Training Deep Networks with Material-Aware Supervision

Tiancheng Zhi

CMU-CS-21-138

September 2021

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Srinivasa G. Narasimhan (Co-Chair)
Martial Hebert (Co-Chair)
Matthew P. O'Toole
Sing Bing Kang (Zillow Group)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was sponsored by the Heinz Endowment, the Chemimage Corporation, Facebook, Zillow, the National Science Foundation under award numbers CNS1446601, IIS1730147, and ECCS2038612, the Department of Transportation under award numbers DTRT13GUTC26 and 69A3551747111, the Office of Naval Research under award number N00014-15-1-2358, and a University Transportation Center T-SET grant. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Deep Networks, Supervision Signal, Material

For my parents

Abstract

Deep learning is a strong tool for predicting scene properties from images. Typical supervised methods require large scale real data with ground truth, which is hard to obtain. This situation demands techniques with little ground truth real data.

Without annotations, an apparent question is: Where does the supervision signal come from for training deep networks? In this thesis, we demonstrate that the awareness of materials provides such easy-to-obtain signals. We also present a framework that can be used for different tasks to exploit material-aware supervisions.

We consider four forms of supervision signals in the framework: ground truth and photometric supervisions from appearance models, and adversarial and confidence supervisions from appearance locations. Specifically, given a task, an approximate appearance model can be built to describe the whole or part of the scene. With this model, we could render synthetic images for ground truth supervision or optimize the networks using photometric supervision. The scene may also contain spatially-varying materials providing additional appearance location information. Such information can be used for separating special appearances using adversarial supervision, or fixing failure cases using confidence supervision.

We present four applications to demonstrate the effectiveness of the proposed framework. In the first application, we introduce an approach for fine-grained recognition of powders on complex backgrounds, to provide an example of synthetic ground truth supervision from translucent material awareness. We build a blending model for synthesizing images of translucent powders on various backgrounds. As a second contribution, we demonstrate a method for recovering human texture and geometry from an RGB-D video, as an example of photometric supervision from Lambertian material model. In the third task, we propose a floor appearance decomposition approach for realistic object insertion, as an example of adversarial supervision for diffuse-specular separation and direct sunlight detection. We obtain coarse locations of specular and sunlight appearances based on layout geometry and the awareness of emissive and transparent materials. Lastly, we present a cross-spectral stereo matching method for road scenes, to show that the confidence supervision from non-Lambertian appearance locations helps fix regions of failure.

We believe that the method proposed in this thesis can be used in more real applications, including interior design, medical imaging, and autonomous driving, especially when ground truth real data are not easy to obtain.

Acknowledgments

I would like to render my warmest thanks to my advisors Srinivasa Narasimhan and Martial Hebert for their unwavering support and invaluable guidance during the five years. Srinivas encouraged me to embrace new challenges which have not been solved. From him I learnt how to define and handle an important problem that is often ignored by other researchers. Martial provided insightful comments and advice on my work, pointing out critical aspects that were ignored. From him I learnt to think problems outside the box and view things from different perspectives.

I'd also like to express my great gratitude to the rest of my thesis committee: Sing Bing Kang and Matthew O'Toole. Their insightful comments and discussions on my thesis helped me complete this thesis. Sing Bing provided ingenious guidance for my projects and detailed suggestions for the writing. Matt's advice on the thesis is always constructive, practical, and helpful.

I appreciate my mentor and collaborators during my internship at Facebook: Minh Vo, Carsten Stoll, Tony Tung, and Christoph Lassner, for helping me complete the intern project and get the ECCV paper accepted. I'm also thankful to other collaborators during my PhD research: Bernardo Pires and Ivaylo Boyadzhiev. I still remember Bernardo and I went out for data collection at 1am, driving a Jeep on Pittsburgh roads. Without their help, I could not complete my projects.

I also want to thank other faculty members at CMU: Yannis, Aswin, and Tai Sing. I had a nice TA experience with Yannis. Aswin provided helpful comments on my projects in the imaging group meetings. Tai Sing cared about my status and progress. From him I gained a lot of support during the five years.

I would like to thank my ILIM labmates: Chao, Shumian, Dinesh, Robert, Suren, Jian, Joe, Adithya, Bowei, Supreeth, Mark, Sid, Gaurav, Raaj, Yifei, Tiffany, Sharon, Xudong, Mengqing, Fangyu, Zhiyu, Bo, Manchen, Geng, for helping me with projects and making life in ILIM enjoyable. Special thanks to Supreeth for giving me strong help with handling cameras when I joined ILIM with zero hardware knowledge. I also thank other friends inside or outside CMU: Chaoyang, Xiaofang, Chen, Yanzhe, Ziqiang, Tianshi, Xingyu, Shuqi, Tian, Yimeng, Junjue, Yufei, Donglai, Wenxuan, Chenchen, Wen, Zechun, Yi, Yang, Bole, Xiaopeng, Minghan, Shan, Heshan, Qianhao, Ji, Haoyang, Daoyu, Rui, Xunyu, Junbang, and Zhe, for sharing happiness and listening to my sadness.

Last but definitely not least, I would like to thank my parents and other family members for their consistent and unparalleled support during my PhD journey. I would also like to give a special thank to Ruiyu for the encouragement during hard times when I was feeling down. I could not complete the thesis without such strong support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Deep learning is a strong tool for predicting scene properties from images. It is usually more effective than classical techniques, because it establishes associations that could be not described by classical models through large sampling of observations, especially for ill-posed problems. Typical supervised deep learning methods require a large amount of real data with ground truth labels. However, obtaining ground truth on real data is hard: it requires additional measurements or human annotations, which are manually intensive and cost prohibitive. This situation demands techniques for training deep networks with little ground truth real data.

Without ground truth annotations, an apparent question is: *Where does the supervision signal come from for training deep networks?* Existing works exploit geometry, reflectance, semantics, shape, motion and image priors for supervising or regularizing the training of neural networks. In this thesis, we demonstrate that the awareness of materials can also provide such supervision signals, which are usually easy to obtain. Such signals may partially overlap with the aforementioned priors but are not completely the same. We also present a framework that can be applied to different tasks to exploit material-aware supervisions.

In the following sections, we first introduce the definition of the material awareness and material types we study in this thesis (Sec. 1.1), and then explain the motivation, opportunities and challenges of incorporating material awareness (Sec. 1.2). Next, we review the common sources of supervision signals that are used for training deep networks with little ground truth real data (Sec. 1.3). Finally, we present our contributions, including a unified framework and a brief summary of its four applications (Sec. 1.4).

1.1 Definition of Material-Awareness

We define materials as the qualitative properties describing how objects interact (emit, transmit, reflect) with light. Typical examples include diffuse, specular, transparent, translucent, and emissive materials. Specifically, we define light sources as "emissive materials". Although light sources are usually treated as illumination conditions rather than as materials, we include it in our list of material types. Here, we focus on the property of the source objects that they "emit light", rather than the property of the emitted light. It is similar to how we treat other materials.

Specifically, the knowledge of emissive materials comes either directly from the observation

(a) Diffuse (b) Specular/Glossy

(c) Transparent (d) Translucent (e) Emissive

Figure 1.1: Examples of material types [32, 181]. It is easy for human to roughly perceive the qualitative material of common objects. Such approximate material awareness provides supervision signals for training deep networks.

of light sources or indirectly from the observation of the consequence of the emissive material casting light to the scene. In the former case, the emissive material is usually directly detected or identified; for the latter case, the knowledge of emissive materials is inferred based on the appearance due to scene-light interaction.

