Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Vision Lab

NPSim: Nighttime Photorealistic Simulation From Daytime Images With Monocular Inverse Rendering and Ray Tracing

Project Thesis

Shutong Zhang
Department of Information Technology and Electrical Engineering

Advisor: Dr. Christos Sakaridis
Supervisor: Prof. Dr. Luc Van Gool

August 18, 2023

Abstract

Semantic segmentation is an important task for autonomous driving. A powerful autonomous driving system should be capable to handle images under all conditions, including nighttime. Generating accurate and diverse nighttime semantic segmentation datasets is crucial for enhancing the performance of computer vision algorithms in low-light conditions. In this thesis, we introduce a novel approach named NPSim, which enables the simulation of realistic nighttime images from real daytime counterparts with monocular inverse rendering and ray tracing. NPSim comprises two key components: mesh reconstruction and relighting. The mesh reconstruction component generates an accurate representation of the scene's structure by combining geometric information extracted from the input RGB image and semantic information from its corresponding semantic labels. The relighting component integrates real-world nighttime light sources and material characteristics to simulate the complex interplay of light and object surfaces under low-light conditions. The scope of this thesis mainly focuses on the implementation and evaluation of the mesh reconstruction component. Through experiments, we demonstrate the effectiveness of the mesh reconstruction component in producing high-quality scene meshes and its generality across different autonomous driving datasets. We also propose a detailed experiment plan for evaluating the entire pipeline, including both quantitative metrics in training state-of-the-art supervised and unsupervised semantic segmentation approaches and human perceptual studies, aiming to indicate the capability of our approach to generate realistic nighttime images and the value of our dataset in steering future progress in the field. NPSim not only has the ability to address the scarcity of nighttime datasets for semantic segmentation, but it also has the potential to improve the robustness and performance of vision algorithms under low-lighting conditions.

Contents

1 Introduction **1**

2 Related Work **3**
2.1 Novel View Synthesis . . . 3
2.2 Day-to-Night Transformation . . . 3
2.3 Nighttime Driving Scene Understanding . . . 4

3 Materials and Methods **5**
3.1 Problem Setting . . . 5
3.2 Data Preparation . . . 5
3.3 Geometric Mesh Reconstruction . . . 8
3.4 Depth Refinement Kernel . . . 8
3.4.1 Dual-reference Cross-bilateral Filter . . . 8
3.4.2 Dual-reference Variance Filter . . . 9
3.4.3 Normal-Guided Depth Refinement . . . 10
3.5 Mesh Post-processing Kernel . . . 11
3.5.1 Uncertain Faces Deletion . . . 11
3.5.2 Mesh Completion . . . 12
3.6 Realistic Nighttime Scene Relighting . . . 12
3.6.1 Material Characteristics Prediction . . . 13
3.6.2 Probabilistic Light Source Activation and Relighting . . . 13
3.6.3 Image Post-processing . . . 14

4 Results and Experiment Plans **15**
4.1 Geometry Mesh Reconstruction . . . 15
4.1.1 Datasets and Metrics . . . 15
4.1.2 Mesh Comparison . . . 15
4.1.3 Generalization to Other Datasets . . . 17
4.2 Testing Plans . . . 18
4.2.1 Datasets and Metrics . . . 18
4.2.2 Experiments . . . 18

5 Conclusion **19**

A Light Source Annotation Examples **21**

B Hyper-parameter Tuning **23**

C Mesh Reconstruction Examples **25**

D Model Selection **27**

List of Figures

3.1 **Method overview.** Our pipeline contains two components. The **Geometric Mesh Reconstruction** component first utilizes network $\mathbf{F_g}$ to estimate the geometric information of an input RGB image, then reconstruct scene mesh based on depth using the Worldsheet [17] (**A.II**: Section 3.3). It also contains a depth refinement kernel (**A.I**: Section 3.4) and a mesh post-process kernel (**A.III**: Section 3.5) to optimize depth and mesh, respectively. The **Realistic Nighttime Scene Relighting** (**B**: Section 3.6) component first generates nighttime light sources using probabilistic light source activation. Then predict the material characteristics using network $\mathbf{F_{ir}}$. Following that, it uses ray tracing to render the linear nighttime clear image. Last, it processes the linear nighttime image to simulate artifacts and finally generates the output nighttime image $\mathbf{I_n}$. In this thesis, we implemented the Geometric Mesh Reconstruction component. 6

3.2 **Annotation Statistics.** We show the annotation statistics of 230 images nighttime reference images. The left image shows the number of instances of each type of light source, and the right image shows the number of pixels occupied by each type of light source. All results are presented in the base-10 logarithm. 6

3.3 **Annotation Examples.** We present several examples of our inactive light source annotation. The left column shows the input daytime RGB image, the middle column shows the annotated inactive light source mask, where each instance has its own identity and bounding box, and the right column superimposes the RGB image and the light source mask. 7

3.4 **Depth comparison before and after the Dual-reference cross-bilateral filter.** We present two examples for depth comparison. Column **(a)** shows the original RGB image, column **(b)** shows the depth estimated by iDisc [39], and column **(c)** shows the depth after dual-reference cross-bilateral filter optimization. The above comparison shows that the dual-reference cross-bilateral filter improves the depth estimation at the pixel level. 9

3.5 **Uncertain region detected by the Dual-reference variance filter.** Column **(a)** shows the original RGB image, column **(b)** shows the semantic annotation and column **(c)** shows the generated uncertain map, where the uncertain region is represented by the yellow region. . . 10

3.6 **Completed uncertain region.** Column (**a**) shows the original RGB image, column (**b**) shows the semantic annotation and column (**c**) shows the completed uncertain region, where the uncertain region is represented by the yellow region. 12

4.1 **Mesh reconstruction result comparison.** We compared the mesh reconstruction result with mesh reconstructed by other methods with the following settings, (1): Worldsheet with MiDaS depth, (2): Worldsheet with iDisc depth, (3): SIMBAR reconstruction. Our method can preserve more accurate geometric information and construct smoother mesh surfaces than all other methods. 16

4.2 **Inferred surface normal comparison.** We compared the surface normal inferred from the reconstructed mesh with other methods (as stated in Fig. 4.1.2). Our reconstructed meshes have more accurate surface normal with respect to iDisc [39] surface normal. 16

4.3 **Mesh comparison from different angles and background mesh.** We compared the mesh reconstruction result with mesh reconstructed by other methods (, as stated in Fig. 4.1.2.) with a 3 meters zoom in and a 45 degrees rotation clockwise as well as background mesh generated with our pipeline. The result shows that our method can efficiently break the unexpected connection between foreground and background objects and make the background mesh watertight. 16

4.4 **Optimization loss curve.** We present the optimization loss curve of two examples shown in Fig. 4.1.2. 17

4.5 **Mesh reconstructed from different datasets.** We showcase the mesh reconstruction based on two different datasets: BDD100K dataset [65] (column (a) and (b)) and Cityscapes dataset [6] (column (c) and (d)). The above results demonstrate the capability of our pipeline to generate reasonable scene mesh when applied to other datasets. However, we notice that roads close to the camera are not correctly reconstructed, this is a result of inaccurate surface normal prediction caused by the ego vehicle. 17

A.1 **More Annotation Examples.** We present more examples of our inactive light source annotation. The left column shows the original RGB image, the middle column shows the annotated inactive light source mask, where each instance has its own identity and bounding box, and the right column superimposes the RGB image and the light source mask. 22

B.1 **Small λ_2.** We present an example of surface normal and final mesh with different λ_2 values. The upper row shows an example of setting λ_2 too small and the lower row shows an example of setting λ_2 too large. 23

B.2 **Large λ_3.** We present an example of surface normal and final mesh when setting λ_3 too large. 24

C.1 **More mesh reconstruction examples.** We present more examples of mesh reconstructed using our method and their corresponding surface normal. 25

C.2 **More mesh reconstruction examples.** We present more examples of mesh reconstructed using our method and their corresponding surface normal. 26

List of Tables

3.1 **Inactive light source class and their definition**. We defined 21 types of inactive light sources, each of them belonging to one category, including building, vehicle, object and group. 7

A.1 **Inactive light sources examples.** We present examples of different types of inactive light sources. The second and third columns show the daytime appearance and nighttime appearance of each example, respectively, arranged in the same order. 21

D.1 **iDisc normal estimation retrain result**. We present three different settings of our networks and their corresponding evaluation result. 27

Chapter 1

Introduction

Accurately parsing input images under uncommon visual conditions is a required ability for safe autonomous driving systems. Thus, semantic segmentation datasets under various adverse conditions are required, especially at nighttime. However, most well-known autonomous driving datasets such as the Cityscapes dataset [6] and KITTI dataset [13] are mostly under normal conditions (clear daytime image with high illumination and low exposure time). Though several recent works, including Oxford RobotCar dataset [34], BDD100K [65] dataset and ACDC dataset [47], have been focusing on creating datasets under nighttime conditions, there are still obvious limitations and gaps. For example, the Oxford RobotCar does not contain any semantic annotations and thus is not capable for training a segmentation network. BDD100K dataset contains a large number of nighttime images; however, only 345 of them can be used for the task of semantic segmentation. ACDC dataset contains 4006 adverse condition images in which 1006 of them are in the nighttime; it still shows a drastic drop in the accuracy of semantic scene understanding at nighttime compared to other conditions for both segmentation algorithms [3, 4, 10, 28, 31, 52] and domain adaptation methods [26, 32, 45, 53, 55, 56, 63]. What makes things even worse is the increased difficulty in the semantic annotation of real nighttime images due to their low quality, which leads to annotation errors that have a negative impact on the quality of models trained on such data. An alternative approach for semantic nighttime scene understanding is via the generation of partially synthetic nighttime data. Specifically, individuals can obtain images captured during the daytime, which is less noisy and relatively easier to annotate, then transform these images to nighttime through style transfer [11, 12]. Subsequently, annotations on daytime images can be directly used to the synthesized nighttime images, given that the underlying semantic content remains consistent. However, style transfer falls short in generating realistic results due to its inability to account for factors such as changes in illumination that occur from day to night and the complicated geometry and variations in light sources. In this thesis, we propose the NPSim, an alternative approach for generating partially synthetic nighttime images via monocular inverse rendering and ray tracing.

Our proposed method NPSim aims to generate photo-realistic nighttime images based on clear daytime images and their corresponding standard semantic annotations. Importantly, we also utilize light source annotations as additional input. Different from previous works that used style transformation [67], 2D [41] or implicit representation such as NeRF [36], NPSim focuses on the fundamentals of scene lighting, using traditional rendering method ray tracing to produce realistic nighttime images by restoring the object orientation, material characteristics, and light sources based on the input image. It leverages the explicit representation by reconstructing scene mesh from a given RGB image. To preserve more accurate geometric information, we propose Geometry Mesh Reconstruction component: We first utilize an off-the-shelf depth and normal estimation model to predict the initial depth map and normal map. Then we use predicted depth to reconstruct the scene mesh and refine it using a normal-guided optimization-based method. Besides, our proposed NPSim considers real-world light sources to generate realistic nighttime images. In the Realistic Nighttime Scene Relighting component, we employ ray tracing to generate nighttime visuals by considering geometric scene mesh from previous steps, material attributes, and authentic light sources from the real world. Our probabilistic light source activation also has the potential to activate different light source combinations, generating multiple nighttime images based on one single input image.

In summary, the contributions of our work are:

- A dataset of inactive light sources and the corresponding light source dataset, containing strength and chromaticity tuples (Section 3.2).
- A physically-based day-to-night simulation pipeline that contains a Geometry Mesh Reconstruction component (Section 3.3, Section 3.4, Section 3.5) and a Realistic Nighttime Scene Relighting component (Section 3.6).

In the experiment, our method can achieve better geometry reconstruction compared with previous works [17, 66]. Additionally, we show that our method can be generalized to other autonomous driving datasets such as the Cityscapes dataset [6] and the BDD100K dataset [65] for the task of mesh reconstruction. The structure of the thesis is as follows: We first describe methods we used for data collection, then we introduce our day-to-night transformation method NPSim. In the end, we present the results of the Geometric Mesh Reconstruction component and testing plans for the entire data generation pipeline.

