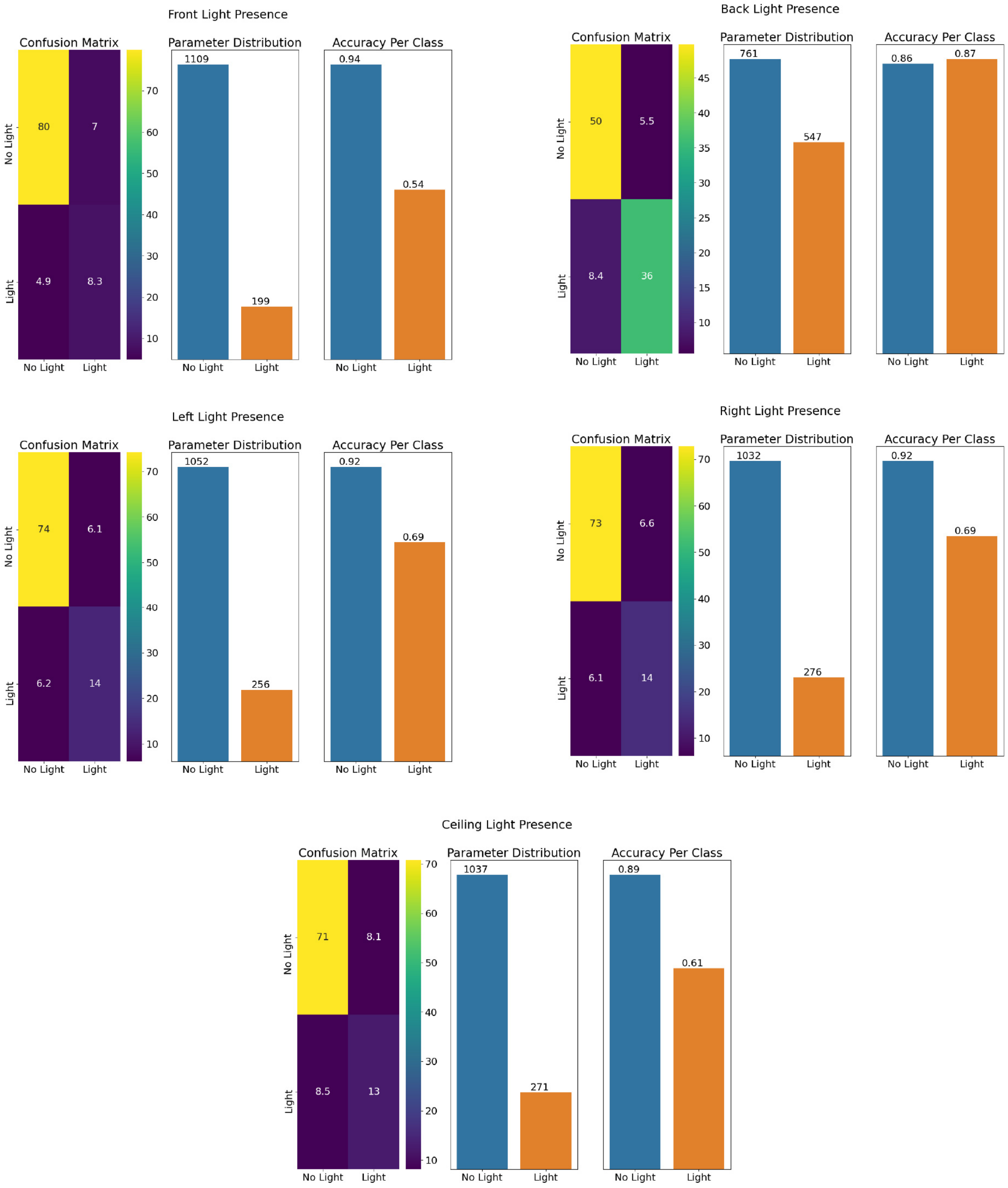


Figure A.2: Light Presence confusion matrices and data distribution per cuboid face.