As shown in Fig. 1.1, it is easy for humans to roughly perceive the qualitative visual properties of materials: cloth is diffuse, glass is specular and transparent, metal is glossy, headlights and LEDs are emissive, and makeups are translucent. We define the knowledge of these material types for the whole or a part of a scene as "material awareness". This approximate material awareness provides abundant information that may be exploited as supervision signals for training deep networks. For example, knowing that a material is translucent, we should take the background color into consideration when modeling its appearance. For diffuse materials, we can calculate shading by simply applying Lambert's cosine law. If a region is specular, we know that any method assuming constant intensity across views is unreliable. Given the location of glass doors and windows, we can roughly predict the direct sunlight region on the floor for a specific sun direction. In this thesis, we use such information to supervise the training of deep networks.

1.2 Motivation and Challenges

As shown in Fig. 1.2, complex materials are everywhere. On common roads, the ground is mostly diffuse while the vehicle surfaces are often specular. The headlights can create strong reflections.

(a) Road Scene (b) City View [2] (c) Mirror [232] (d) Glass

(e) Empty Room [43] (f) Human (g) Translucent Powders

Figure 1.2: Complex materials in the world. (a) Road scene with glass windshields, glossy car surfaces, blinking LED lights; (b) Big metallic landmark in a city; (c) Floor mirror reflects indoor scenes; (d) Glass tumblers with complex geometry-related appearance; (e) Empty room with strong specular reflection and direct sunlight (from transparent windows) on the floor; (f) Human image with skin, clothing, and a specular tag on the clothing; (g) Powder samples with different thicknesses.

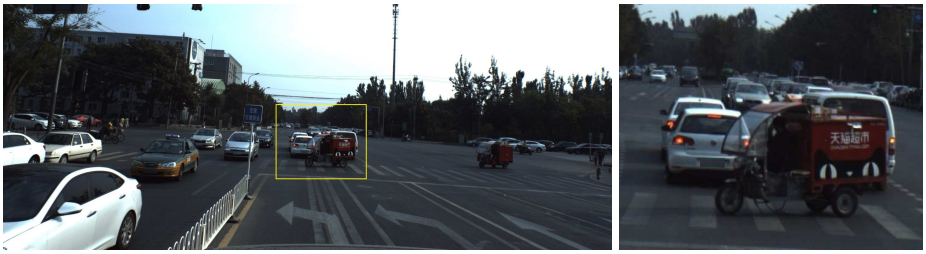
The LED tail lights are blinking. Through the glass windshield we can see the seats inside the car. The windshield itself can also reflect the sky. In a city, the buildings consist of diffuse cement and glass windows. Some landmarks could even be metallic and reflect the surrounding scene. In indoor scenes, emissive (lamps), transparent (windows), specular (mirrors, floors), and diffuse (walls) materials create complex appearances. Fig. 1.2 also provides other examples, including the glass tumblers with complex geometry-related appearance, the human clothing with a specular tag, and translucent powder samples with different thicknesses.

It is hard for computer vision algorithms to analyze those complex scenes without the awareness of materials. Below we provide several challenging cases to demonstrate the importance of material awareness:

Road Scenes: Fig. 1.3 provides two examples when material awareness is critical for understanding road scenes. Without knowing the specular property of the bus windows, the algorithm may confuse between the real cars and the reflected cars. The occlusion by a transparent windshield is also a challenging case, because the algorithm needs to understand that the scene transmitted through the windshield actually reveals the information about the car behind it. These cases may lead to the failure of depth estimation and object detection algorithms.

Glass Surfaces: Fig. 1.4 provides examples when computer vision algorithms fail without knowledge of glass surfaces. In the first example, the depth estimation algorithm [128] recovers the depth of the transmitted scene rather than the depth of the glass window itself. In the second example, the specular reflection removal algorithm [230] incorrectly removes texture or shading

(a) Specular Reflection on Windows



(b) Occlusion by Transparent Windshield

Figure 1.3: Challenging cases on roads [76]. (a) The bus window reflects surrounding cars. (b) The transparent windshield of the pedicab occludes the car. These cases may lead to failure of depth estimation and object detection algorithms.

outside the glass region.

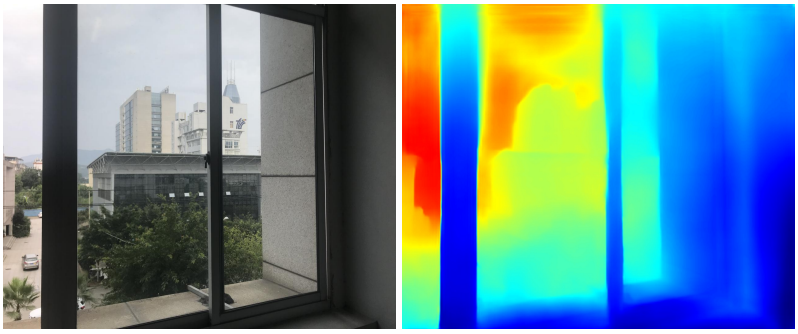
Mirror: Fig.1.5 shows an example [232] in NYUDv2 dataset [190] where the depth map captured using Kinect is incorrect in the mirror region. It captures the depth of the reflected scene rather than the mirror itself.

Screen: Fig.1.6 provides an example [136] when depth estimation fails on a specular laptop screen. Because the laptop screen reflects a scene that is far away, the algorithm overestimates its depth value, resulting in a hole in the 3D visualization.

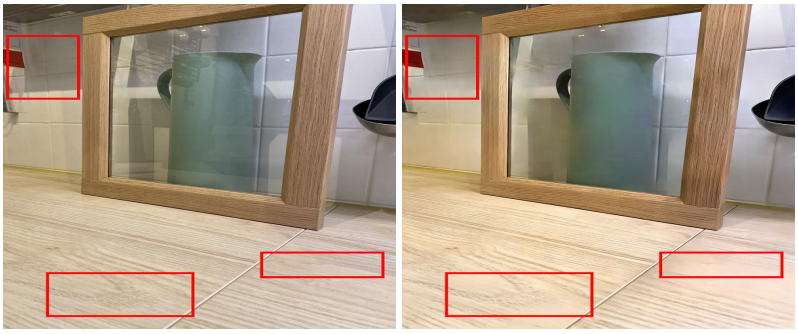
Translucent Powders: Fig.1.7 shows an challenging case when four different powder samples are deposited in a kitchen. The powders are translucent, making it easy for them to be blended with the background, especially when their color is similar to the background. Besides, the thin powder on background color smoothly changes with the powder thickness, making it hard to tell the powder-background boundary.

These cases demonstrate the importance of incorporating material awareness into computer vision algorithms. Fortunately, the material awareness is usually easy to obtain. When there is only a single type of material, the priors can be hard coded for this specific one. When multiple materials appear in a single scene, they can be coarsely identified by semantic segmentation methods [36, 141]. However, the key challenge is that such supervision signals are usually qualitative and coarse. The methods utilizing such priors should properly handle inaccuracies and uncertainties, usually via multiple priors and regularizations. Another challenge is that the material information might be hard to be represented in a mathematical or programmable form. Translating to qualitative descriptions to quantitative equations requires effort. These are problems to be solved for different tasks.

In the next section, we review related works incorporating supervision signals from various

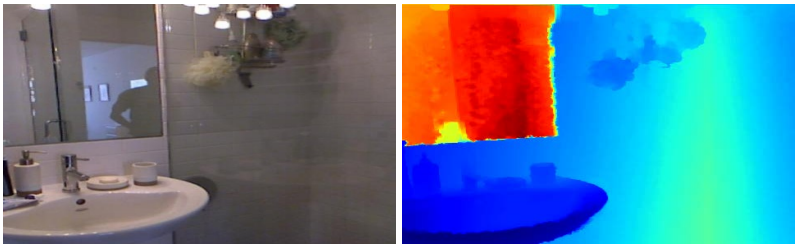


(a) Depth Estimation [128]



(b) Specular Reflection Removal [230]

Figure 1.4: Challenging cases with glass. (a) The depth estimation method Mega Depth [128] estimates the depth of the transmitted scene rather than the glass position. (b) The reflection removal method BDN [230] incorrectly removes texture or shading outside the glass region. These two examples are given by Mei *et al*. [149].