Chapter 2

Related Work

Our work aims to create realistic nighttime images for autonomous driving based on single-view daytime images, it is closely related to novel view synthesis, day-to-night transformation and nighttime driving scene understanding. In this section, we will present an overview of the most relevant works and their limitations.

2.1 Novel View Synthesis

View synthesis is a fundamental task in computer vision that involves generating new images of a scene from different viewpoints. Early works focused on geometric reconstruction usually combine Structure-from-Motion (SfM) and Multi-View Stereo (MVS) that rely on sparse feature matching and depth estimation. However, these methods require multiple viewpoints of a single scene and cannot handle complex scenes [48]. The recent influential Neural Radiance Fields (NeRF) technique [36] shows strong ability in novel view synthesis and is capable for both indoor and outdoor scenes [25, 33, 50, 44] or objects [51]. Different from previous works that leverage geometry information, it often designs as a multi-layer perceptron (MLP) [40] that maps 3D coordinates to radiance and density values. NeRF learns to model the radiance field by minimizing the difference between the synthesized images and the actual training image during training, then renders different views of the scene at test time. Though being powerful and reliable in view synthesis, NeRF suffers from its high computational cost and limited generalizability. Many other works also explored view synthesis using generative networks, such as Variational Autoencoders (VAEs) [22], Generative Adversarial Networks (GANs) [14] and Diffusion Models [16]. They have shown remarkable abilities in generating realistic images, but their lack of 3D understanding makes it hard to capture the underlying geometry and thus may generate artifacts in novel views.

2.2 Day-to-Night Transformation

Day-to-night transformation is a challenging task that aims to convert images captured during the day into realistic nighttime representations. This process relies on scene relighting, a core task in computer graphics and computer vision that involves modifying the lighting conditions and then rendering the original scene under new conditions. Many previous works have explored this with different methodologies. [27] learns inverse rendering from a single image, estimating the geometry and materials of the scene and spatially-varying illumination. [62] proposed to complement the intrinsic estimation from volume rendering using NeRF and from inversing the photometric image formation model using convolutional neural networks (CNNs) for outdoor scene relighting. Differently, [66] leveraging explicit geometric representations from a single image by estimating depth information using an external network to perform scene relighting.

All these methods have achieved remarkable performance in scene relighting. Nevertheless, those methods only handle daytime images for both input and output, neglecting the impact of internal light sources. As a result, they are inadequate for accomplishing effective day-to-night transformation. For day-to-night transformation, most works utilized generative methods, such as CycleGAN [67], pix2pix [18] and EnlightenGAN [19]. Such purely data-driven approaches cannot accurately render spatially varying illumination, especially at night. Furthermore, although these methods sometimes do succeed in turning inactive light sources (e.g. street lights or windows) from off to on, the lights they produce are not accurate and realistic. Relighting daytime images to nighttime is also addressed in [41], which did not consider 3D geometry or materials and thus cannot model the interaction of light with the scene at night time. Moreover, nighttime-activated light sources are modelled in 2D instead of 3D, which leads to unrealistic illumination in the output image.

2.3 Nighttime Driving Scene Understanding

Parsing and understanding the driving scene is a crucial ability for autonomous driving cars. Semantic segmentation has developed rapidly over the past few years and achieved remarkable progress. However, comprehension of nighttime driving scenes is still in its early stages, mainly due to the significant domain gap between daytime and nighttime scenes. Some works performed domain adaptation to close this gap. [23] Utilized a physics-based prior for domain adaptation, aiming to minimize the distribution shift between daytime and nighttime neural network feature maps. [63] then relied on the pixel-level adaptation via explicit transforms from source to target. An alternative method is to train traditional segmentation models on nighttime driving datasets, however, this requires annotated nighttime images which are hard to obtain. Though many datasets such as the Oxford RobotCar dataset and the BDD100K dataset have been including nighttime images [65, 34], there has been a lack of emphasis on nighttime scene comprehension. As a result, these datasets do not offer adequate resources for training an effective model on nighttime image segmentation. A recently proposed autonomous driving dataset ACDC focused specifically on adverse conditions, contains 4006 images that are evenly distributed across four weather conditions: rain, fog, snow and night [47]. Each image comes with a pixel-level semantic annotation and a reference image that is taken at the same location under normal conditions (clear daytime). Though the ACDC dataset puts a larger emphasis on nighttime (it includes 1006 nighttime images, with 400 from the training set, 106 from the validation set and 500 from the testing set), the gap still remains due to the shortage of annotated nighttime images caused by the difficulties of manual annotation.

Different from all methods discussed above, our method targets the generation of realistic nighttime images through simulation based on images from daytime datasets. In our image simulation pipeline, we utilize geometric information to reconstruct scene mesh and consider real-world light sources during relighting. As shown in the remaining sections of the paper, our work has the potential to close the gap in nighttime driving scene understanding.

Chapter 3

Materials and Methods

In this section, we first formalize the problem setting. Next, we introduce our data preparation procedure. Finally, we introduce our data generation pipeline involving geometry mesh reconstruction and realistic nighttime scene relighting. Fig. 3.1 shows an overview of our data generation pipeline.

3.1 Problem Setting

Our goal is to generate realistic nighttime images based on daytime images in the ACDC dataset [47] reference split using a physics-based method. As described in Section 2.3, the ACDC dataset reference split contains 2006 daytime images from the training and validation reference set, in which 1003 of them come with a corresponding semantic annotation. Given those 1003 images and their semantic annotations, we propose a data processing pipeline that generates realistic nighttime images for each input image.

3.2 Data Preparation

As shown in Fig. 3.1, except for daytime RGB images I_d and its corresponding segmentation annotations S, our pipeline also takes a binary mask M_i indicating the inactive light sources in I_d, and a set of nighttime light sources E as input. In this section, we show the techniques employed for the creation of binary masks and the light sources.

Binary mask $\mathbf{M_i}$ Though some works have already been focusing on light source separation [64], they are not sufficient in such complex situations. Thus, we manually annotated inactive light sources using Segments.ai [1], an online image labelling platform based on super-pixels. To control the quality of our annotations, two annotators first annotated 100 images, then conducted a cross-check by checking each other's annotations and then modifying the annotation rule until a consensus was reached. Ultimately, we defined 12 classes of inactive light sources and categorized them into three primary groups: buildings, vehicles, and objects. We also propose that light sources from the same category should be mutually related, for example, windows on the same floor or car lights from the vehicle. Thus, to ensure the interconnection between light sources, we also defined 9 group classes, see Table 3.1 for all classes and their definitions. In this thesis, we annotated 230 out of 253 images from the nighttime reference split (we filtered out 23 images as they were taken at twilight). Fig. 3.2 shows the statistics of annotated pixels and the number of instances for each class. Fig. 3.3 shows some visual examples of our annotation. More examples can be found in Appendix A.

Inactive light source $\mathbf{I_d}$ To make our nighttime images as realistic as possible, we planned to collect the light source dataset E from a real-world setting. Following the method described in [41], we will first place gray cards under different nighttime illuminations (i.e. near different light sources identified in Table 3.1). Next, we will capture images of each gray card to extract both chromaticity value $[\frac{r}{g}, \frac{b}{g}]$ and strength value s.

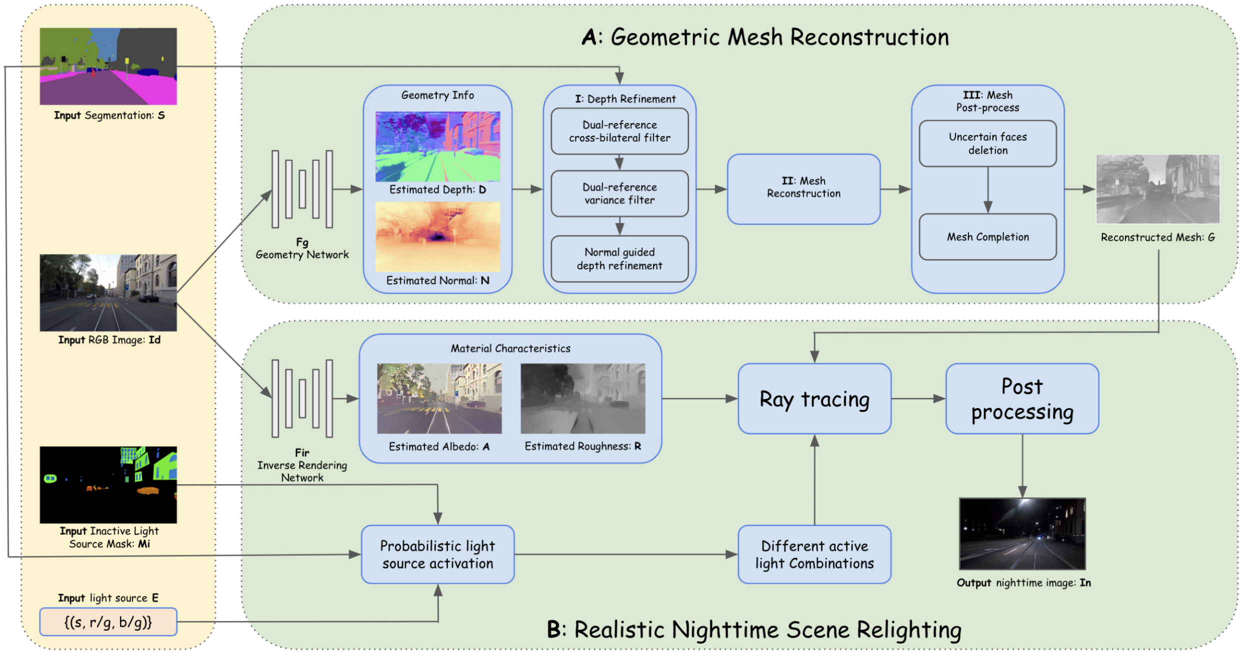
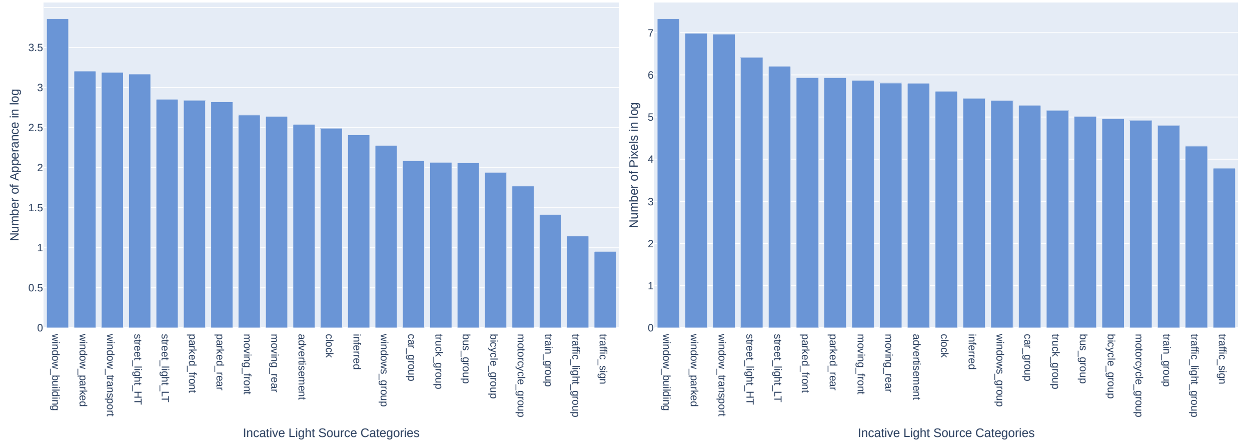


Figure 3.1: **Method overview.** Our pipeline contains two components. The **Geometric Mesh Reconstruction** component first utilizes network $\mathbf{F_g}$ to estimate the geometric information of an input RGB image, then reconstruct scene mesh based on depth using the Worldsheet [17] (**A.II**: Section 3.3). It also contains a depth refinement kernel (**A.I**: Section 3.4) and a mesh post-process kernel (**A.III**: Section 3.5) to optimize depth and mesh, respectively. The **Realistic Nighttime Scene Relighting** (**B**: Section 3.6) component first generates nighttime light sources using probabilistic light source activation. Then predict the material characteristics using network $\mathbf{F_{ir}}$. Following that, it uses ray tracing to render the linear nighttime clear image. Last, it processes the linear nighttime image to simulate artifacts and finally generates the output nighttime image $\mathbf{I_n}$. In this thesis, we implemented the Geometric Mesh Reconstruction component.