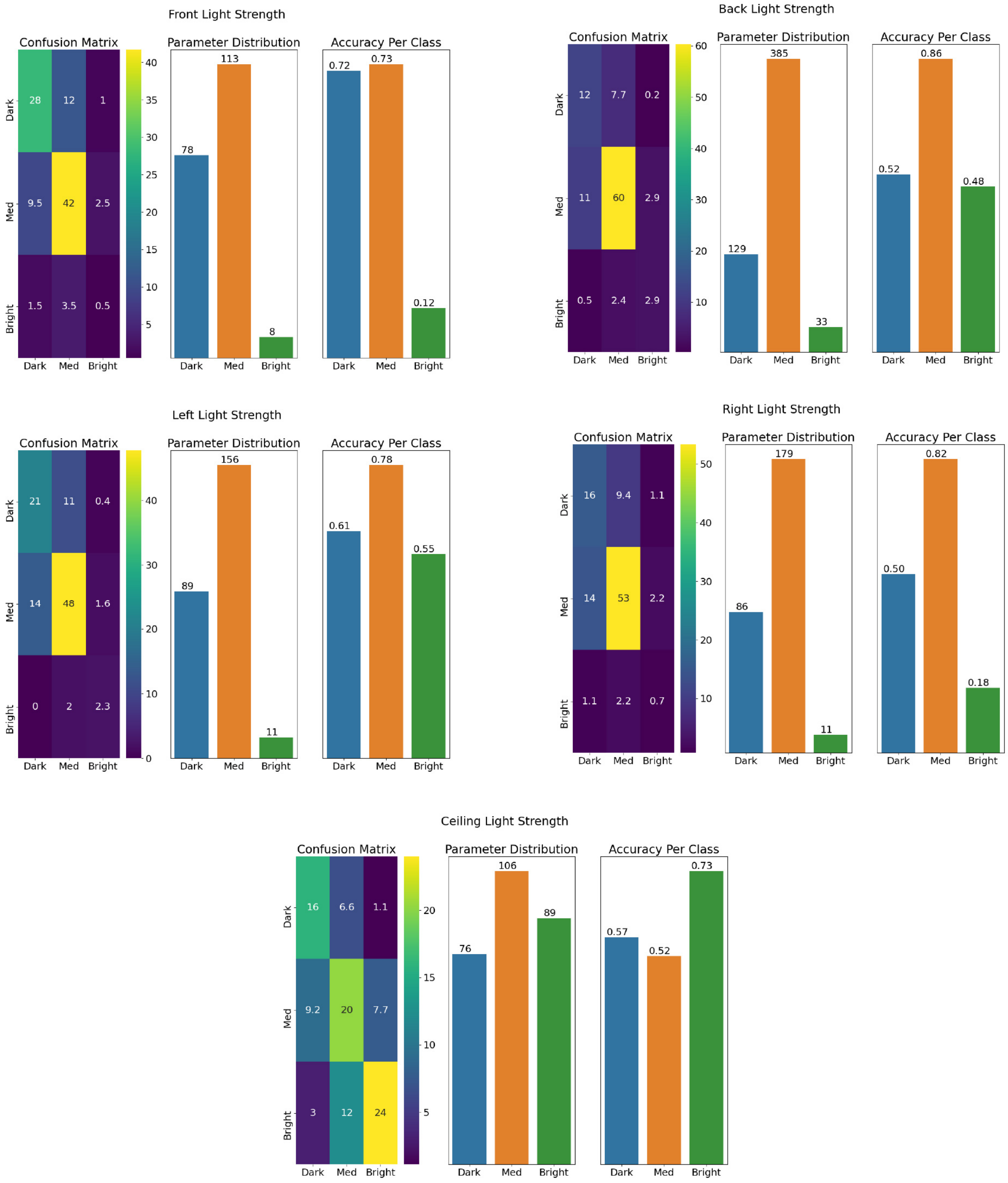


Figure A.3: Light Strength confusion matrices and data distribution per cuboid face.