(a) Image with Mirror (b) Depth from Kinect

Figure 1.5: Challenging case with mirror. The image is from NYUDv2 dataset [190]. Kinect incorrectely captures the depth of the reflected scene rather than the mirror itself. This example is given by Yang *et al*. [232].

(a) RGB (b) Estimated Depth (c) 3D Visualization

Figure 1.6: Challenging case with specular surface. The depth estimation method (Neural RGB→D [136]) fails on the specular laptop screen.

Figure 1.7: Challenging case with translucent powders. The translucent powders blend into the backgrounds, making it hard to be detected.

sources, which provides inspiration of how to handle these challenges.

1.3 Common Sources of Supervision Signals

Before introducing the supervision signals from material awareness, we first review the common sources of supervision signals (geometry constraints, appearance models, statistical priors) for training deep networks when little ground truth real data is available.

Geometry Constraints: Geometry constraints usually establish the correspondence between pixels in different images (2D-2D correspondence) or between a 3D point and a pixel (3D-2D correspondence).

One typical application using 2D-2D correspondence is self-supervised depth estimation. With the constraints provided by epipolar geometry [57, 62, 246] or camera poses [248], an image from one view can be transformed into another view via warping. The supervision signal is obtained by comparing the warped image and the observed image. This method usually assume the world is diffuse, such that the intensity of the same point does not change across views. This method is also extended to the estimation of optical flow [138, 139], using the correspondences provided by flow vectors. Especially, the 2D-2D correspondence could also exist in the same image. For example, when an object is symmetric, the correspondence can be established with the reflection rule [223].

2D-3D correspondence is usually given by the camera matrix. By projecting a 3D object to a 2D image, the information on the 2D image can be used to supervise the prediction of the 3D point properties. This method is applied to the prediction of 3D shapes [97, 140, 152], poses [110, 170], and textures [189]. Specifically, the methods for predicting 3D shapes often use differentiable rendering [98], especially differentiable rasterization.

Appearance Models: Appearance models or rendering equations build the mathematical relationship between scene properties and image intensities. The model is usually used for generating synthetic data for supervised learning, or for providing photometric loss in self-supervised learning.

Synthetic data generation has been widely applied for predicting geometry properties (shape [10, 196], depth [12, 203], flow [148], layout [243]), appearance properties (material [126, 130], lighting [130]), and semantic properties (recognition [14], detection [203], segmentation [68, 203]). Since the images can be rendered off-the-shelf, the synthesis procedure does not need to be differentiable and thus traditional computer graphics can be utilized.

Self-supervised learning requires the loss function to be differentiable. Although accurate physics-based models can be implemented with differentiable ray tracing [13], the overhead is high, making the training stage time consuming. Thus, simple appearance models are usually preferred. Typical examples include dichromatic reflectance model for diffuse-specular separation [18], alpha blending model for reflection removal [107], Lambertian reflectance model for intrinsic image decomposition [86], and Blinn-Phong model for material estimation [150].

Statistical Priors: Statistical priors are derived from data. They might be task-specific, because different scenarios may include different data distribution. Below we review several common statistical priors.

Shape models are usually parametric models learned from data. Models usually are built for

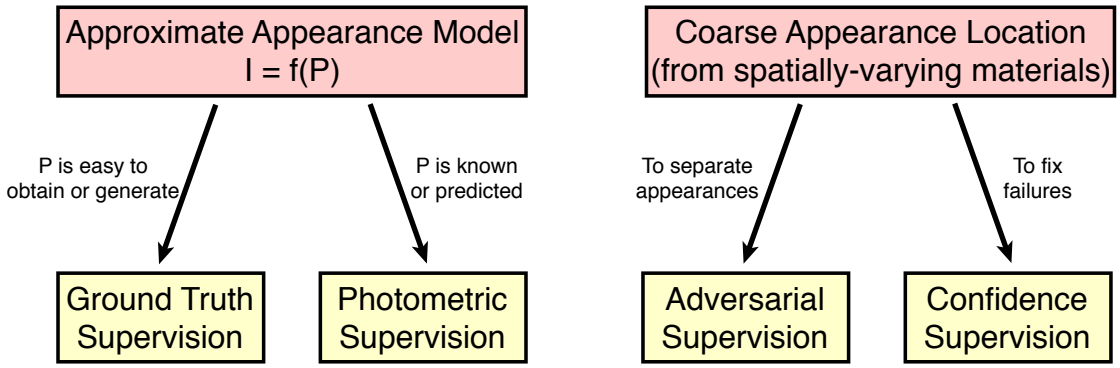


Figure 1.8: Proposed framework for exploiting material-aware supervisions. Given a task, an approximate appearance model could be build to describe the whole or part of the scene. Depending on the availability of scene properties, ground truth supervision and/or photometric supervision could be applied. If additional appearance location information is provided by spatially-varying materials, adversarial supervision and/or confidence supervision could be used for separating appearances or fixing failures.

human bodies [142], faces [48], animals [251], vehicles [214] and common objects [229]. The shape deforms according to the parameters, but within a limited space. By parameterizing the geometry with shape models, the solution space is constrained, providing a strong regularization to the algorithm.

Spatial smoothness usually works as regularization terms providing weak supervision signals. Typical applications include intrinsic image decomposition [127], depth estimation [62] and mesh reconstruction [97].

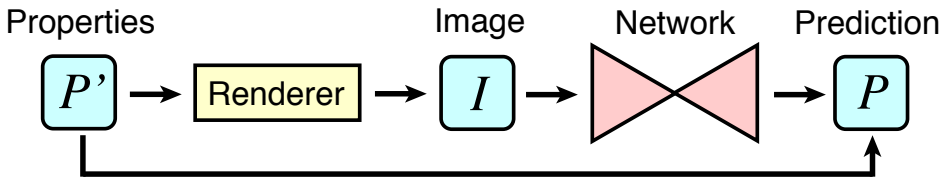
Temporal smoothness includes the smoothness of motion and the smoothness of intensity. Motion smoothness (often in the form of constant velocity) [222] is usually applied on geometry predictions in a video to enhance temporal consistency. The smoothness of intensity can be used for appearance estimation [127], by encouraging the predictions for adjacent frames to be similar.

The architecture of deep convolutional networks (CNN) implicitly includes the natural image prior, called deep image prior. The output of a randomly initialized CNN tends to converge to a natural image during optimization. This prior can be used for unsupervised image restoration [206] and decomposition [54].

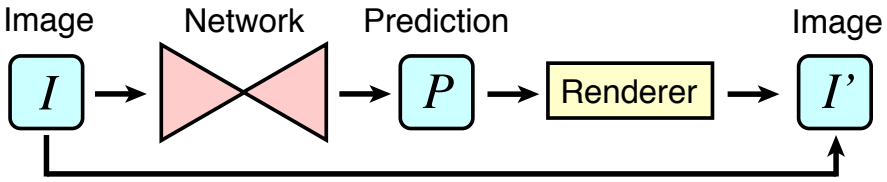
The material-aware supervision signals we study in this thesis partially overlap with the aforementioned priors but not completely the same. They can present as geometry constraints, appearance models, or statistical priors, but they come from material awareness. In the next section, we describe the proposed framework for incorporating such supervision signals into real tasks.