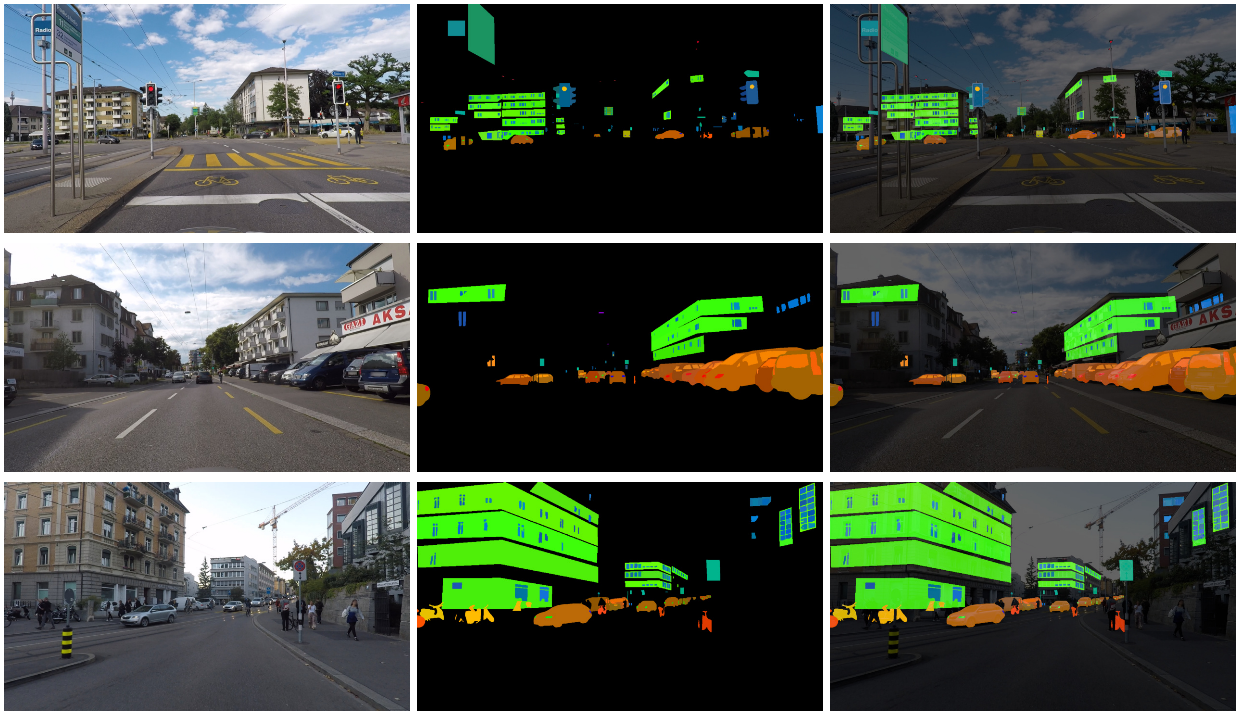
(a) Instance Statistics

(b) Pixel Statistics

Figure 3.2: **Annotation Statistics.** We show the annotation statistics of 230 images nighttime reference images. The left image shows the number of instances of each type of light source, and the right image shows the number of pixels occupied by each type of light source. All results are presented in the base-10 logarithm.

No.	Name	Category	Detailed explanation
1	window_building	building	Building windows that may emit light at night
2	window_parked	vehicle	Windows of parked vehicles, mainly cars
3	parked_front	vehicle	Front light of parked vehicles, mainly cars
4	parked_rear	vehicle	Rear light of parked cars, mainly cars
5	moving_front	vehicle	Front light of moving vehicles, mainly cars
6	moving_rear	vehicle	Rear light of moving vehicles, mainly cars
7	window_transport	vehicle	Windows of public transportation that may emit light
8	street_light_HT	object	High temperature traffic lights, usually brighter
9	street_light_LT	object	Low temperature traffic lights, usually dimer
10	advertisement	object	Advertisements that may emit light at night
11	clock	object	Clocks that emit light at night, mostly appear at bus stop
12	inferred	object	Light sources whose light colour can be inferred from its daytime color
13	windows_group	group	Group of windows that belong to the same floor of the same building
14	car_group	group	Group of light sources that belong to the same car
15	truck_group	group	Group of light sources that belong to the same truck
16	bus_group	group	Group of light sources that belong to the same bus
17	bicycle_group	group	Group of light sources that belong to the same bicycle
18	motorcycle_group	group	Group of light sources that belong to the same motorcycle
19	train_group	group	Group of light sources that belong to the same train
20	traffic_light_group	group	Group of traffic lights that belong to the same panel
21	traffic_sign_group	group	Group of light sources that belong to the same sign

Table 3.1: **Inactive light source class and their definition**. We defined 21 types of inactive light sources, each of them belonging to one category, including building, vehicle, object and group.



(a) Input daytime RGB image

(b) Inactive light source mask

(c) Superposition of (a) and (b)

Figure 3.3: **Annotation Examples.** We present several examples of our inactive light source annotation. The left column shows the input daytime RGB image, the middle column shows the annotated inactive light source mask, where each instance has its own identity and bounding box, and the right column superimposes the RGB image and the light source mask.

3.3 Geometric Mesh Reconstruction

The first component of our pipeline involves reconstructing scene mesh from a daytime RGB image and its corresponding segmentation annotation. Depth and surface normal are key geometric information that we rely on to reconstruct the mesh. In particular, we use iDisc [39] as out Geometry Network (F_g) to estimate depth (D) and normal (N). For depth estimation, we directly utilize the model pre-trained on the KITTI dataset [13]. For normal estimation, the pre-trained model on the NYUv2 dataset [38] doesn't adapt well to the ACDC dataset [47], so we retrain the iDisc model using the DIODE dataset [54] outdoor split, initializing the training with pre-trained swin transformer [30] weight.

Inspired by previous work SIMBAR [66], we apply Worldsheet [17], a novel view geometry scene synthesis method to reconstruct scene mesh. Worldsheet builds a scene mesh by warping a grid sheet onto the scene geometry via grid offset and depth. The grid exhibits a horizontal offset $\Delta\hat{x}$ and a vertical offset $\Delta\hat{y}$. Importantly, there is no need for predicting or adjusting individual vertices in the x and y directions. The mesh vertices are formulated from grid offset and depth as:

$$V_{w,h} = \begin{bmatrix} d_{w,h} \cdot (\hat{x}_{w,h} + \Delta\hat{x}_{w,h}) \cdot \tan(\theta_F/2) \\ d_{w,h} \cdot (\hat{y}_{w,h} + \Delta\hat{y}_{w,h}) \cdot \tan(\theta_F/2) \\ d_{w,h} \end{bmatrix} \tag{3.1}$$

Where d denotes the external depth, x and y denote the horizontal and vertical location of vertices in the mesh coordinates equally spaced from -1 to 1, and θ_F is the camera field of view. Thus, we are able to reconstruct the scene mesh based on external depth. However, this requires that the input depth maps we provide have high precision. In the next two sections, we will introduce the Depth Refinement Kernel that optimizes depth before the reconstruction and the Mesh Post-process Kernel after the reconstruction.

3.4 Depth Refinement Kernel

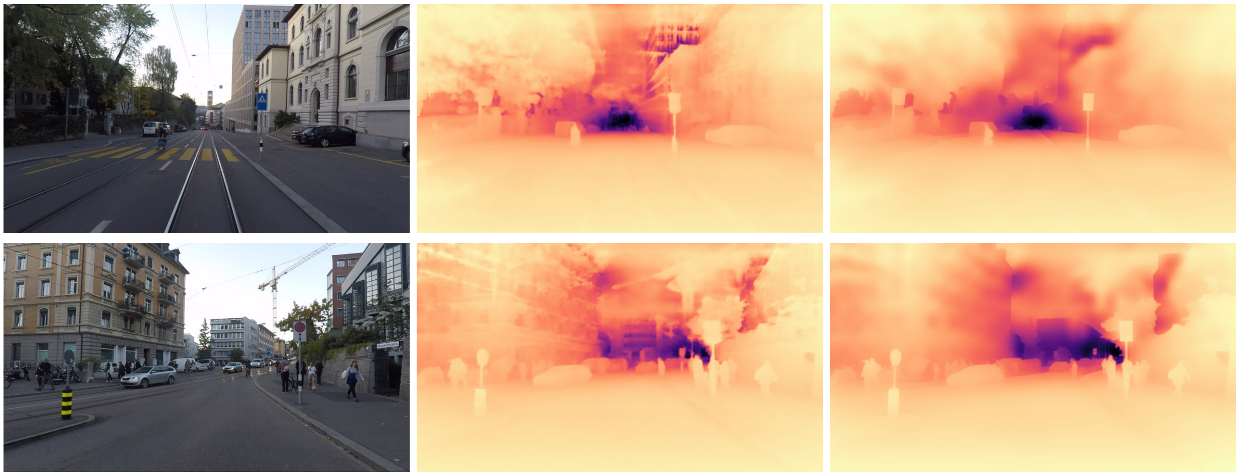
The main purpose of the Depth Refinement Kernel is to optimize depth based on segmentation annotations and predicted normal map, where we treat predicted normal as ground truth during optimization. This kernel consists of three parts: Dual-reference cross-bilateral filter, Dual-reference variance filter and Normal guided depth refinement.

3.4.1 Dual-reference Cross-bilateral Filter

Inspired by [45], we apply the Dual-reference cross-bilateral filter as the first building block of our Depth Refinement Kernel. In our work, we optimize input depth $\hat{d}$ using the RGB image and the semantic annotation to obtain filtered depth d, shown as Eq. (3.2).

$$d(\mathbf{p}) = \frac{\sum_{q \in \mathcal{N}(\mathbf{p})} G_{\sigma_s}(\|\mathbf{q} - \mathbf{p}\|) \left[\delta(h(\mathbf{q}) - h(\mathbf{p})) + \mu G_{\sigma_c}(\|\mathbf{J}(\mathbf{q}) - \mathbf{J}(\mathbf{p})\|)\right] \hat{d}(\mathbf{q})}{\sum_{q \in \mathcal{N}(\mathbf{p})} G_{\sigma_s}(\|\mathbf{q} - \mathbf{p}\|) \left[\delta(h(\mathbf{q}) - h(\mathbf{p})) + \mu G_{\sigma_c}(\|\mathbf{J}(\mathbf{q}) - \mathbf{J}(\mathbf{p})\|)\right]} \tag{3.2}$$

Similar to [45], we use the CIELAB counterpart of the input RGB image $\mathcal{R}$, denoted by $\mathbf{J}$. Where $\mathbf{p}, \mathbf{q}$ means pixel locations, $\mathcal{N}$ means neighbouring pixels and h means semantic classes. δ is the Kronecker delta, G denotes the Gaussian kernel, where G_{σ_s} is the spatial Gaussian kernel and G_{σ_c} is the colour Gaussian kernel lead by constant μ. The numerator and denominator of this equation consist of two main components, in which $\delta(h(\mathbf{q}) - h(\mathbf{p}))$ is for semantic references and $\mu G_{\sigma_c}(\|\mathbf{J}(\mathbf{q}) - \mathbf{J}(\mathbf{p})\|)$ is for colour references. The semantic component of this equation infers that only pixels from different semantic classes can contribute to this term, making the edge of each semantic object sharper. At the same time, the colour component helps to preserve depth that can be implied from the colour variation of the input RGB image. Following [45],



(a) Original RGB image (b) Depth before (c) Depth after

Figure 3.4: **Depth comparison before and after the Dual-reference cross-bilateral filter.** We present two examples for depth comparison. Column **(a)** shows the original RGB image, column **(b)** shows the depth estimated by iDisc [39], and column **(c)** shows the depth after dual-reference cross-bilateral filter optimization. The above comparison shows that the dual-reference cross-bilateral filter improves the depth estimation at the pixel level.