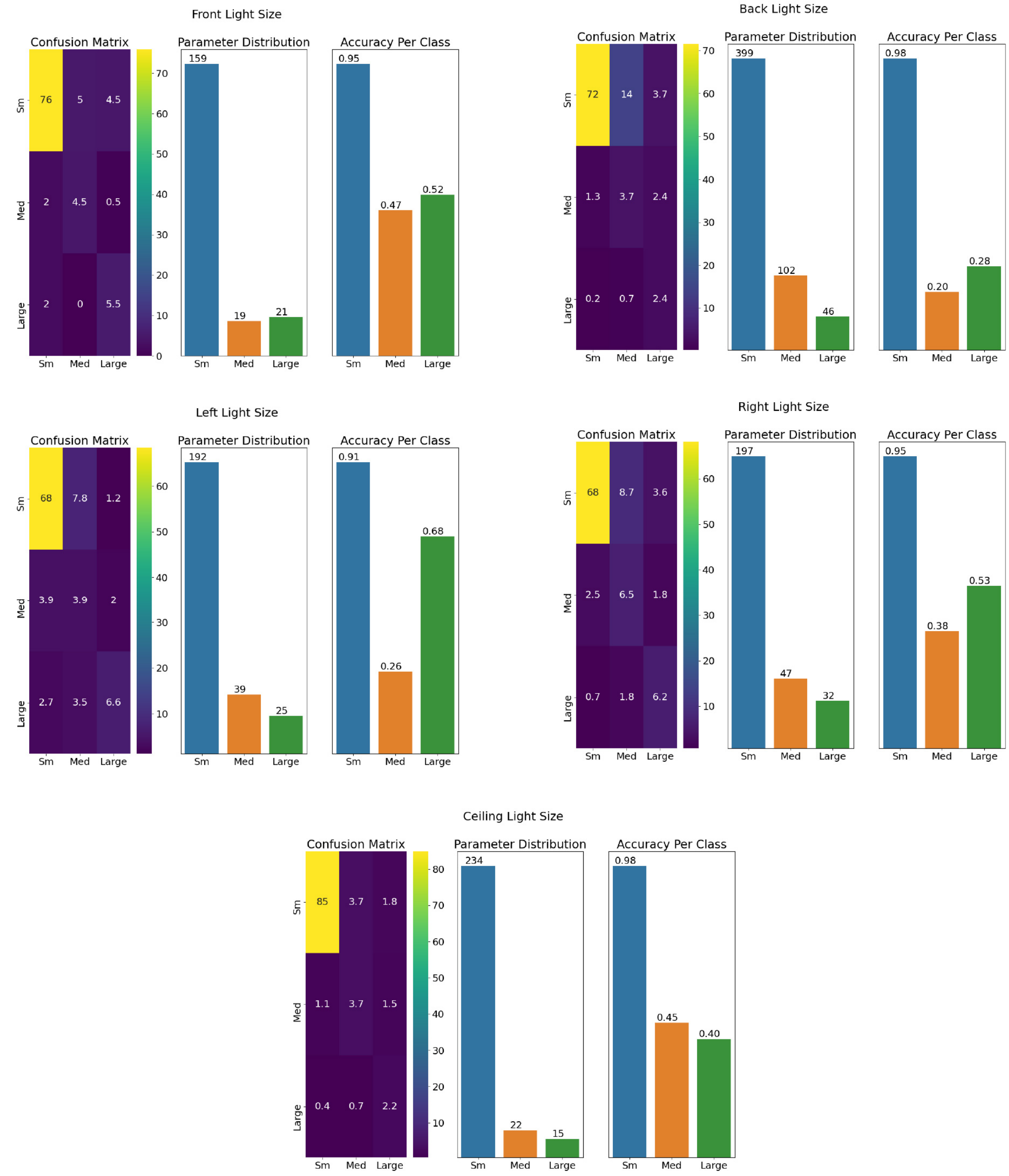


Figure A.4: Light Size confusion matrices and data distribution per cuboid face.