1.4 Contributions

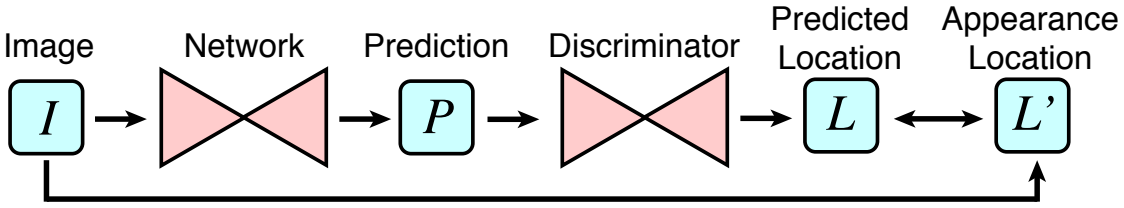
We propose a framework to exploit material awareness for supervising the training of deep networks. Given a task, the framework can be used as a protocol to choose the suitable ways



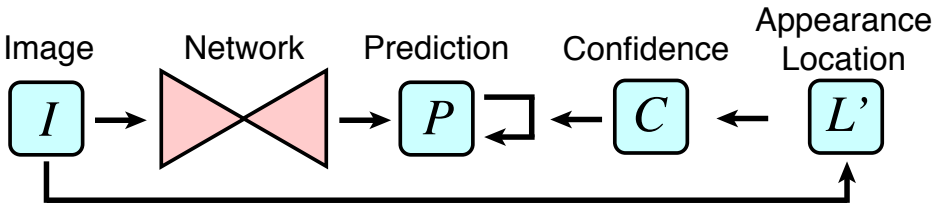
(a) Ground Truth Supervision



(b) Photometric Supervision



(c) Adversarial Supervision



(d) Confidence Supervision

Figure 1.9: Illustrations of four different forms of supervision signals. (a)(b) are provided by appearance models while (c)(d) come from appearance locations. The strength of the supervision from strong to weak is: (a) ground truth supervision, (b) photometric supervision; (c) adversarial supervision, (d) confidence supervision.

of incorporating material-aware supervisions. We applied this framework to different tasks to demonstrate its effectiveness on a wide range of applications. In this section, we first introduce the framework and then briefly summarize the four applications that will be presented in the following chapters.

1.4.1 Proposed Framework

The proposed framework for exploiting material-aware supervisions is shown in Fig.1.8. We consider two main sources of supervision signals: appearance model and appearance location. The appearance model is usually a mathematical equation describing the reflectance of a specific type of material. Typical examples include: N-dot-L shading for diffuse materials, dichromatic reflectance model for specular materials, and foreground-background blending model for translucent materials. The model is usually approximate and simple. The appearance location information comes from the awareness of spatially-varying materials. For example, the location of strong specular reflections can be infered from the position of emissive materials. The awareness of non-Lambertian materials tells us where an algorithm assuming constant intensity across views may fail.

Based on the two sources, we consider four different forms of supervision signals: ground truth and photometric supervisions from appearance models, and adversarial and confidence supervisions from appearance locations. Specifically, given a task, an approximate appearance model based on material awareness could be built to describe the whole or part of the scene. Let $I = f(P)$ be the model, where I denotes the image, P denotes scene properties, and f is the appearance function. When P is easy to obtain or generate, we could render synthetic images for ground truth supervision. When P is either known or predicted by the network, we can optimize the networks using photometric supervision, assuming f is differentiable. The scene may also contain spatially-varying materials providing additional appearance location information. If the goal is to separate such appearances, the adversarial supervision can be applied. A discriminator can be trained to recover the appearance location while the prediction network prevents it from doing so. In this way the appearance can be removed from the original image. The adversarial supervision is usually good for cases where the appearance location information is a coarse version of the prediction target. If the goal is to fix failures caused by such appearances, the confidence supervision could be used by assigning a lower confidence to regions with special appearances. The confidence supervision usually requires other supervisions as data terms. Given a task, we may check the conditions of each supervision signal and choose the appropriate forms to incorporate material awareness for training deep networks. In Chapter 6, we discuss this in detail to show how to select the suitable supervision signals for different tasks.

The four forms of supervision signals have different strengths. As shown in Fig. 1.9, the ground truth supervision is the strongest one because it directly provides the "correct" answer to the task. The photometric supervision is weaker than the ground truth because the signal is indirect. The inaccuracy of the photometric equation may lead to noisy and non-robust predictions. To deal with this issue, other regularization terms are usually required. The adversarial supervision is usually weaker because unlike ground truth and photometric supervisions which supervise the output of the network, it supervises the discriminator network. The output of the discriminator is then used to supervise the prediction network. The weakest signal is confidence

	Powder Recognition	Human Reconstruction	Floor Decomposition	Road Scene Stereo Matching
Input/Knowledge	Multispectral image	(a) RGB-D video (b) Human model (c) Lighting	(a) Panoramas (b) Layout (c) Semantics	(a) RGB-NIR stereo (b) Semantics
Target Property				
Geometry		Mesh		Depth
Appearance		Texture	Diffuse/specular ambient/sunlight	
Semantics	Powder classes			
Supervision Signal				
Appearance Model	GT	GT & Photometric	Photometric	Photometric
Appearance Location			Adversarial	Confidence
Material Type				
Emissive			✓	✓
Diffuse		✓	✓	✓
Specular/Glossy			✓	✓
Transparent			✓	✓
Translucent	✓			

Table 1.1: Comparison of tasks presented in this thesis based on inputs or prior knowledge, target properties, supervision signals, and material types.

supervision, because it only provides information about whether one predicted value is more confident than the other one rather than the value itself. It has to be used together with another supervision signal providing information about the values themselves.

1.4.2 Applications

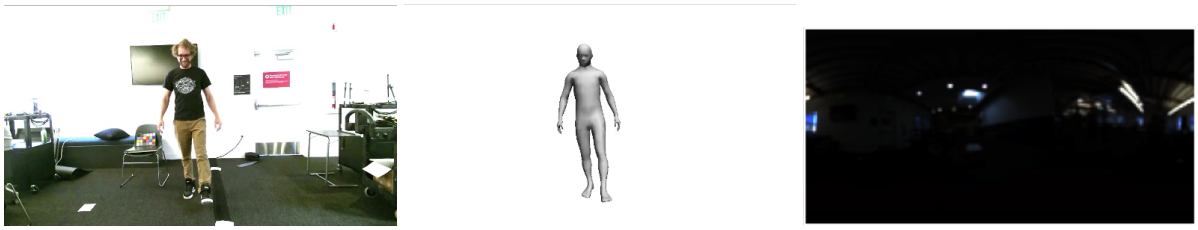
This thesis studies four tasks to demonstrate the effectiveness of the proposed framework. As listed in Tab. 1.1, the four tasks (powder recognition, human reconstruction, floor decomposition, and road scene stereo matching) are selected to cover different input/output modalities and material types. They also cover different branches and supervision signals in the proposed framework.

Powder Recognition: In Chapter 2, we introduce an approach for fine-grained recognition of powders on complex backgrounds, to provide an example of synthetic ground truth supervision from translucent material awareness. As shown in Fig. 1.10, the input is a multispectal image with bands ranging from visible light (RGB) to near infrared (NIR) and short wave infrared (SWIR). The output is an 101-class (100 powders + 1 background) per-pixel recognition result.

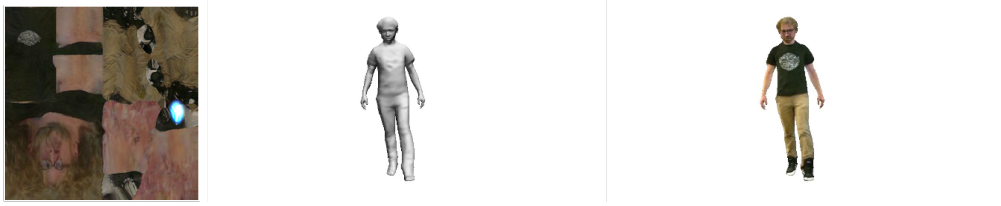
Because powders are translucent, the thin powders on background appearance depends on the thickness of the powder. To obtain supervision signals, we build an appearance model for synthesizing images of translucent powders on various backgrounds. This model takes powder color, background color, and thickness as variables, considering the translucent prior. The model parameters are also easy to calibrate. With the ground truth supervision provided by the synthetic

(a) Multispectral Image (b) Output recognition result (c) Powders w/ Different Thicknesses

Figure 1.10: Powder Recognition. The powder recognition task takes RGB, NIR and several SWIR spectral bands as input, and detects and recognizes powder samples. The key challenge is the translucency of powders. In (c), each column is one type of powder. Depending on the thickness, the thin powders show different colors.