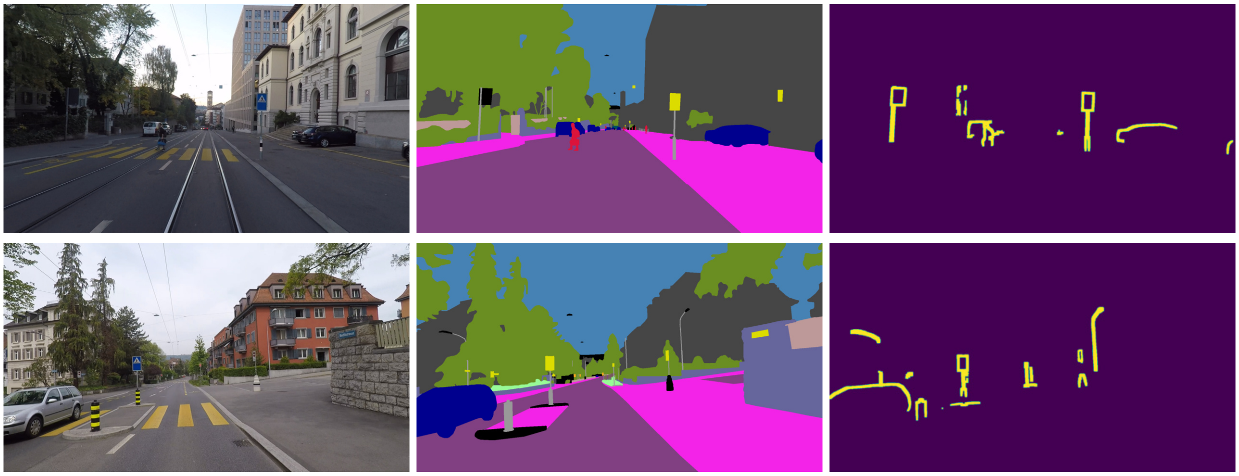
we initially set $\mu = 5$ and $\sigma_s = 10$. For the colour component, we discovered that when its weight grows larger, the depth change at semantic edges will become smoother and more spurious faces will be created (as described in Section 3.4.2). Thus, we decreased its weight and set $\sigma_c = 5$. Fig. 3.4 shows the comparison of depth maps before and after the Dual-reference cross-bilateral filter.

3.4.2 Dual-reference Variance Filter

One disadvantage of the Worldsheet [17] is that its generated scene mesh is an equal offsets mesh grid with only one layer of vertices. This creates spurious faces at depth discontinuities, connecting foreground objects with background objects. Although those spurious faces are not visible from the camera angle, they will still generate unrealistic reflections during the final relighting process. To solve this issue, we designed a Dual-reference variance filter to identify spurious faces based on depth maps and semantic annotations, as shown in Eq. (3.3). We propose that spurious faces usually happen at uncertain regions that meet the following two criteria: **(1)** regions that contain at least one semantic boundary. **(2)** regions that have a large variation in depth. For instance, consider a scenario where a region includes both a section of a moving car and a segment of the road or perhaps a part of a traffic sign alongside a portion of a building. We further define a set of foreground objects that are used to identify semantic boundaries, including vehicles, persons, poles, traffic lights and traffic signs.

$$\mathcal{U}(r(\mathbf{p}, l)) = (\mathcal{V}(d(r(\mathbf{p}, l))) > \mu) \text{ and } (\mathcal{V}(h(r(\mathbf{p}, l))) > 0) \tag{3.3}$$

In Eq. (3.3), $U(\cdot)$ denotes the binary value of the uncertain map, $\mathbf{p}$ denotes pixel location and $r(\mathbf{p}, l)$ represents the square region with a size of $l * l$ pixels with $\mathbf{p}$ as the upper left corner. The two components of Eq. (3.3) correspond to depth and semantic annotations, respectively. In the depth component $d(\cdot)$, we alert the region to be uncertain if the variance of all depth inside region $r(\mathbf{p}, l)$ is larger than a constant μ. In the semantic component $h(\cdot)$, we alert the region if variance of all semantic value inside region $r(\mathbf{p}, l)$ is larger than zero: when a semantic change happens. Two components are connected by a logic and operator,



(a) Original RGB image (b) Semantic annotation (c) Uncertain map

Figure 3.5: **Uncertain region detected by the Dual-reference variance filter.** Column **(a)** shows the original RGB image, column **(b)** shows the semantic annotation and column **(c)** shows the generated uncertain map, where the uncertain region is represented by the yellow region.

meaning the region will be marked as uncertain if both components are alerted simultaneously. In our experiment, we set $\mu = 0.001$ and $l = 8$. Fig. 3.5 shows two visual examples of the uncertain region detected by the Dual-reference variance filter.

3.4.3 Normal-Guided Depth Refinement

Depth estimation provides coarse geometric information regarding the input image, while surface normal offers further intricate local details. To increase depth accuracy, we propose an optimization-based depth refinement method based on surface normal. For each input image, we formulate a loss function based on the interrelationship between its depth and surface normal, then use gradient descent to optimize depth by minimizing the loss. Next, we will describe loss terms we used for the normal-guided depth refinement.

Normal loss. Given a depth map, we can infer the surface normal by computing the cross product of gradient vectors between neighbouring pixels, as shown in Eq. (3.4).

$$\hat{\mathbf{N}} = \nabla\mathbf{X} \times \nabla\mathbf{Y} = (1, 0, \frac{\partial z}{\partial x}) \times (0, 1, \frac{\partial z}{\partial y}) = (-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1) \tag{3.4}$$

Where $\frac{\partial z}{\partial x}$ and $\frac{\partial z}{\partial y}$ denotes the gradient of depth with respect to x and y in the camera space, which can be computed via chain rule $\frac{\partial z}{\partial x} = \frac{\partial z}{\partial u} \cdot \frac{\partial u}{\partial x}$ and $\frac{\partial z}{\partial y} = \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial y}$. And the transformation between pixel space and camera space is shown as Eq. (3.5).

$$\begin{aligned} u \cdot d = f_x \cdot x + c_x &\longleftrightarrow \frac{\partial u}{\partial x} = \frac{f_x}{d} \\ v \cdot d = f_y \cdot y + c_y &\longleftrightarrow \frac{\partial v}{\partial y} = \frac{f_y}{d} \end{aligned} \tag{3.5}$$

Where f_x and f_y denote the focal length along the x and y axis, c_x and c_y denote the principal point along the x and y axis, respectively. The final expression of this loss is formalized in Eq. (3.6). In this equation, we treated depth estimated by iDisc [39] $\mathbf{N}_{\text{est}}$ as ground truth.

$$L_{\text{normal}} = ||\hat{\mathbf{N}} - \mathbf{N}_{\text{est}}||_2^2 \tag{3.6}$$

Continuity loss. Although the normal loss can optimize depth to align its inferred normal with the reference surface normal, it lacks the ability to account for sudden variations in depth. As a result, this creates spurious faces, as discussed in Section 3.4.2. To tackle this issue, we proposed a continuity loss that directly optimizes gradient vectors based on surface normal, formalized as Eq. (3.7).

$$L_{continuity} = \frac{1}{n}\sum_{i=1}^{n}((\nabla \mathbf{X}_{\mathrm{i}} \cdot \mathbf{N}_{\mathrm{i}})^2 + (\nabla \mathbf{Y}_{\mathrm{i}} \cdot \mathbf{N}_{\mathrm{i}})^2) \cdot (1 - \mathcal{U}_{\mathrm{i}}) \tag{3.7}$$

In this equation, $\mathcal{U}$ is the uncertain map derived by the dual-reference variance filter in Eq. (3.3), $\nabla \mathbf{X}$ and $\nabla \mathbf{Y}$ are gradient vectors computed in Eq. (3.4), n is the total number of pixels of the input image and i represents the index of currently computing pixel. This loss term is aware of depth discontinuities as the sudden depth change between neighbouring pixels will create a gradient vector that is opposite from the normal vector, resulting large value of the dot product. Thus, it can help eliminate spurious faces created by the normal loss. The participation of the uncertain mask term $(1 - \mathcal{U}_{\mathrm{i}})$ deals with the case when foreground objects have similar normal as background objects, avoiding the optimization process pushing them into the background.

Depth loss. The optimization process should respect the initial depth predicted by iDisc [39], meaning that the optimized depth should not deviate significantly from the initial estimation. Thus, we add a depth loss that punishes any depth change with respect to the estimated depth, shown as Eq. (3.8).

$$L_{depth} = \|\hat{d} - d_{est}\|_2^2 \tag{3.8}$$

In summary, the final loss we used to optimize depth is the weighted sum of each individual loss, shown in Eq. (3.9), where λ's are the weights for the loss terms.

$$L_{final} = \lambda_1 L_{normal} + \lambda_2 L_{continuity} + \lambda_3 L_{depth} \tag{3.9}$$

In our experiments, we applied grid search to determine those hyper-parameters and set $\lambda_1 = 1$, $\lambda_2 = 1$ and $\lambda_3 = 5$, more details can be found in Appendix B. With those differentiable loss terms, we optimize the predicted depth using the gradient-based optimizer Adam [21] for 1000 steps with a learning rate of 0.0001.

3.5 Mesh Post-processing Kernel

As explained in Section 3.4.2, the scene mesh reconstructed by Worldsheet [17] constitutes a single-layer mesh grid based on conventional depth map prediction. Consequently, its outcomes contain spurious faces and inaccurately link foreground and background objects within the invisible areas of the input image. To solve this issue, [59] proposed to predict the density field of the input image and map the location in the frustum to volumetric density in order to learn the real 3D feature. However, the performance of their method on complex scenes such as images in the ACDC [47] dataset is limited. To bridge the gap, we designed a Mesh Post-process Kernel that contains two steps: uncertain faces deletion and mesh completion.

3.5.1 Uncertain Faces Deletion

The first part of Mesh Post-process Kernel deals with the removal of spurious faces created by the Worldsheet [17]. Specifically, we first perform re-projection of all vertices from 3D to 2D and flagged vertices located within the uncertain area, as described in Section 3.4.2, as uncertain vertices. Furthermore, we defined faces that contain at least one uncertain vertex as spurious faces. Following this, we traverse through all faces and remove those that are identified as spurious faces. The resulting object contains the foreground and background object mesh separately, as well as point clouds between them.

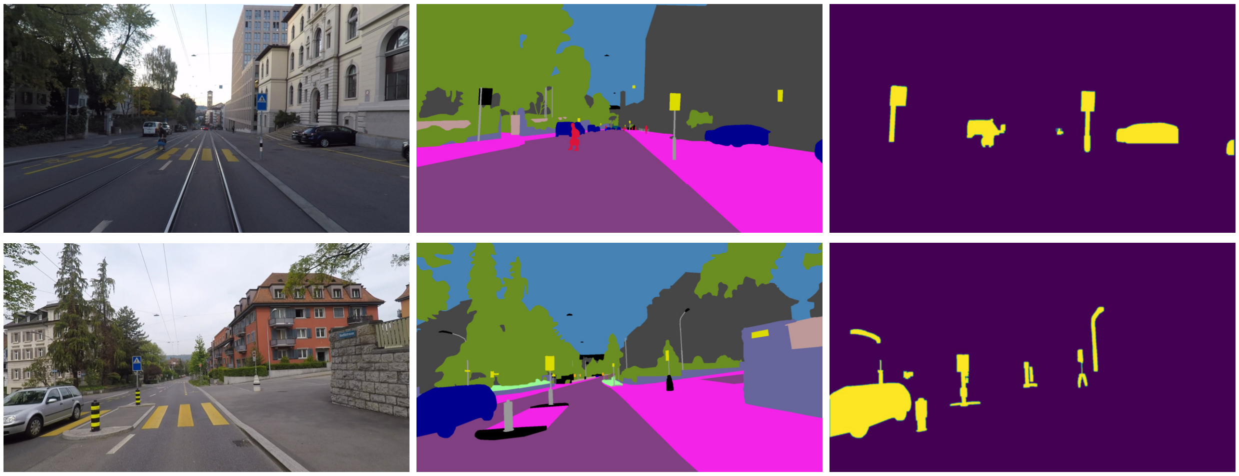

(a) RGB input (b) Semantic annotation (c) New uncertain region

Figure 3.6: **Completed uncertain region.** Column (**a**) shows the original RGB image, column (**b**) shows the semantic annotation and column (**c**) shows the completed uncertain region, where the uncertain region is represented by the yellow region.