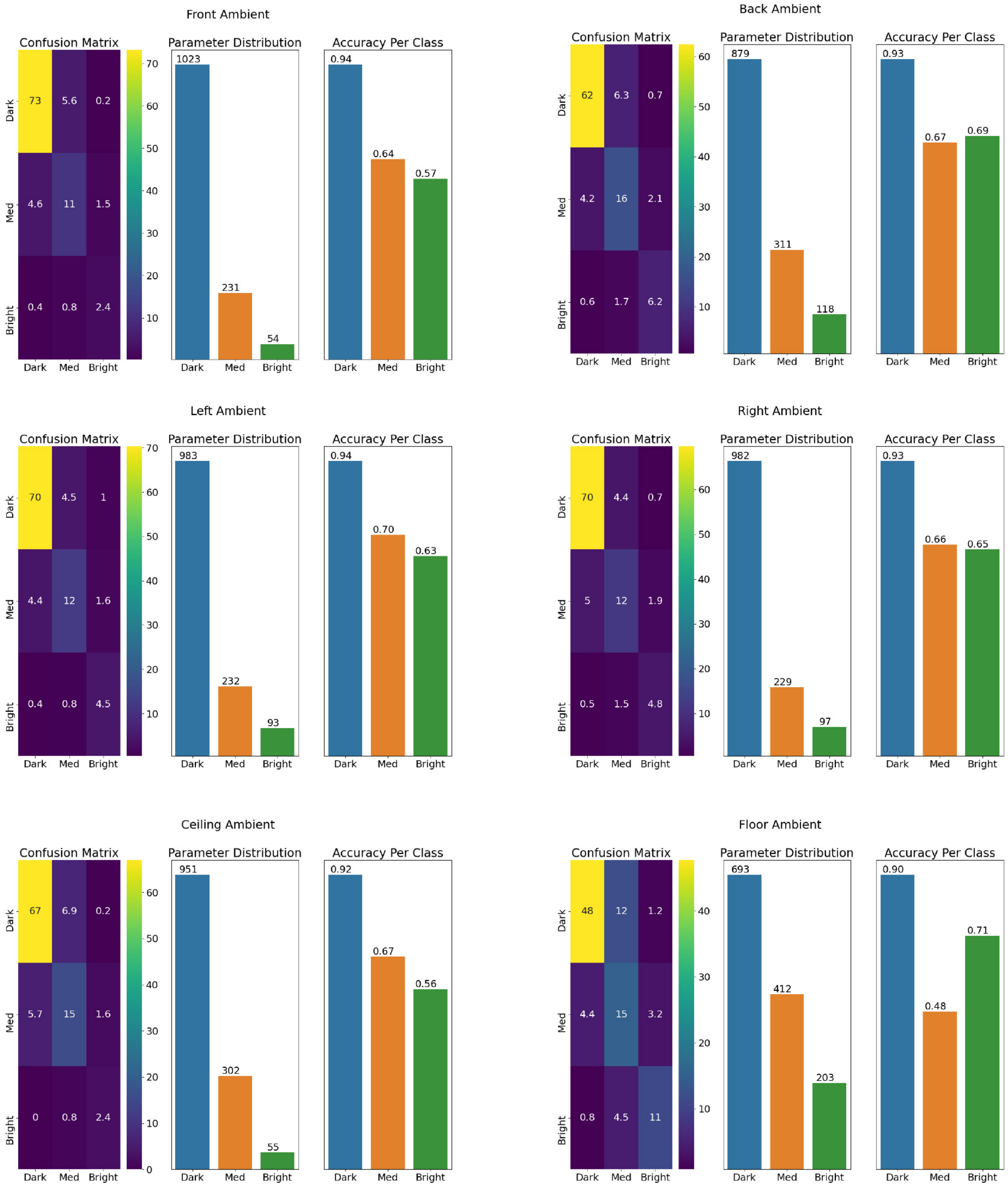


Figure A.5: Ambient value confusion matrices and data distribution per cuboid face.