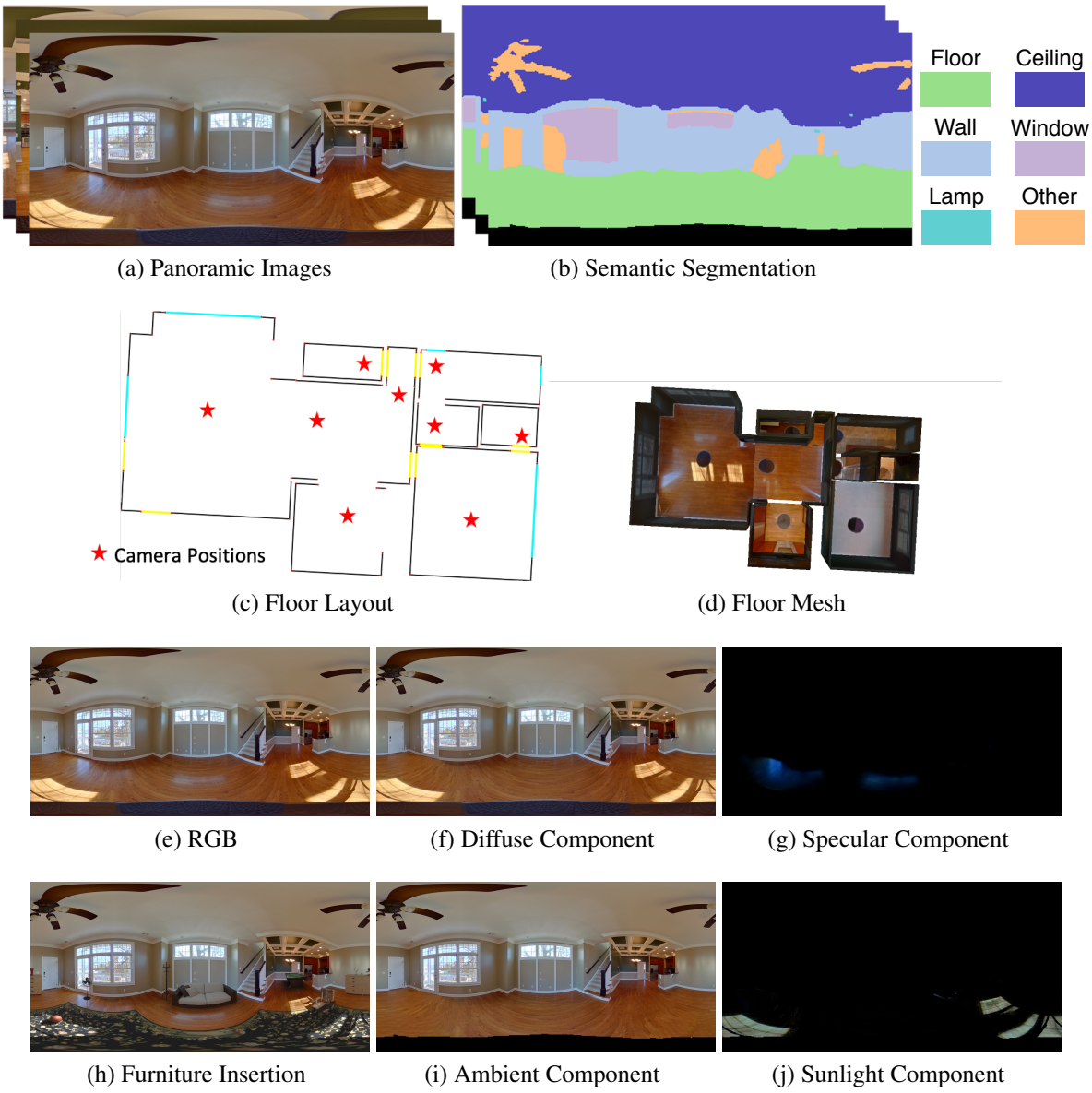


(a) RGB Video (b) Coarse Parametric Mesh (c) Captured Lighting



(e) Output Texture (d) Output Fine Mesh (f) Rendering Result

Figure 1.11: Human Reconstruction. The human reconstruction task takes an RGB video (a), a coarse parametric mesh from RGB-D tracking (b), and a captured environment map (c) as input. It reconstructs a per-frame fine mesh (d), together with a constant high resolution full body texture (e). We can render (f) using the reconstructed texture and mesh.



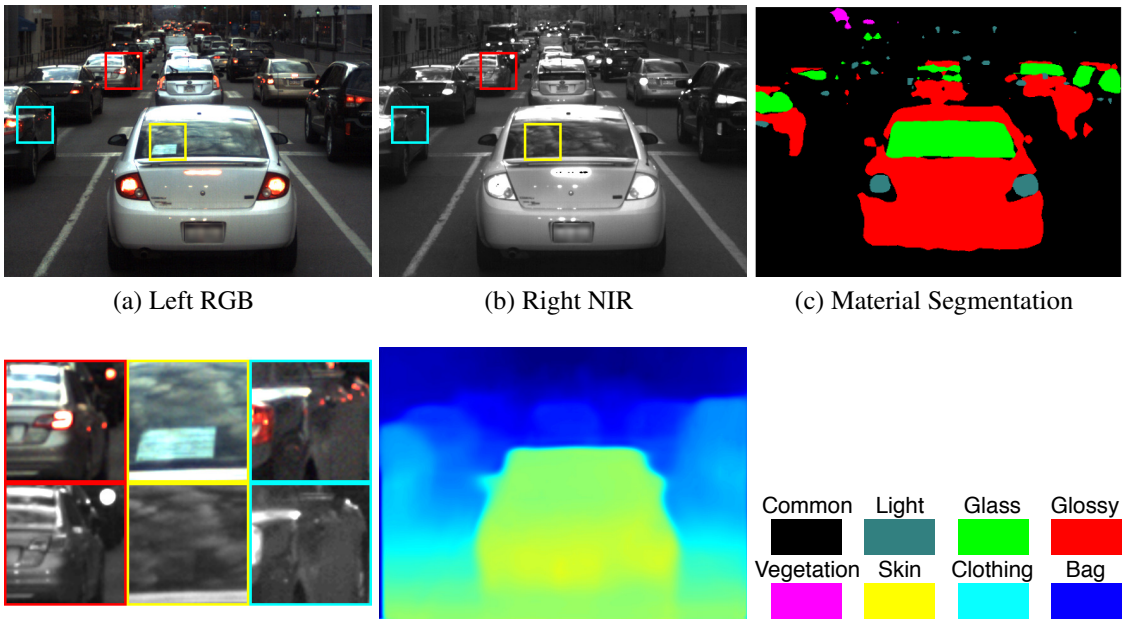
(a) Panoramic Images (b) Semantic Segmentation

(c) Floor Layout (d) Floor Mesh

(e) RGB (f) Diffuse Component (g) Specular Component

(h) Furniture Insertion (i) Ambient Component (j) Sunlight Component

Figure 1.12: Floor Decomposition. (a)(b)(c) are the inputs including panoramic images, semantic segmentation and floor layout with camera poses. (d) is a 3D visualization of the floor mesh. Our method decomposes the floor region of an RGB panorama (e) into diffuse (f) and specular (g) components, and further decompose the diffuse component into ambient (i) and sunlight (j) component. This decomposition enables high quality furniture insertion result (h), including soft and hard shadows, occluded specular reflections, and sunlight cast on the objects.