3.5.2 Mesh Completion

The first part of the Mesh Post-process Kernel introduces holes in the mesh surface. The second part is designed to complete these holes and ensure the mesh surface becomes watertight. To achieve this, we first determine regions that need to be completed by finding the union of uncertain regions and their neighbouring foreground semantic segments. After that, we complete the mesh sheet by adding vertices into the newly determined uncertain region. In particular, for each newly determined uncertain region, we break it into horizontal lines with one-pixel height. Then adding linearly distributed vertices based on their left and right vertices location. In the end, we add faces between newly added vertices and boundary vertices to make the mesh sheet watertight. Fig. 3.6 shows an example of the newly determined uncertain region, and Algorithm 1 shows the operation process. Furthermore, to ensure the pixel-level accuracy of foreground objects, we also apply mesh completion to foreground objects. For the intersection of uncertain regions and foreground object semantics, we select a pixel that is adjacent to the persistent foreground semantics. Here, we assigned the vertex depth as the average of its neighbouring vertices. Following that, we extend this operation to all pixels within the intersection region by employing a breadth-first search approach. Lastly, we establish connections between the newly introduced foreground vertices and their corresponding persisting foreground mesh, effectively rendering each foreground object watertight.

3.6 Realistic Nighttime Scene Relighting

The second component of our pipeline aims to relight the reconstructed scene for realistic nighttime images based on material characteristics and nighttime light sources, which involves four steps. Though we did not implement this component in this thesis, we will explain the detailed method in the rest of this section. First of all, we utilize an inverse rendering network $\mathbf{F_{ir}}$ to predict the material characteristics, and then we will use a probabilistic light source activation to generate light sources. After that, we plan to apply the traditional rendering technique ray tracing to render clear linear nighttime images. In the end, we will also run post-processing to the clear nighttime image to simulate artifacts caused by the camera, including exposure time, noise and ISP. The remaining part of this section illustrates each step in detail.

Algorithm 1 Background Mesh Completion Algorithm

```
Require: Binary mask list $M$ indicating new uncertain maps, Incomplete mesh sheet $G_u$
Ensure: Each mask maps to a region that needs to be completed on the mesh sheet
for Every mask $m \in M$ do
    $I_t \leftarrow$ top row index of $m$ that has missing vertex
    $I_b \leftarrow$ bottom row index of $m$ that has missing vertex
    for $i = I_t$ to $I_b$ do
        $J_l \leftarrow$ index of the leftmost vertex row $i$ that is missing
        $J_r \leftarrow$ index of the rightmost vertex row $i$ that is missing
        $G_l \leftarrow G_u[i][J_l - 1]$
        $G_r \leftarrow G_u[i][J_r + 1]$
        for $j = J_l$ to $J_r$ do
            $G_u[i][j] \leftarrow \frac{G_l(j_r - j+1)+G_r(j-j_l+1)}{j_r-j_l+2}$    ▷ Add vertices uniformly between $G_l$ and $G_r$
        end for
    end for
end for
Connect vertices in $G_u$ to form faces    ▷ Make G watertight triangle mesh
```

3.6.1 Material Characteristics Prediction

Many prior studies have been focused on predicting material characteristics on small objects [24, 20] or indoor scenes [49, 60, 37, 27]. Nevertheless, only a few of them have made an attempt to tackle the same task with outdoor datasets [57]. In our work, we explored different existing methods and tested their performance on a synthetic outdoor optical flow dataset MPI Sintel [2]. Our testing results show that most indoor methods demonstrate good generalization capabilities on the outdoor dataset, likely due to the relatively small disparity between outdoor and indoor materials. Ultimately, based on these findings, we employe [27] as our inverse rendering network $\mathbf{F_{ir}}$ to predict albedo and roughness from the ACDC [47] RGB images. In the future, we will also test the performance of [57] once it is released and adapt it into our pipeline if it is proved to be better than [27].

3.6.2 Probabilistic Light Source Activation and Relighting

Unlike most scene relighting methods that consider sunlight as the only light source [33, 50, 62], our day-to-night simulation pipeline considers over 30 inactive light sources per scene (shown in Fig. 3.2). Thus, to generate realistic and diverse nighttime images, we design the probabilistic light source activation kernel to manage the activation of each light source. We define the activation of each light source as an independent random variable with a Bernoulli Distribution $f(\mathbf{p}, \mathbf{l})$, where $\mathbf{p}$ is the probability for event $\mathbf{l}$ to happen, meaning that there is a probability of p for the light source to be active and $(1-p)$ to be inactive. Furthermore, we also consider that light sources exhibit interdependence in the real world. For instance, adjacent windows are likely to be active or inactive simultaneously, front light and rear light of a moving car are also to be active together. To capture this real-life characteristic, we utilize group masks, as defined in Section 3.2, to group light sources together. This grouping approach ensures that each group of light sources shares the same $\mathbf{p}$ parameter yet remains independent from other groups. By incorporating this interdependency feature into our light source activation model, we can accurately emulate real-world light sources' behaviour and enhance our simulations' authenticity. Once the scene mesh, material attributes, and light sources are established, we perform ray tracing [58] to render the nighttime image. For each daytime image, we employ various activation parameters for the inactive sources, creating multiple nighttime images.

3.6.3 Image Post-processing

Nighttime images usually contain artifacts such as noise caused by low illumination and long exposure time at night. To simulate this real-world scenario, we will apply post-processing to clear linear images generated by ray tracing. Following [41], we plan to adopt the well-established heteroscedastic Gaussian model [8, 9, 29, 35] for noise. Given a nighttime clear image I_c, we will generate the nighttime noisy image I_n with the following equation:

$$\mathbf{I}_n = \mathbf{I}_c + \mathcal{N}\left(\mathbf{0}, \beta_1 \mathbf{I}_{\mathrm{c}} + \beta_2\right) \tag{3.10}$$

Where β_1 and β_2 are shot and read noise parameters, which we empirically determined based on measuring the noise of real noisy/clean nighttime image pairs for different ISO levels.

Chapter 4

Results and Experiment Plans

In this section, we will apply our pipeline to images within the ACDC dataset and present the results of the Geometry Mesh Reconstruction component. Additionally, we will demonstrate the generalizability of our pipeline by applying it to other autonomous driving datasets. such as the Cityscapes dataset [6]. Furthermore, we will outline our testing plans for the entire pipeline.

4.1 Geometry Mesh Reconstruction

4.1.1 Datasets and Metrics

Datasets In this work, we used images in the ACDC dataset [47], which is a large-scale dataset consisting of 4006 images evenly distributed across four different adverse conditions: snow, fog, rain and night. Each adverse condition image comes with a high-quality fine pixel-level semantic annotation and a reference image of the same scene taken under normal conditions (clear daytime). Our work mainly focused on applying our pipeline to the reference split of the ACDC dataset. Except for the ACDC dataset, we also use other autonomous driving datasets, including the BDD100K dataset [65], Cityscapes dataset [6] and the Dark Zurich dataset [46]. In particular, we use daytime images and its corresponding semantic annotations for all selected datasets, showing that our pipeline has a strong adaptability.

Metrics The Geometry Mesh Reconstruction component generates scene mesh and geometric information such as depth and surface normal. However, none of the datasets mentioned above provide ground truth for evaluation. Thus, we conduct a qualitative comparison of the generated mesh and its surface normals to evaluate the Geometry Mesh Reconstruction component of our pipeline.

4.1.2 Mesh Comparison

In order to evaluate the qualitative result of our reconstructed mesh, we compare our mesh reconstruction result with results from the following settings: **(1)** mesh constructed by the Worldsheet [17] using MiDaS v2.1 [42] as the external depth backbone based on input RGB. **(2)** mesh constructed by the Worldsheet using iDisc [39] as the external depth backbone. **(3)** mesh reconstructed by method presented in the SIMBAR [66] using Dense Prediction Transformer (DPT) monodepth models [43] as the external depth backbone, shown as Fig. 4.1. Moreover, we also compare surface normals of the generated mesh in Fig. 4.2, further showing that our method can estimate most of the geometric characteristics and reconstruct better mesh. Noted that for surface normal, we treated estimation of iDisc [39] as the ground truth. To show the effect of the mesh post-processing kernel, we compared our reconstructed scene mesh with **(2)** and **(3)** from different viewing angles, as shown in Fig. 4.3. We also present the optimization loss curve in Fig. 4.1.2. More qualitative results on scene mesh reconstruction are presented in Appendix C.

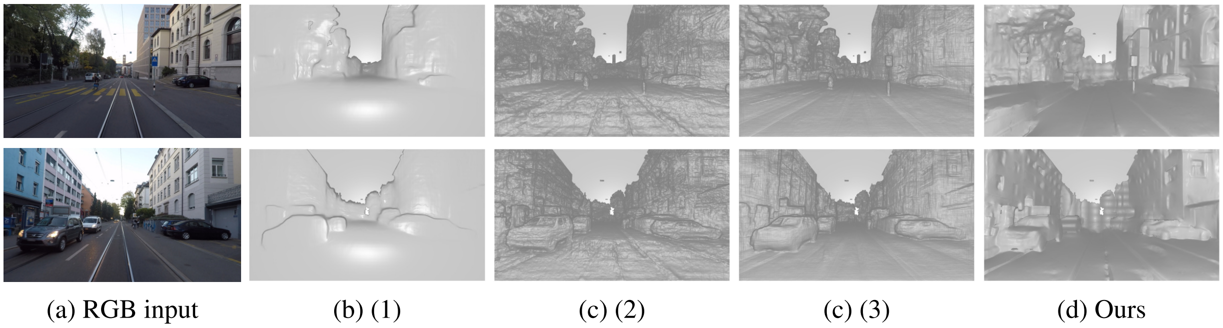


Figure 4.1: **Mesh reconstruction result comparison.** We compared the mesh reconstruction result with mesh reconstructed by other methods with the following settings, (1): Worldsheet with MiDaS depth, (2): Worldsheet with iDisc depth, (3): SIMBAR reconstruction. Our method can preserve more accurate geometric information and construct smoother mesh surfaces than all other methods.

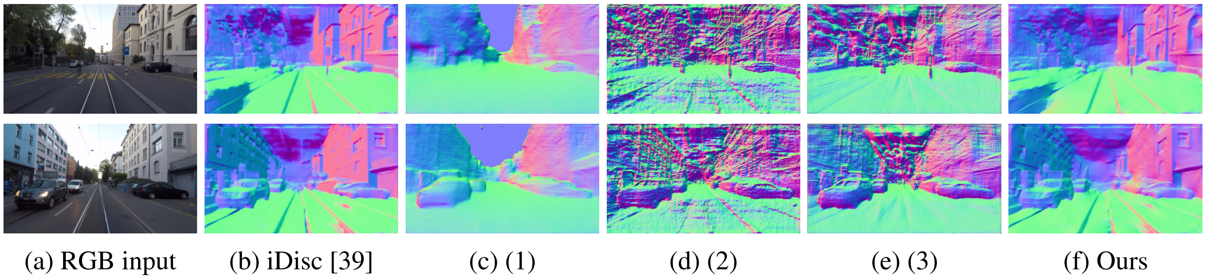


Figure 4.2: **Inferred surface normal comparison.** We compared the surface normal inferred from the reconstructed mesh with other methods (as stated in Fig. 4.1.2). Our reconstructed meshes have more accurate surface normal with respect to iDisc [39] surface normal.