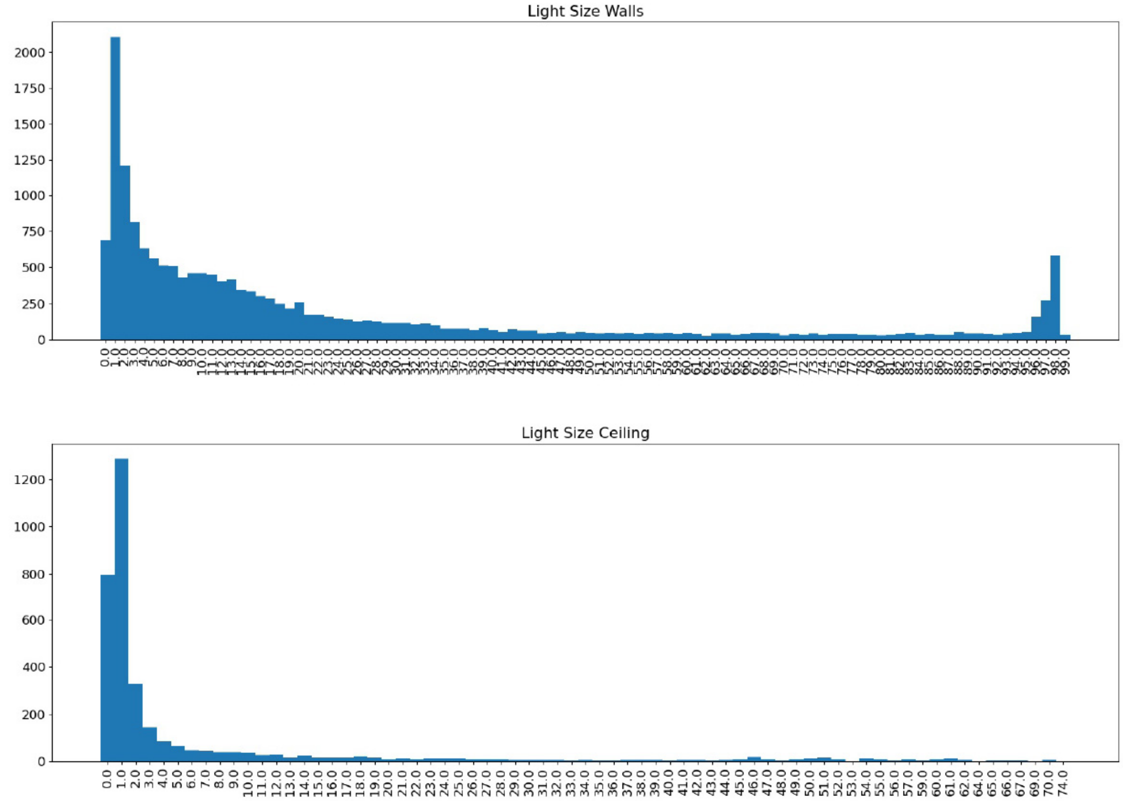


Figure A.6: Binning results of Light Size for walls and ceilings. Note that the ceiling lights are much smaller and have virtually no large lights.

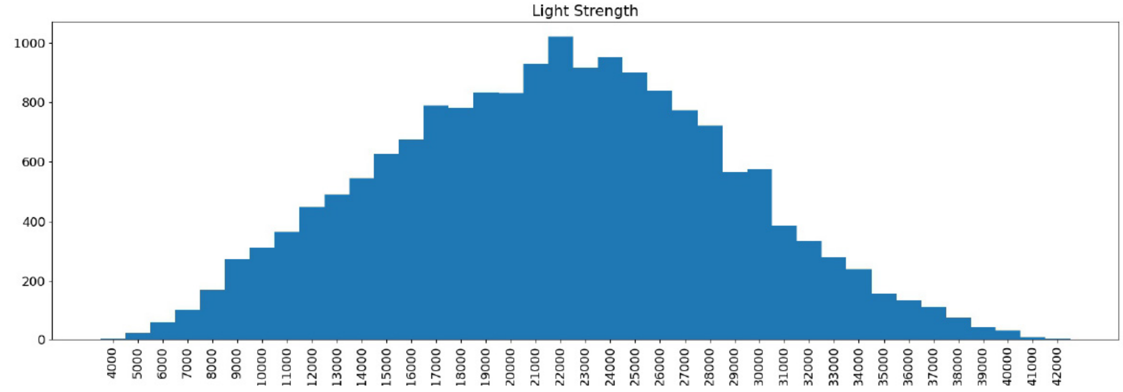


Figure A.7: Binning results of Light Strength for all faces but the floor.