(a) Left RGB (b) Right NIR (c) Material Segmentation

(d) Unreliable Regions (e) Predicted Disparity

Figure 1.13: Road Scene Stereo Matching. The inputs consist of an RGB-NIR stereo pair (a) & (b), while the target is to predict the disparity map (c). To fix the failures on non-Lambertian regions, we use semantic segmentation to locate and recognize those materials. By assigning a lower confidence to regions with unreliable matching, we correctly recover the disparity map.

data, we are able to achieve reasonable recognition performance.

One limitation of this work is that the accuracy may not be sufficient for a safety application demanding near perfect detection of dangerous powders. It may be improved by adding more data or considering a wider spectral range. Besides, the proposed blending model cannot model metallic and glowing powders. Special models should be used to describe their appearances.

Human Reconstruction: In Chapter 3, we demonstrate a method for recovering human texture and geometry from an RGB-D video, as an example of photometric supervision from Lambertian material assumption. As shown in Fig. 1.11, given the RGB frames, the captured environment map, and the coarse per-frame human mesh from RGB-D tracking, our method reconstructs spatiotemporally consistent and detailed per-frame meshes along with a high-resolution albedo texture.

To obtain supervision signals, we first train deep models on synthetic data and then adapt them to real data via self-supervised learning. Both synthetic data generation and self-supervised learning rely on the Lambertian appearance model. Additional spatio-temporal priors are incorporated to improve the robustness of the algorithm. The recovered mesh and texture enables state of art free-viewpoint rendering of humans on challenging real videos.

One limitation of this work is that it cannot handle topology changes, because the clothing deformation is modeled as vertex deformation. Modeling deformation using implicit functions [174] or separate meshes [236] could possibly resolve this issue. Another limitation is that we rely on captured lighting. Lighting estimation could be incorporated in the future.

Floor Decomposition: In Chapter 4, we propose a floor appearance decomposition approach for realistic object insertion, as an example of adversarial supervision for diffuse-specular separation and direct sunlight detection. As shown in Fig. 1.12, the inputs to the system consist of panoramic images, floor plan with camera poses, and material semantics from semantic segmentation. Our target is to separate the floor appearance into diffuse and specular components, and further decompose the diffuse component into ambient and sunlight components.

The key supervision signal comes from appearance locations. The coarse locations of specular and sunlight appearances are obtained from layout geometry and material semantics. Using photometric supervision together with adversarial supervision driven by the coarse locations, we are able to separate diffuse/specular and ambient/sunlight components, enabling photorealistic virtual furniture insertion into empty rooms.

One limitation of this work is that the confusion between specular reflection and direct sunlight cannot be resolved when they overlap with each other. Multi-task learning could be a potential option to solve the problem. Besides, we currently process only the floor. Applying similar techniques for the wall and ceiling is a direction we intend to pursue.

Road Scene Stereo Matching: In Chapter 5, we present a cross-spectral stereo matching method for road scenes, to show that the confidence supervision from non-Lambertian appearance locations helps fix regions of failure. The input to the system is an RGB-NIR stereo pair, and the output is a disparity map.

Since no depth ground truth is available for training, we obtain photometric supervision via a warping-based image synthesis method which works for Lambertian materials only. We locate unreliable non-Lambertian regions provided by a material recognition module and fix them using confidence-based supervision.

One limitation of this method is that the spectral difference problem cannot be completely handled via spectral translation. This problem could potentially be solved by converting RGB and NIR into an intermediate representation. In addition, the results are generally blurry at object boundaries, due to the usage of smoothness loss. This problem could potentially be solved by explicitly consider occlusion in the disparity maps [216].

Chapter 2

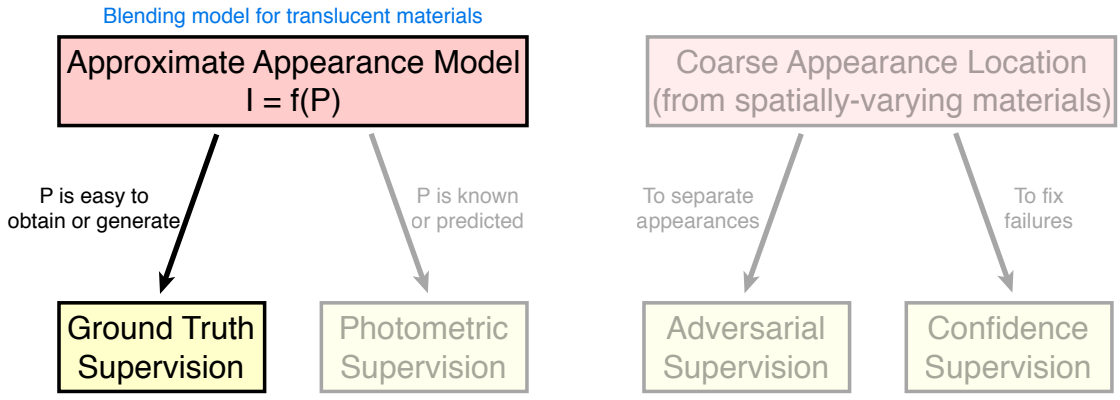
Ground Truth Supervision from Appearance Model

In the previous chapter, we have introduced the idea of using material awareness to supervise the training of deep networks. We proposed a framework with four forms of supervision signals (ground truth, photometric, adversarial, and confidence) derived from two sources (appearance model, appearance location). In this chapter, we introduce ground truth supervision based on synthetic data. Ground truth supervision is a good choice when the scene properties as the input to the renderer have to be easy to obtain or generate.

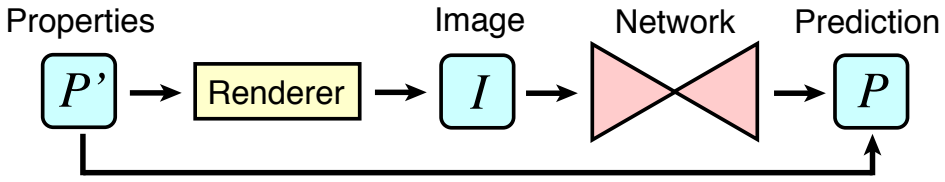
As shown in Fig. 2.1, we study powder recognition, as an example of using synthetic ground truth supervision from a translucent appearance model for training deep networks. Specifically, we study the fine-grained recognition of a type of translucent materials—powders. Because visible light does not provide enough information for this task, we look at multispectral images. We build an appearance model for rendering powder on background images. Given the translucent material prior, this model takes powder color, background color, and thickness as variables into consideration. By supervised training on data synthesized based on this model and a very small amount of real data, the deep network achieves a reasonable accuracy.

2.1 Application: Multispectral Imaging for Powder Recognition

Hundreds of materials such as drugs, explosives, makeup, food or other chemicals are in the form of powder. It is important to detect and recognize such powders for security checks, drug control, criminal identification, and quality assessment. However, visual powder recognition is challenging for many reasons. Powders have deceptively simple appearances — they are amorphous and matte with little texture. Fig. 2.2 shows 20 powders that exhibit little color or texture variation in the Visible (RGB, 400-700nm) or Near-Infrared (NIR, 700-1000nm) spectra but are very different chemically (food ingredients to poisonous cleaning supplies). Unlike materials like grass and asphalt, powders can be present anywhere (smudges on keyboards, kitchens, bathrooms, outdoors, etc.) and hence the scene context is of little use for accurate recognition. To make matters worse, powders can be deposited on other surfaces with various thicknesses (and

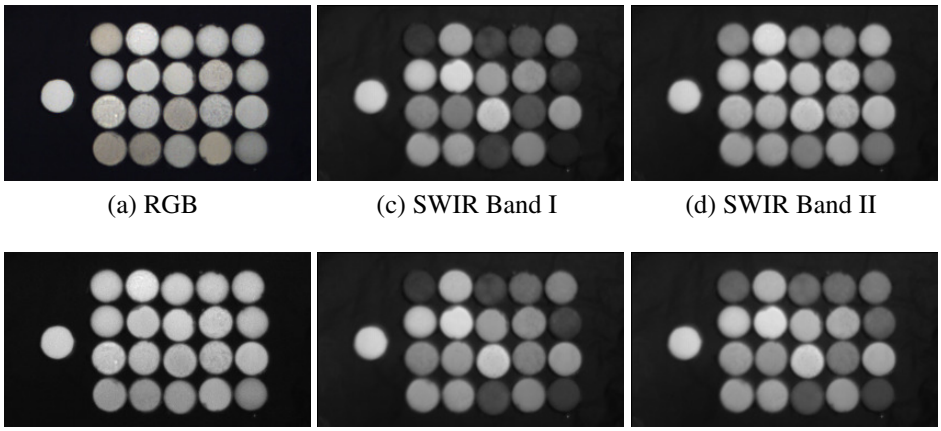


(a) Framework



(b) Ground Truth Supervision

Figure 2.1: The powder recognition task uses appearance model information from material awareness for ground truth supervision. A blending model for translucent materials is built for modeling thin powder appearance. This model is used to generate synthetic data for training deep networks with ground truth supervision.