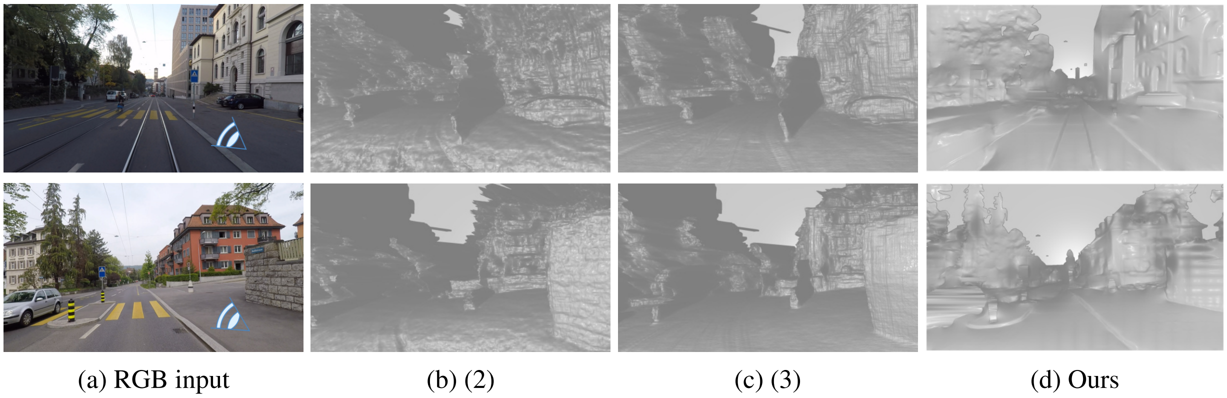
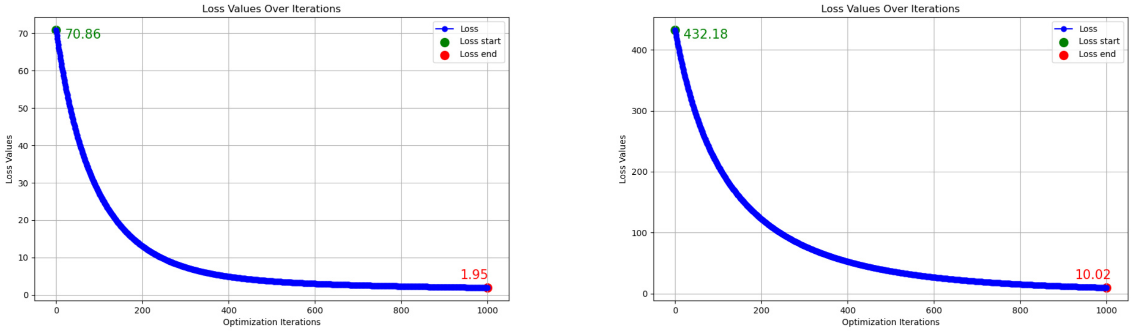


Figure 4.3: **Mesh comparison from different angles and background mesh.** We compared the mesh reconstruction result with mesh reconstructed by other methods (, as stated in Fig. 4.1.2.) with a 3 meters zoom in and a 45 degrees rotation clockwise as well as background mesh generated with our pipeline. The result shows that our method can efficiently break the unexpected connection between foreground and background objects and make the background mesh watertight.



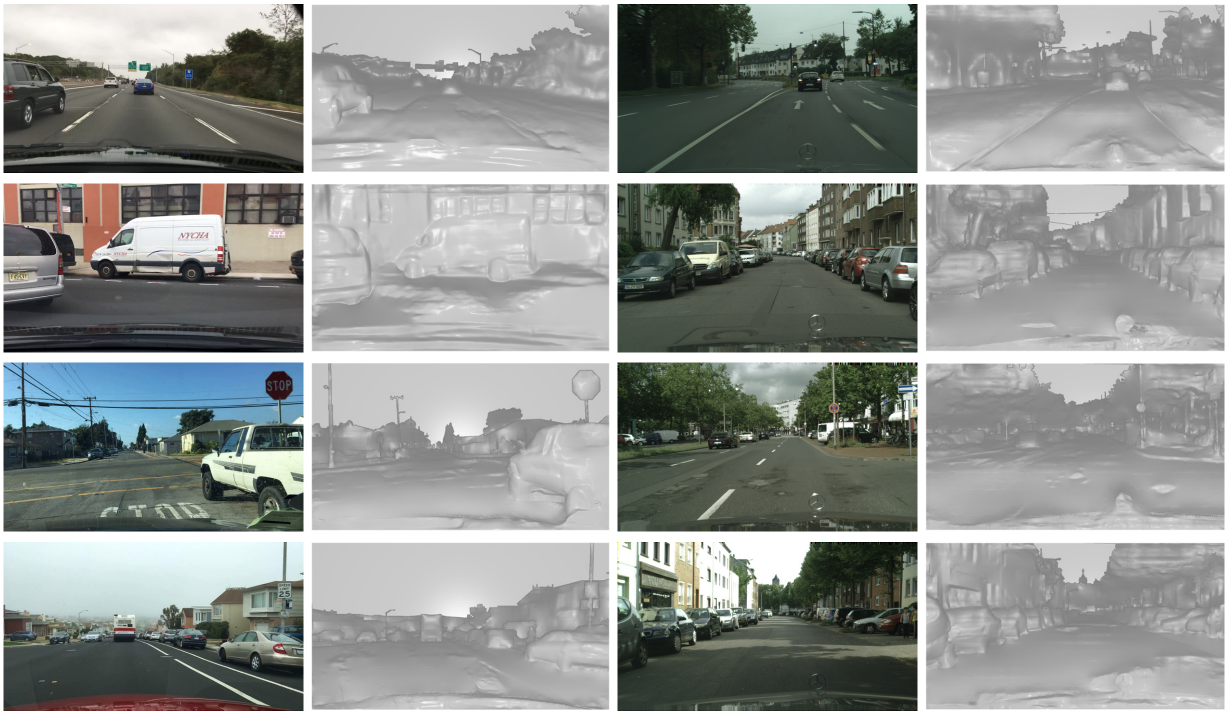
(a) Loss curve of upper image

(b) Loss curve of bottom image

Figure 4.4: **Optimization loss curve.** We present the optimization loss curve of two examples shown in Fig. 4.1.2.

4.1.3 Generalization to Other Datasets

Our pipeline can be generalized to a wide range of datasets, including the Cityscapes dataset [6], the BDD100K dataset [65] and the Dark Zurich dataset [46] daytime split. To demonstrate this on the Geometric Mesh Reconstruction component, we replace the input RGB $\mathbf{I_d}$ and semantic annotation $\mathbf{S}$ with samples from the new dataset. Fig. 4.5 illustrates the qualitative result of the scene mesh reconstructed from different datasets. We observe that our pipeline exhibits excellent generalization across various datasets.



(a)

(b)

(c)

(d)

Figure 4.5: **Mesh reconstructed from different datasets.** We showcase the mesh reconstruction based on two different datasets: BDD100K dataset [65] (column (a) and (b)) and Cityscapes dataset [6] (column (c) and (d)). The above results demonstrate the capability of our pipeline to generate reasonable scene mesh when applied to other datasets. However, we notice that roads close to the camera are not correctly reconstructed, this is a result of inaccurate surface normal prediction caused by the ego vehicle.

4.2 Testing Plans

4.2.1 Datasets and Metrics

Datasets For the experiment of the entire pipeline, we will use the ACDC dataset [47] described in Section 4.1.1. In particular, we will apply our pipeline to reference images in which semantic annotations are available and conduct evaluation on the nighttime split testing set. The evaluation of the ACDC dataset is done via an online server.

Metrics The goal of our experiment is to show that our synthetic nighttime images can serve as training data and improve the performance of current segmentation detection methods. To evaluate the retrained model, we will use the standard semantic segmentation evaluation method mean Intersection of Union (mIoU) and the Uncertainty-Aware semantic segmentation Intersection of Union (AUIoU) introduced by [47]. The main distinguishing feature of those methods is the incorporation of image regions that possess indiscernible semantic content, referred as ”invalid regions” during the process of annotation and evaluation.

4.2.2 Experiments

Our experiments aim to demonstrate the superiority of our generated nighttime images over the original images and other synthetic nighttime images for the task of semantic segmentation. To achieve this, we will conduct a comprehensive evaluation by training various state-of-the-art methods and architectures using different versions of nighttime images and then evaluate them using the ACDC dataset nighttime testing set. In particular, for the training data, we plan to use the original daytime reference images as the baseline, comparing it with dimmed daytime images, CycleGAN [67] transferred nighttime images, our nighttime images, as well as the real nighttime images in the ACDC dataset [47]. To ensure robustness and accuracy, we will train the model with various methods including DeepLabV3+ [4], SegFormer [61], Mask2Former [5], and HRNet [52], as well as architectures including ResNet [15], Swin Transformer [30] and Vision Transformer [7]. Moreover, for each combination listed above, we also plan to train them with different sizes of training datasets with nighttime images generated using our pipeline, showing that our pipeline is able to generate multiple nighttime images from one single daytime image by activating different light sources combinations described in Section 3.6.2. We will report the mIoU and AUIoU of all methods on the nighttime split of the ACDC dataset, as well as the IoU on each semantic class. Additionally, we will also present the qualitative comparison of selected semantic segmentation methods, trained using different training datasets listed above, on the ACDC dataset nighttime split. By presenting these qualitative results, we aim to better understand how different datasets impact the segmentation results and pinpoint the strengths and weaknesses of nighttime images generated using our pipeline.

Chapter 5

Conclusion

In this thesis, we have presented a physics-based pipeline NPSim that performs day-to-night transformation with two components: Geometric Mesh Reconstruction and Realistic Nighttime Scene Relighting. This work stands apart from all prior works as it is the first to accomplish the task of relighting outdoor scenes from day to night using only a single image. The distinctiveness of our method lies in its explicit estimation of the scene's geometry and materials, and then integrate the estimated materials into the geometry alongside the light sources. The innovative aspect is the consideration of light sources that remain inactive during the daytime but become active at night. By incorporating these elements into the relighting process, this work has the potential to achieve a remarkable advancement in generating realistic night scenes from daytime photographs, setting it apart from all previous research in this field.

Our mesh reconstruction component reconstructs better scene mesh by preserving geometric information such as depth and surface normal. It also gets rid of the potential unrealistic reflection by removing spurious faces between foreground and background objects. Moreover, the proposed photo-realistic nighttime simulation is a general approach that can be applied to any daytime driving dataset for day-to-night simulation without the need for real nighttime data. This alleviated the need to annotate large sets of real nighttime images and made a significant contribution to constituting a bottleneck for nighttime semantic scene understanding.

Meanwhile, we are also aware of several limitations of our method. Firstly, the generation of inactive light source masks is not automated and requires some human input. This problem could be alleviated by training a neural network using existing annotations to detect inactive light sources from daytime images. Secondly, our mesh reconstruction depends heavily on the estimated geometric information, errors on estimated depth and surface normal will make the reconstructed mesh inaccurate. We noticed that our current depth prediction model is not capable of correctly predicting the surface normal of some parts of the road, especially roads that are far away or under shadow. This is probably caused by domain shift between Diode dataset [54], NYUv2 dataset [38] and the ACDC dataset [47]. We plan to fix it by trying out different models or retraining the current model on a dataset that is closer to the ACDC dataset. Lastly, our relighting component did not consider motion blur caused by long exposure time at night, this may cause our generated nighttime images to be inconsistent with real nighttime images and lower the performance of trained models. Our future works will focus on the simulation of motion blur for moving objects and to narrow the gap between the simulated and real nighttime images.

Appendix A

Light Source Annotation Examples

In this section, we present the appearance of each inactive light source class in both daytime and nighttime, shown in Table A.1. Some common examples are omitted. We also provide more visual examples of our annotation, shown in Fig. A.1.