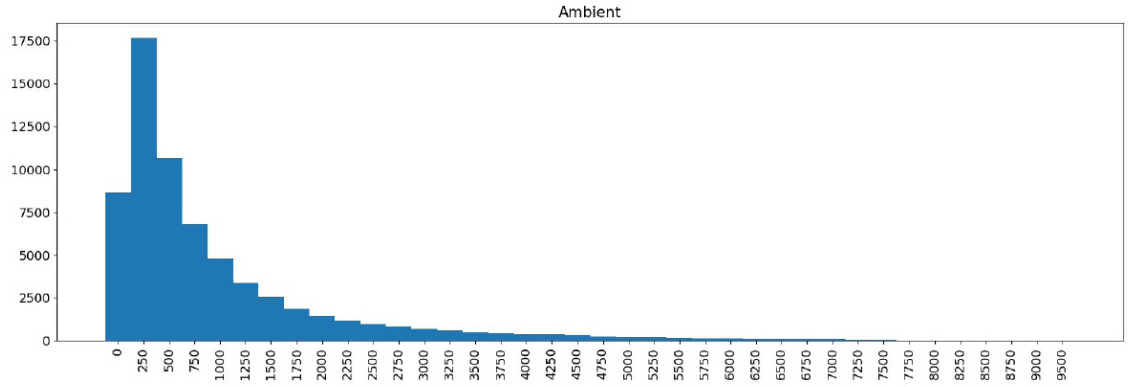


Figure A.8: Binning results of Ambient values for all faces.