(a) RGB (c) SWIR Band I (d) SWIR Band II

(b) NIR (e) SWIR Band III (f) SWIR Band IV

Figure 2.2: White powders that are not distinguishable in visible light (RGB) and Near Infrared (NIR) show significantly different appearances in Shortwave Infrared (SWIR). The leftmost sample is a white patch for white balance while the others are powders. Row 1 (left to right): Cream of Rice, Baking Soda, Borax Detergent, Ajinomoto, Aspirin; Row 2: Iodized Salt, Talcum, Stevia, Sodium Alginate, Cane Sugar; Row 3: Corn Starch, Cream of Tartar, Blackboard Chalk, Boric Acid, Smelly Foot Powder; Row 4: Fungicide, Calcium Carbonate, Vitamin C, Meringue, Citric Acid.

hence, translucencies), ranging from a smudge to a heap. Capturing such data does not only cost time but also consumes powders and degrades surfaces.

We present a comprehensive dataset and approach for powder recognition using multispectral imaging. We have found that a broad range of spectral wavelengths (from visible RGB to Short-Wave Infrared: 400-1700nm) can discriminate powders with reasonable accuracy. For example, Fig. 2.2 shows that SWIR (1000-1700nm) can discriminate powders with little color information in RGB or NIR spectra.

The data collection for powder recognition is hard because of the aforementioned variations in the thicknesses and the surfaces on which powders could be deposited. To overcome this challenge, we present a blending model to faithfully render powders of various thicknesses (and translucencies) against known background materials. The model assumes that thin powder appearance is a per-channel alpha blending between thick powder (no background is visible) and background, where α follows the Beer-Lambert law. This model can be deduced from the more accurate Kubelka-Munk model [112] via approximation, but with parameters that are practical to calibrate. The data rendered using this model is crucial to achieve strong recognition performance on real data.

Our multi-spectral dataset for powder recognition is captured using a co-located RGB-NIR-SWIR imaging system. While the RGB and NIR cameras (RGBN) are used as-is, the spectral response of the SWIR camera is controlled by two voltages. The dataset has two parts: *Patches* contains images of powders and common materials and *Scenes* contains images of real scenes with or without powder. For *Patches*, we imaged 100 thin and thick powders (food, colorants, skincare, dust, cleaning supplies, etc.) and 100 common materials (plastics, fabrics, wood, metal,

paper, etc.) under different light sources. *Scenes* includes 256 cluttered backgrounds with or without powders on them. We incorporate data synthesis into deep learning to perform an 101-class semantic segmentation (including background class) when the powder location is unknown, achieving mean IoU of over 40%.

2.2 Related Work

Near-Infrared Spectroscopy: Unlike sensor manufacturers, physicists treat electromagnetic spectrum from 780nm to 2500nm as near-infrared (NIR), which covers the short-wave infrared (SWIR) range defined in our work. NIR spectroscopy [162] relies on overtone and combination vibrations of molecules to analyze information for astronomy [221], agriculture [27], and medication [28], etc.

Powder Detection and Recognition: Terahertz imaging is used for the detection of powders [215], drugs [100, 101] and explosives [184]. Nelson *et al*. [156] uses SWIR hyperspectral imaging to detect threat materials and to decide whether a powder is edible. However, none of them studied on a large dataset with powders on various backgrounds.

Hyperspectral Band Selection: Band selection [30, 31, 59, 88, 146, 164, 213] is a common technique in remote sensing. MVPCA [31] maximizes variances, which is subject to noise. Since noisy bands can have large variances. ID [30] ranks bands with information divergence, without concerning the correlation between bands, leading to high redundancy. AP[53] selects bands by exemplar-based clustering and message passing. But the number of clusters can not be specified, resulting in trouble when the user wants to select a specific number of bands. A rough set based method [164] assumes two samples can be separated by a set of bands only if they can be separated by one of the bands, which ignores the cross-band information. Moreover, there are usually only a few classes in remote sensing, while there are hundred classes in our case.

Blending Model: Unlike spectral mixture [102] where materials are blended due to low resolution or actual material mixture, the powder against background appearance is the blending of two-layer materials. Alpha Blending [168] is a linear model assuming all channels share the same transparency, which is not true for real powders. Physics based models [22, 71, 91, 112, 151, 191] usually include parameters hard to calibrate. The Kubelka-Munk model [112] models scattering media on background via a two-flux approach. However, it models absolute reflectances rather than intensities, requiring precise instruments for calibration and costing time.

2.3 RGBN-SWIR Powder Recognition Database

We build the first comprehensive RGBN-SWIR Multispectral Database for powder recognition. We first introduce the acquisition system in Section 2.3.1. In Section 2.3.2, we describe the dataset—*Patches* providing resources for image based rendering, and *Scenes* providing cluttered backgrounds with or without powder. To reduce the acquisition time, we present a band selection method in Section 2.3.3, and use selected bands to extend the dataset.

Figure 2.3: Image Acquisition System. RGB, NIR, and SWIR cameras are co-located using beamsplitters. The target is imaged through a $45°$ mirror.

2.3.1 Image Acquisition System

The SWIR camera is a ChemImage DP-CF model [156], with a liquid crystal tunable filter set installed. The spectral transmittance (1000-1700nm) of the filter set is controlled by two voltages ($1.5\text{V} \leq V_0, V_1 \leq 4.5\text{V}$). We call each spectral setting a band or a channel, corresponding to a broad band spectrum (Fig. 2.8). It takes 12min to scan the voltage space at 0.1V step to obtain a 961-band image. The 961 values of a pixel (or mean patch values) can be visualized as a 31×31 SWIR signature image on the 2D voltage space.

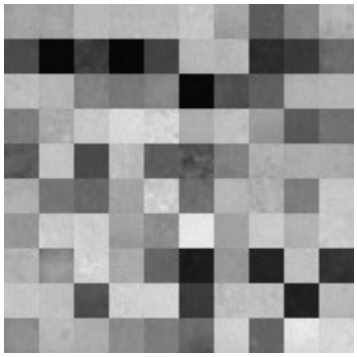
We co-locate the three cameras (RGB, NIR, SWIR) using beamsplitters (Fig. 2.3), and register images via homography transformation. The setup is bulky to mount vertically, hence a target on a flat surface is imaged through a $45°$ mirror. A single light source is placed towards the mirror. We use 4 different light sources for training or validation (**Set A**), and 2 others for testing (**Set B**).