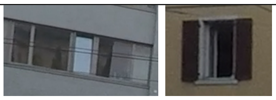
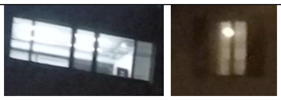
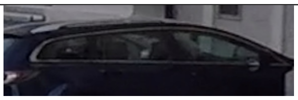
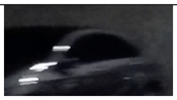
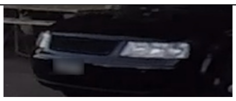
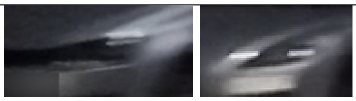
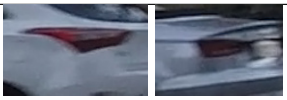
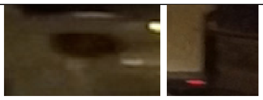
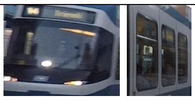
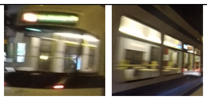
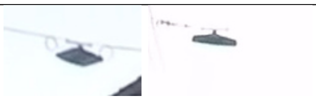
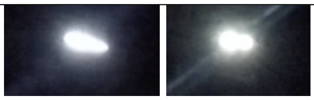
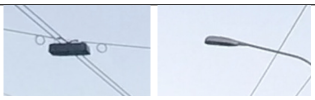
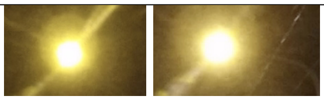
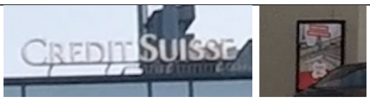
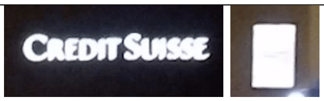
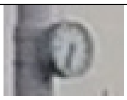
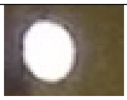
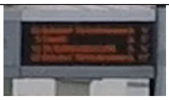
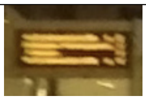
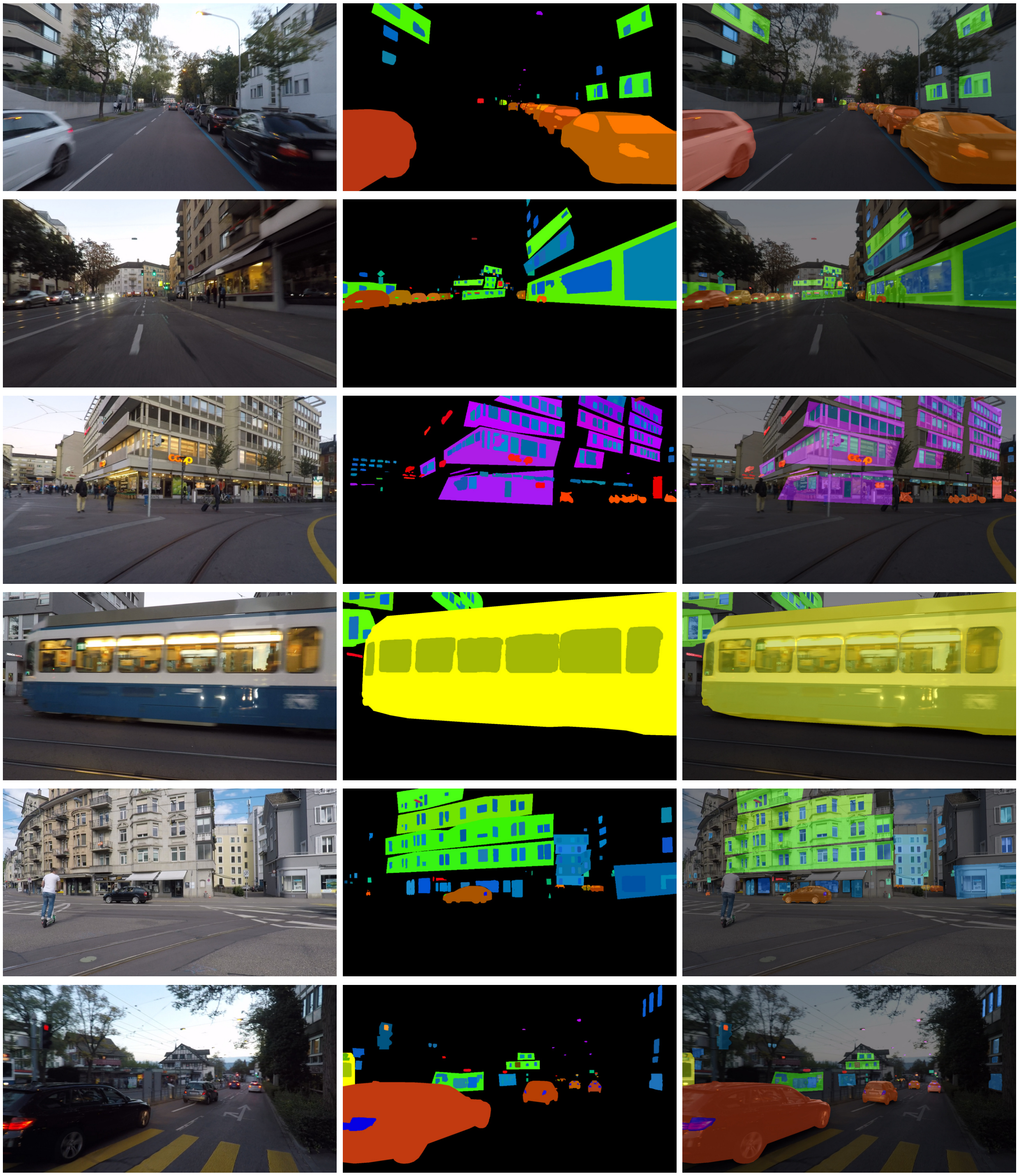
Light source class	Daytime appearance	Nighttime appearance
window_building		
window_parked		
parked_front		
parked_rear		
window_transport		
street_light_HT		
street_light_LT		
advertisement		
clock		
inferred		

Table A.1: **Inactive light sources examples.** We present examples of different types of inactive light sources. The second and third columns show the daytime appearance and nighttime appearance of each example, respectively, arranged in the same order.



(a) Original RGB image

(b) Inactive light source mask

(c) Superposition of (a) and (b)

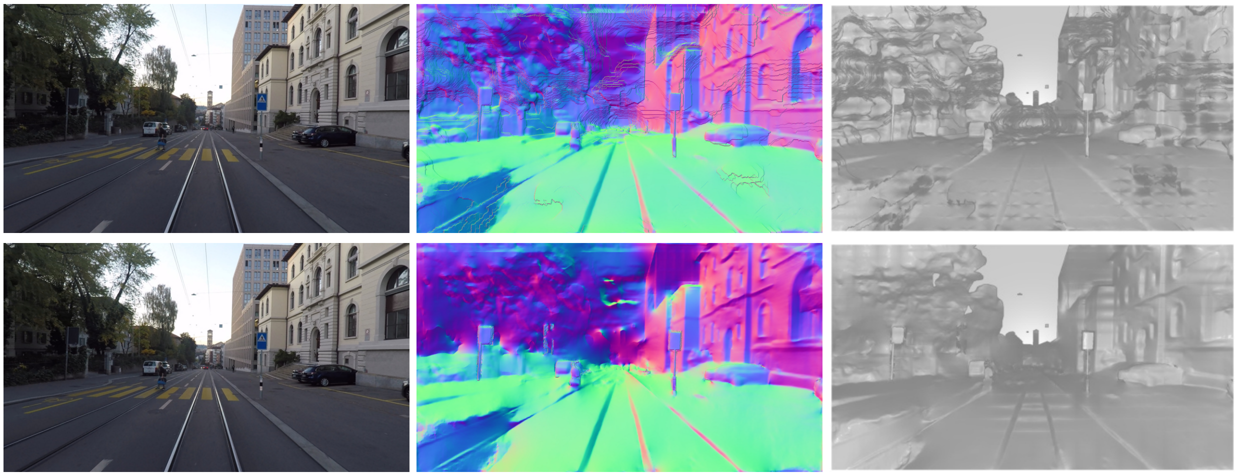
Figure A.1: **More Annotation Examples.** We present more examples of our inactive light source annotation. The left column shows the original RGB image, the middle column shows the annotated inactive light source mask, where each instance has its own identity and bounding box, and the right column superimposes the RGB image and the light source mask.

Appendix B

Hyper-parameter Tuning

To find the optimal choice of hyper-parameters used in the depth refinement kernel, we apply grid search to decide $\lambda_1 \sim \lambda_3$ in Eq. (3.9). In our experiment, we set λ_1 to 1 and then changing λ_2 and λ_3 between 10^{-3} and 10^{3}.In particular, we select 7 values for each parameter: 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}. We tried out 49 different combinations in total. Next, we will present the effect of each loss term on mesh reconstruction by showing qualitative results of reconstructed mesh and optimized surface normal.

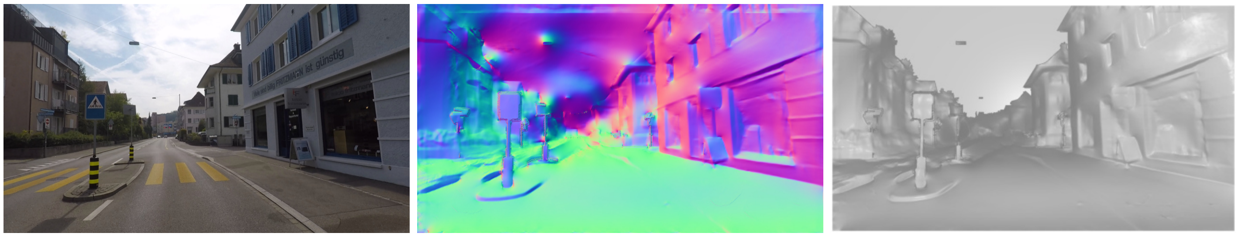
During our experiments, the coefficient of continuity loss λ_2 plays a more important role. When setting λ_2 small, gaps caused by sudden changes in depth will be created in both refined surface normal and reconstructed mesh. Those gaps will not violate other loss terms much but will have a significant negative impact on relighting. The upper row of Fig. B.1 shows an example of gaps on surface normals and reconstructed mesh. Similarly, having λ_2 too large will also cause some negative effects, as shown in the lower row of Fig. B.1. Though it doesn't have a negative effect on the refined surface normal, the final reconstructed mesh will have some wave-like artifacts, this kind of effect is mainly caused by the dual-reference cross-bilateral filter during depth refinement.



(a) Original RGB image (b) Refined surface normal (c) Reconstructed mesh

Figure B.1: **Small λ_2.** We present an example of surface normal and final mesh with different λ_2 values. The upper row shows an example of setting λ_2 too small and the lower row shows an example of setting λ_2 too large.

Compared to λ_2, λ_3 (the coefficient of depth loss) plays a less important role. We discover that making λ_3 too large will not affect the reconstruction result much. However, having it too small will make



(a) Original RGB image (b) Refined surface normal (c) Reconstructed mesh

Figure B.2: **Large λ_3.** We present an example of surface normal and final mesh when setting λ_3 too large.

the refined depth deviate more from the originally predicted depth, especially for foreground objects as they are usually surrounded by uncertain regions. This will further result in wider uncertain regions and remaining unexpected faces after after mesh post-processing component. We present one visual example in the Fig. B.2.

Similarly, the effect of λ_1 (the coefficient of normal loss) will largely depend on λ_2 and λ_3 together. When setting both λ_2 and λ_3 large, the result will have a combined effect of large λ_2 and large λ_3, with wider uncertain region and wave-like visual artifacts. When setting both λ_2 and λ_3 smaller, we observed similar effects as shown in the upper row of Fig. B.1. In the end, we set $\lambda_1 = 1$, $\lambda_2 = 5$ and $\lambda_3 = 1$

Appendix C

Mesh Reconstruction Examples

In this section, we present more qualitative examples of the reconstructed mesh and its corresponding surface normal, shown in the figure below. Mesh reconstructed using our method has smoother surface normal and more detailed geometry information.