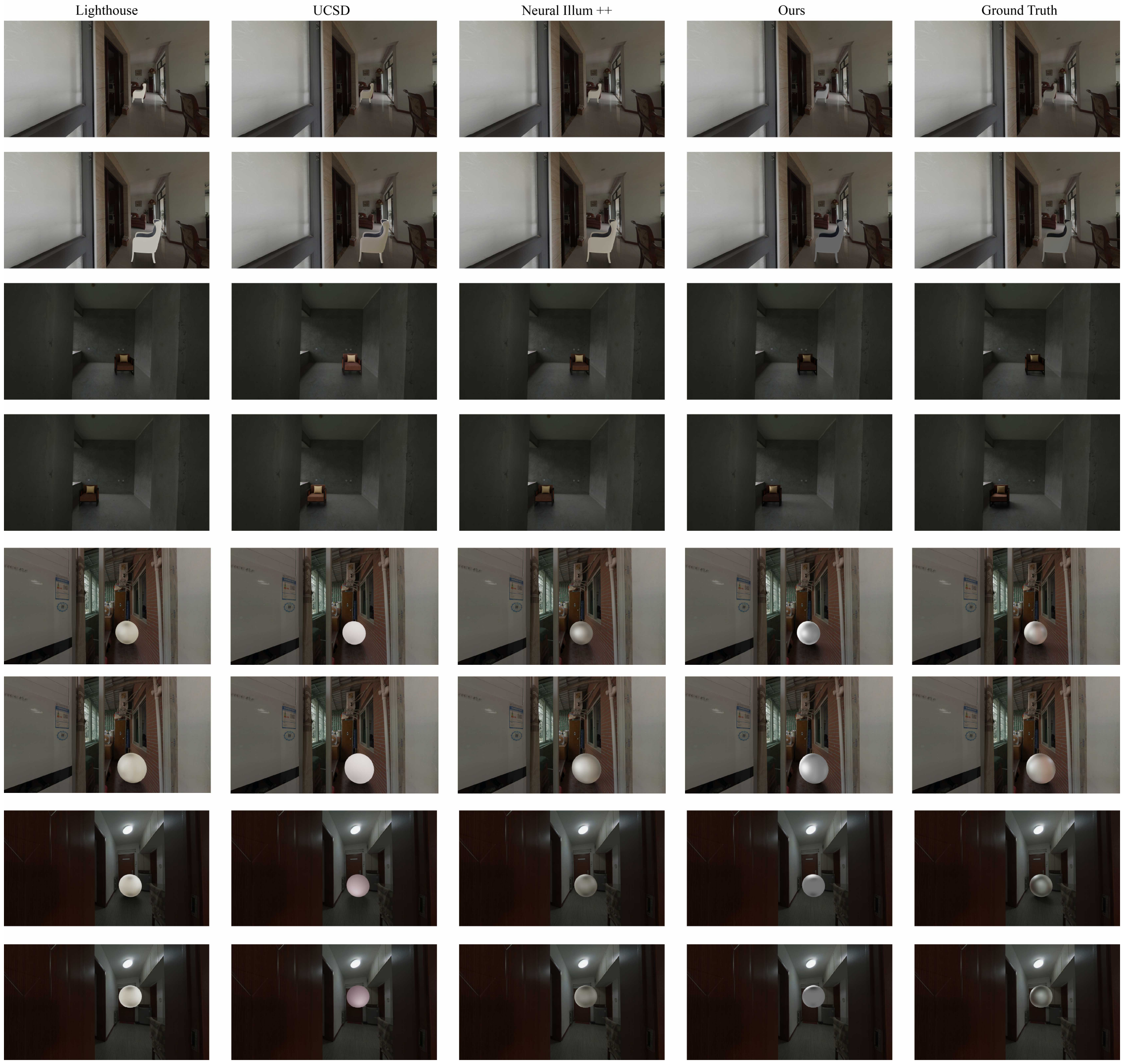


Figure A.9: Two Stills Qualitative Evaluation 2

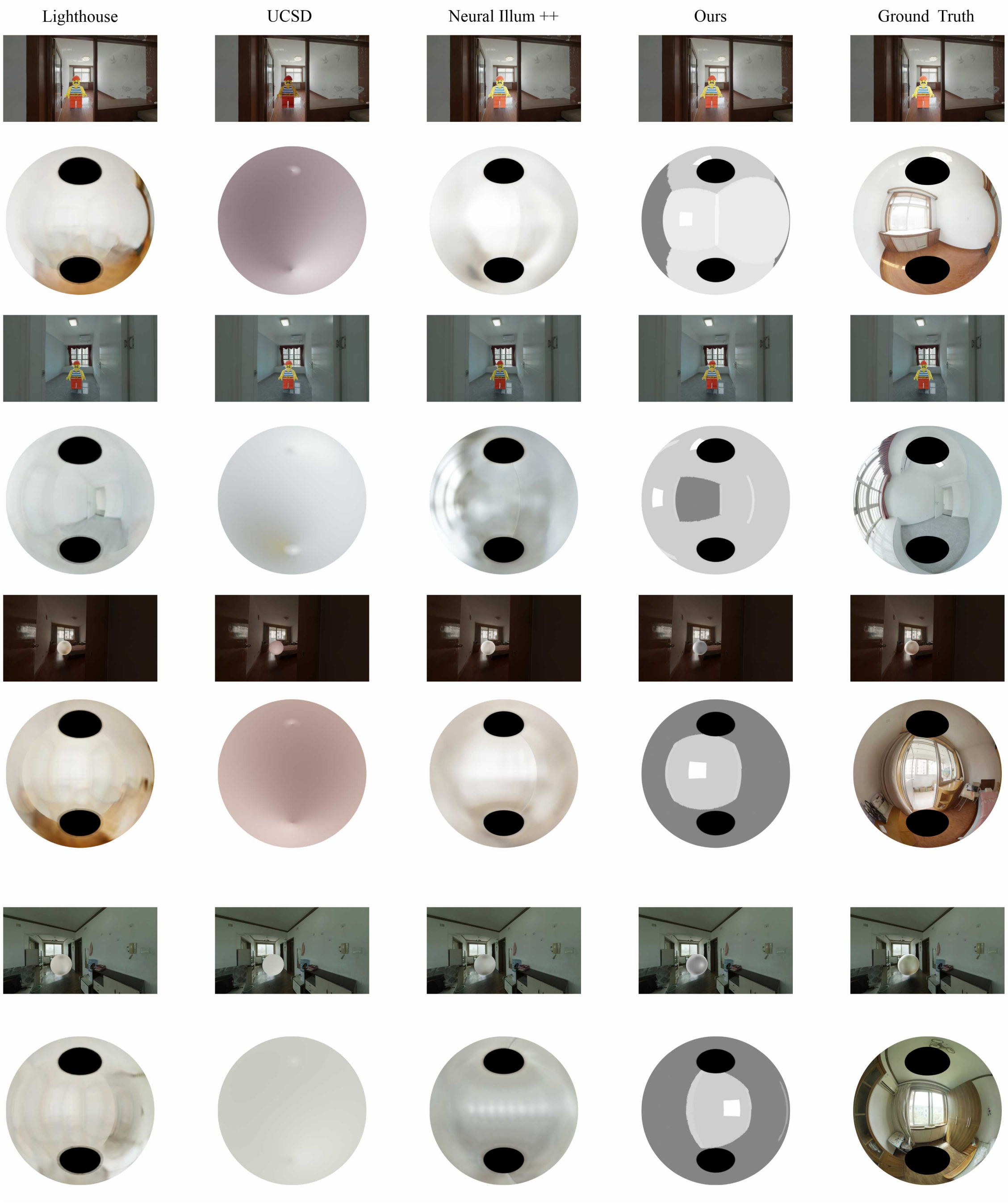