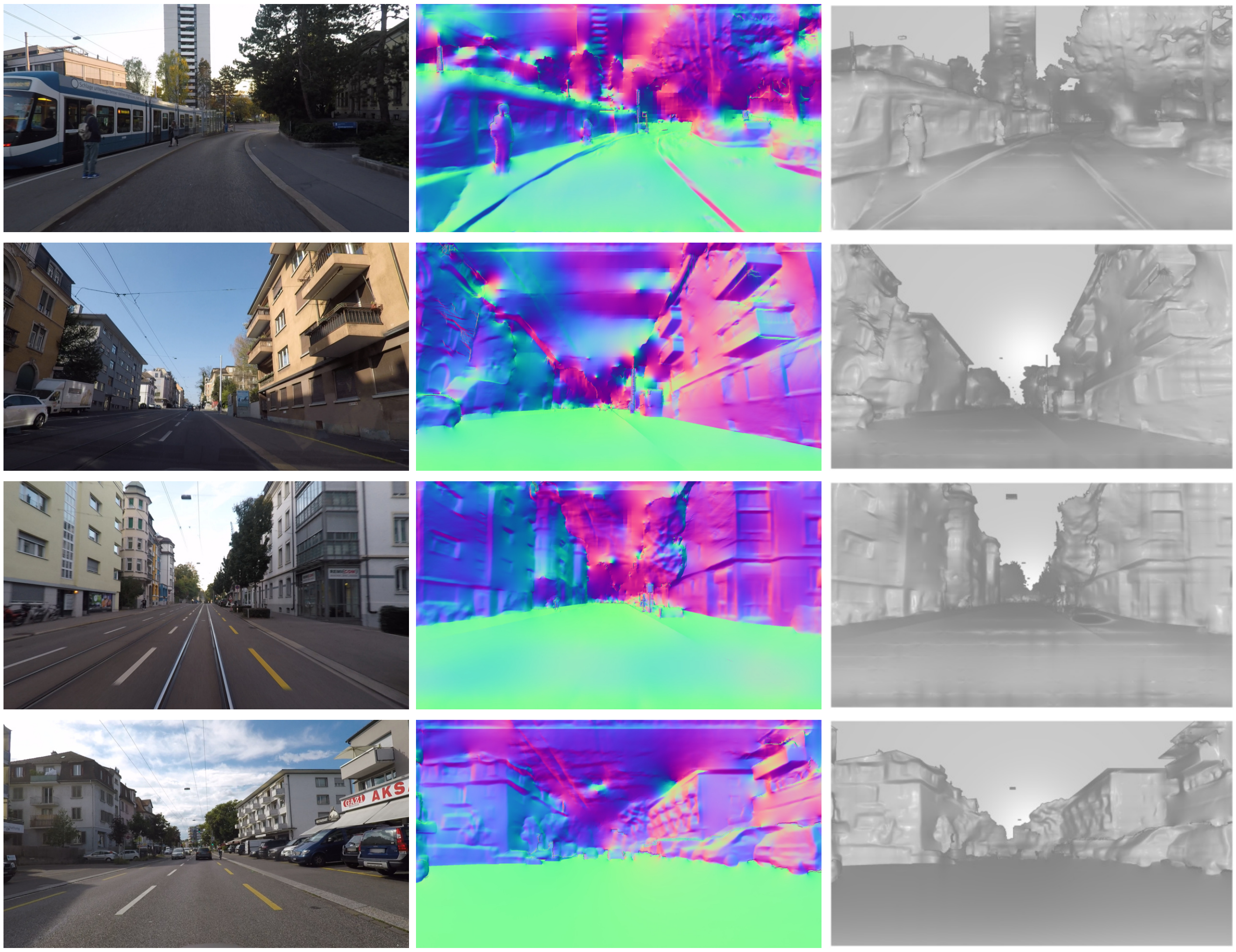
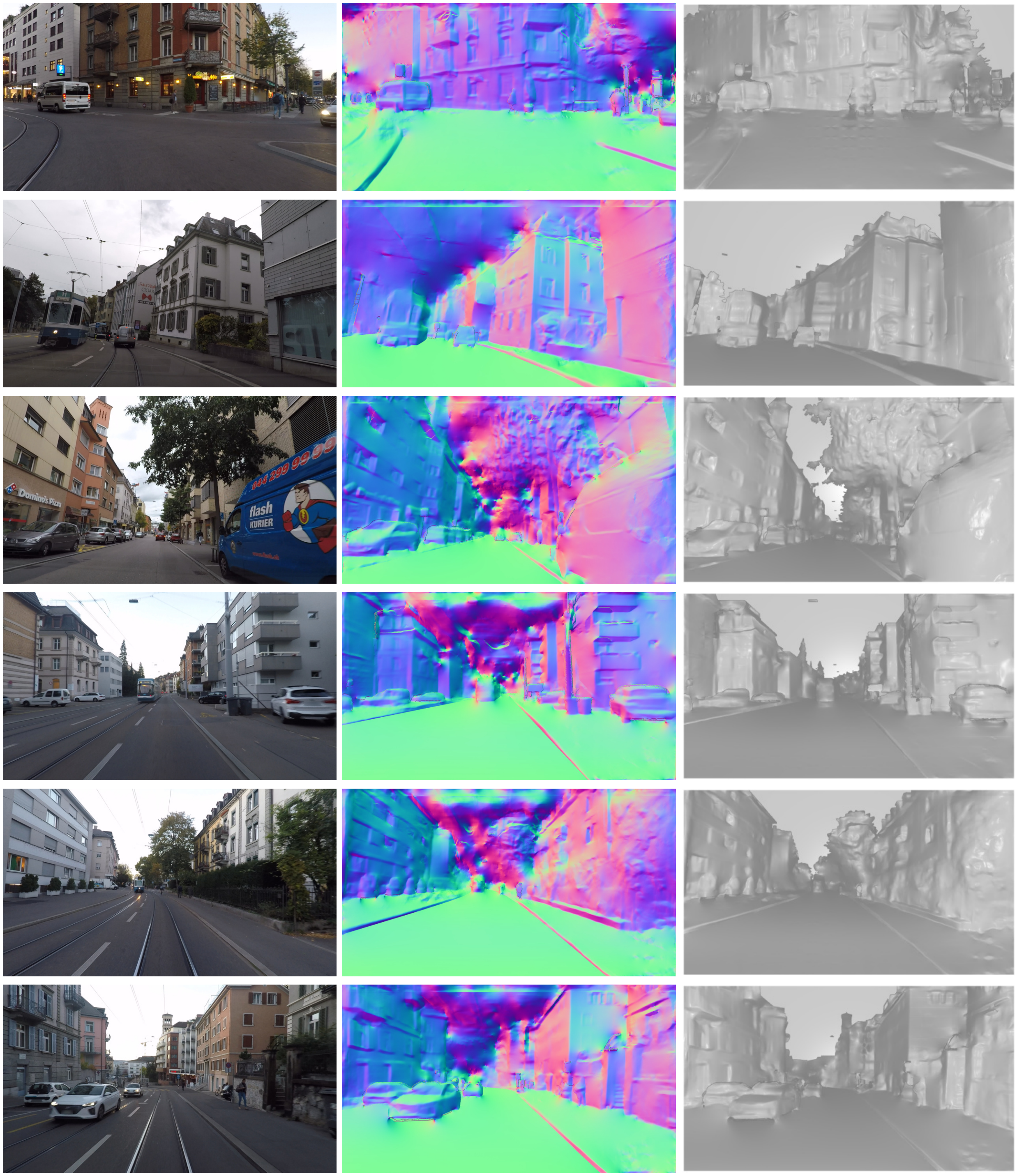

(a) RGB input (b) Corresponding surface normal (c) Reconstructed mesh

Figure C.1: **More mesh reconstruction examples.** We present more examples of mesh reconstructed using our method and their corresponding surface normal.



(a) RGB input (b) Corresponding surface normal (c) Reconstructed mesh

Figure C.2: **More mesh reconstruction examples.** We present more examples of mesh reconstructed using our method and their corresponding surface normal.

Appendix D

Model Selection

In this section, we present the selection process of network $\mathbf{F_g}$ and $\mathbf{F_ir}$. For network F_g, we directly use the iDisc depth model pre-trained on the KITTI dataset [13]. However, for the surface normal prediction, as for our own knowledge, all previous works have focused on indoor datasets such as the NYUv2 dataset [38]. To overcome the domain gap between indoor and outdoor scenes, we retrained the iDisc surface normal network on the Diode dataset [54] outdoor split. To maximize the performance, we tried several training settings, with their evaluation results shown in Table D.1.

Setting	rmse_angular	a_1(<5deg)	a_2(<11.5deg)	a_3(<22.5deg)	a_4(<30deg)
iDisc NYUv2	60.181	0.037	0.184	0.312	0.387
iDisc NYUv2+Diode	77.862	0.044	0.130	0.212	0.281
iDisc Diode	**44.874**	**0.290**	**0.435**	**0.563**	**0.625**

Table D.1: **iDisc normal estimation retrain result**. We present three different settings of our networks and their corresponding evaluation result.

Bibliography

[1] Labeling platform for robotics and av — segments.ai.

[2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, October 2012.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[5] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] Alexey DosoViTskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Alessandro Foi. Clipped noisy images: Heteroskedastic modeling and practical denoising. *Signal Processing*, 89:2609–2629, 12 2009.

[9] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008.

[10] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[11] Leon Gatys, Alexander Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv*, 08 2015.

[12] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, oct 2020.

[15] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

[17] Ronghang Hu, Nikhila Ravi, Alexander C. Berg, and Deepak Pathak. Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[19] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *IEEE Transactions on Image Processing*, 30:2340–2349, 2021.

[20] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, June 2023.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[22] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[23] Attila Lengyel, Sourav Garg, Michael Milford, and Jan C. van Gemert. Zero-shot day-night domain adaptation with a physics prior. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4399–4409, October 2021.

[24] Chenhao Li, Trung Thanh Ngo, and Hajime Nagahara. Inverse rendering of translucent objects using physical and neural renderers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12510–12520, June 2023.

[25] Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. Mine: Towards continuous depth mpi with nerf for novel view synthesis. In *ICCV*, 2021.

[26] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[27] 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.

[28] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, July 2017.

[29] Xinhao Liu, Masayuki Tanaka, and Masatoshi Okutomi. Practical signal-dependent noise parameter estimation from a single noisy image. *IEEE Transactions on Image Processing*, 23(10):4361–4371, 2014.

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[31] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[32] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[33] Linjie Lyu, Ayush Tewari, Thomas Leimkuehler, Marc Habermann, and Christian Theobalt. Neural radiance transfer fields for relightable novel-view synthesis with global illumination. In *ECCV*, 2022.

[34] Will Maddern, Geoffrey Pascoe, Matthew Gadd, Dan Barnes, Brian Yeomans, and Paul Newman. Real-time kinematic ground truth for the oxford robotcar dataset. *arXiv preprint arXiv: 2002.10152*, 2020.

[35] Markku Makitalo and Alessandro Foi. Optimal inversion of the generalized anscombe transformation for poisson-gaussian noise. *IEEE Transactions on Image Processing*, 22(1):91–103, 2013.

[36] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[37] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting Triangular 3D Models, Materials, and Lighting From Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8280–8290, June 2022.

[38] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[39] Luigi Piccinelli, Christos Sakaridis, and Fisher Yu. idisc: Internal discretization for monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[40] Marius-Constantin Popescu, Valentina Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8, 07 2009.

[41] Abhijith Punnappurath, Abdullah Abuolaim, Abdelrahman Abdelhamed, Alex Levinshtein, and Michael S. Brown. Day-to-night image synthesis for training nighttime neural isps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[42] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021.

[43] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022.

[44] Viktor Rudnev, Mohamed Elgharib, William Smith, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. Nerf for outdoor scene relighting. In *European Conference on Computer Vision (ECCV)*, 2022.

[45] Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In *European Conference on Computer Vision (ECCV)*, pages 707–724, 2018.

[46] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[47] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

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

[49] 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. In *International Conference on Computer Vision (ICCV)*, 2019.

[50] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2021.

[51] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*, 2021.

[52] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[53] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[54] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR*, abs/1908.00463, 2019.

[55] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Mathieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019.

[56] Zhonghao Wang, Mo Yu, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-mei Hwu, Thomas S. Huang, and Honghui Shi. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[57] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8370–8380, June 2023.

[58] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, jun 1980.

[59] Felix Wimbauer, Nan Yang, Christian Rupprecht, and Daniel Cremers. Behind the scenes: Density fields for single view reconstruction. *arXiv preprint arXiv:2301.07668*, 2023.

[60] Haoqian Wu, Zhipeng Hu, Lincheng Li, Yongqiang Zhang, Changjie Fan, and Xin Yu. Nefii: Inverse rendering for reflectance decomposition with near-field indirect illumination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4295–4304, June 2023.

[61] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[62] Siqi Yang, Xuanning Cui, Yongjie Zhu, Jiajun Tang, Si Li, Zhaofei Yu, and Boxin Shi. Complementary intrinsics from neural radiance fields and cnns for outdoor scene relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16600–16609, June 2023.

[63] Y. Yang and S. Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[64] Yusaku Yoshida, Ryo Kawahara, and Takahiro Okabe. Light source separation and intrinsic image decomposition under ac illumination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5735–5743, June 2023.

[65] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[66] X. Zhang, N. Tseng, A. Syed, R. Bhasin, and N. Jaipuria. Simbar: Single image-based scene relighting for effective data augmentation for automated driving vision tasks. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3708–3718, jun 2022.

[67] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.