Figure A.10: Renders With Environment Maps 2

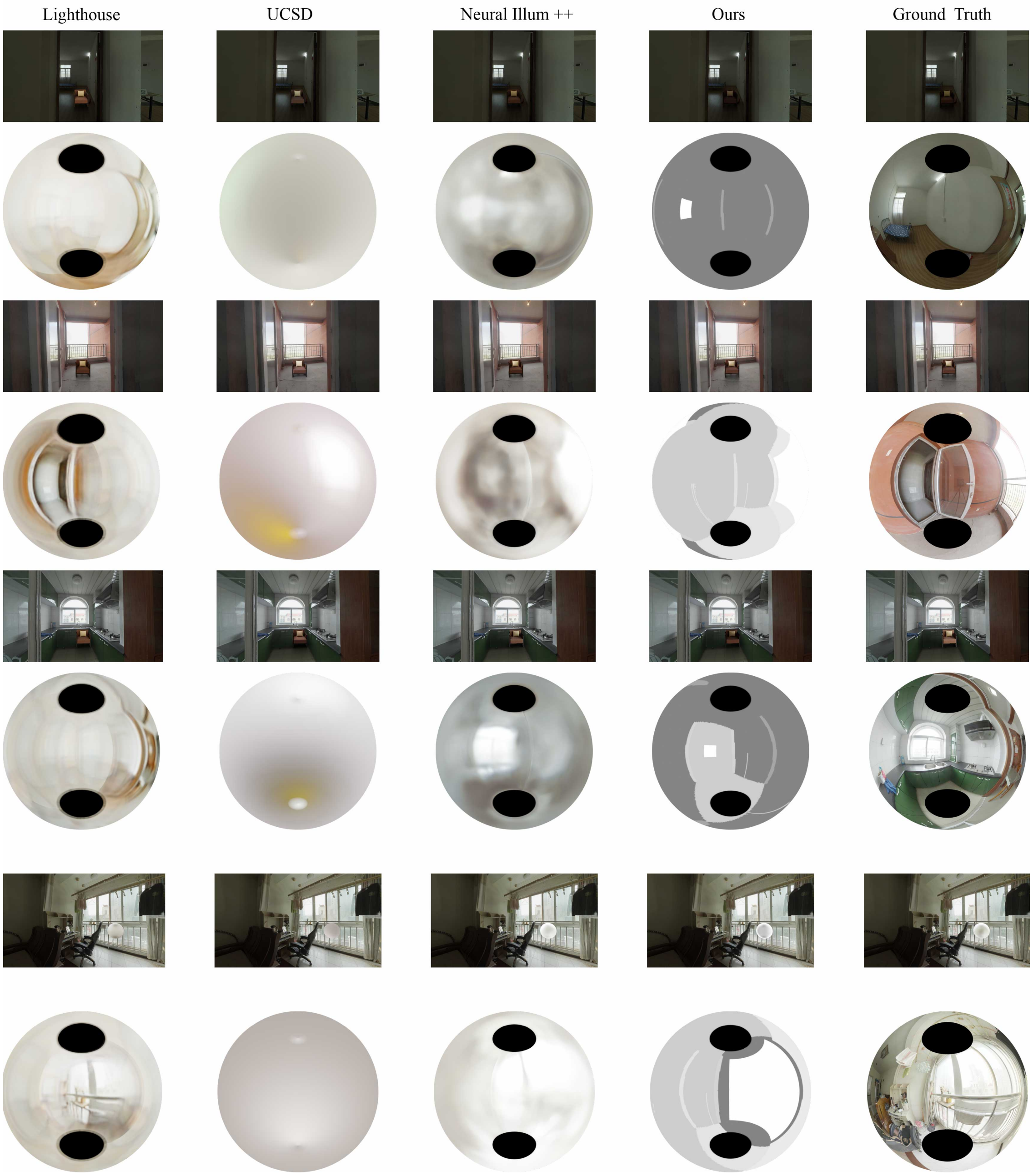


Figure A.11: Renders With Environment Maps 3