2.3.2 Patches and Scenes

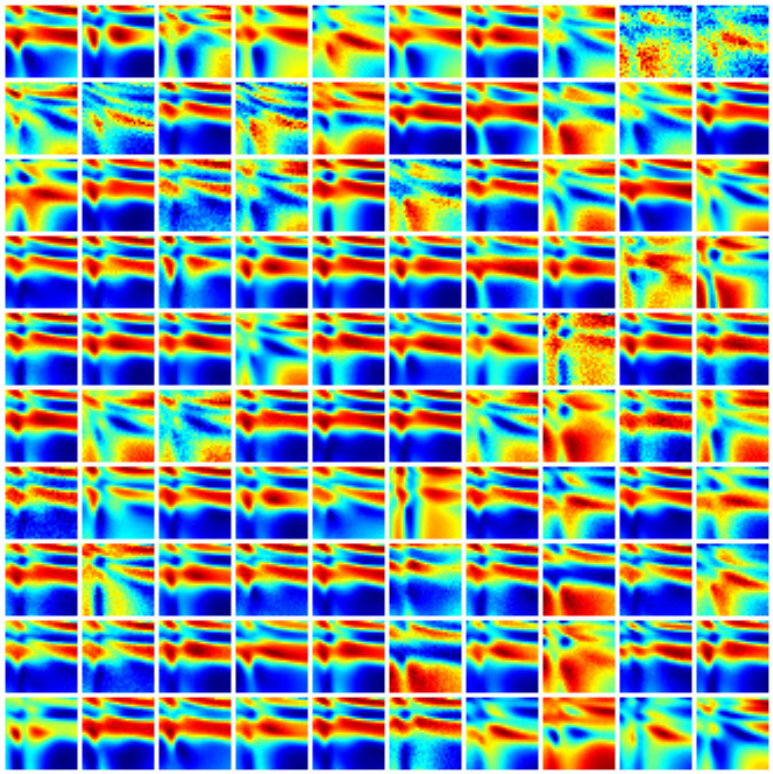
The dataset includes two parts: **Patches** provides patches (size 14×14) to use for image based rendering; **Scenes** provides scenes (size 280×160) with or without powder. White balance is done with a white patch in each scene.

Patches (Tab. 2.2) includes 100 powders and 100 common materials that will be used to synthesize appearance on complex backgrounds. Powders are chosen from multiple common groups - food, colorants, skincare, dust, cleaning supplies, etc. Examples include Potato Starch (food), Cyan Toner (colorant), BB Powder (skincare), Beach Sand (dust), Tide Detergent (cleansing), and Urea (other). See Tab. 2.1 for the full list and Fig. 2.5 for powder samples. The RGBN images and SWIR signatures of the 100 powder patches are shown in Fig. 2.4. Common materials (surfaces) on which the powders can be deposited include plastic, fabrics, wood, paper,

(a) Thick RGB Patch



(b) Thick NIR Patch



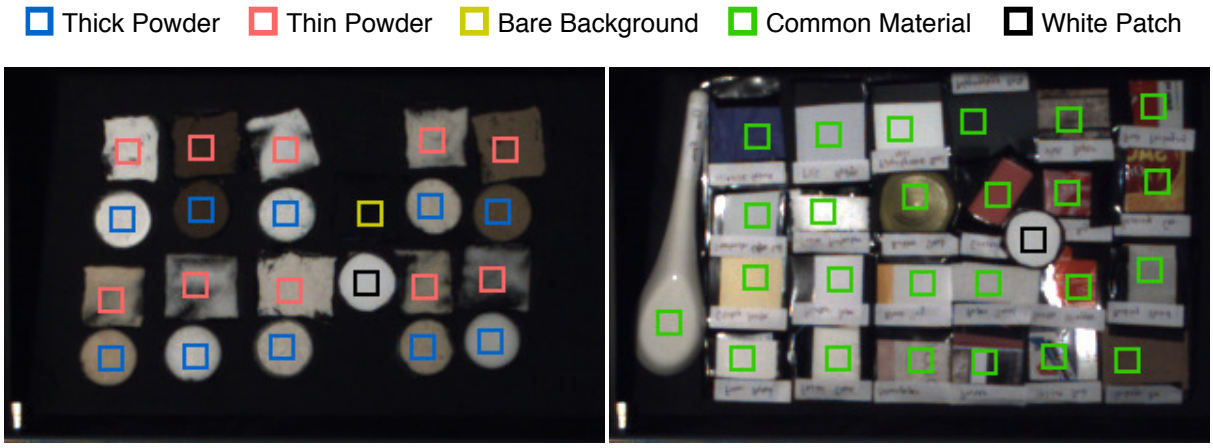
(c) Thick SWIR Signature

Figure 2.4: Hundred powders. Thick RGB patches, NIR patches and normalized SWIR signatures are shown.

Figure 2.5: Thick and thin powder samples

	Name	Category	Legend
1	Ajinomoto	Food	
2	Almond Flour	Food	
3	Aqua Glow	Colorant	
4	Aspirin	Other	
5	Baby Powder	Skincare	
6	Baking Soda	Food	
7	Barley Water	Food	
8	BB Powder	Skincare	
9	Beach Sand	Dust	
10	Blackboard Chalk	Dust	
11	Black Frit	Dust	
12	Black Iron Oxide	Colorant	
13	Black Pepper	Food	
14	Black Toner	Colorant	
15	Blue Pigment	Colorant	
16	Borax Detergent Booster	Cleansing	
17	Boric Acid	Cleansing	
18	Bronze Metallic	Colorant	
19	Brown Dye	Colorant	
20	Brown Sugar	Food	
21	Calcium Carbonate	Dust	
22	Cane Sugar	Food	
23	Caralluma	Food	
24	CC Powder	Skincare	
25	Celtic Sea Salt	Food	
26	Charcoal	Colorant	
27	Chaste Tree Berry	Food	
28	Chicken Bath	Dust	
29	Citric Acid	Food	
30	Cobalt Frit	Dust	
31	Cocoa	Food	
32	Coconut Flour	Food	
33	Coconut Oil	Food	
34	Coffe Mate	Food	
35	Corn Starch	Food	
36	Cream of Rice	Food	
37	Cream of Tartar	Food	
38	Cream of Wheat	Food	
39	Cyan Toner	Colorant	
40	Detox Powder	Dust	
41	Dragon Blood	Other	
42	Dry Milk	Food	
43	Espresso	Food	
44	Eye Shadow	Skincare	
45	Fake Moss	Other	
46	Flower Fuel	Other	
47	Fuchsia Dye	Colorant	
48	Fungicide	Cleansing	
49	Garlic	Food	
50	Ginger	Food	
51	Green Bean Water	Food	
52	Green Glow	Colorant	
53	Green Pigment	Colorant	
54	Guar Gum	Food	
55	Gym Chalk	Dust	
56	Hibiscus	Food	
57	Iodized Salt	Food	
58	Loose Powder	Skincare	
59	Lotus	Food	
60	Magenta Toner	Colorant	
61	Matcha	Food	
62	MCT Oil	Food	
63	Meringue	Food	
64	Milk Replacer	Food	
65	Moringa	Food	
66	Nail Dipping	Colorant	
67	Onion	Food	
68	Orange Glow	Colorant	
69	Orange Peel	Food	
70	Pearl Powder	Skincare	
71	Pet Moist	Cleansing	
72	Potassium Iodide	Other	
73	Potato Starch	Food	
74	Quick Blue Bleach	Cleansing	
75	Red Bean Water	Food	
76	Root Destroyer	Cleansing	
77	Sandalwood	Other	
78	Schorl Tourmaline	Dust	
79	Shaving Powder	Skincare	
80	Silver Metallic	Colorant	
81	Smelly Foot Powder	Cleansing	
82	Sodium Alginate	Food	
83	Spanish Paprika	Food	
84	Stain Remover	Cleansing	
85	Stevia	Food	
86	Stone Cement	Dust	
87	Sun Powder	Skincare	
88	Talcum	Skincare	
89	Teal Azul Dye	Colorant	
90	Tide Detergent	Cleansing	
91	Urea	Other	
92	Vanilla	Food	
93	Vitamin C	Food	
94	Wheat Grass	Food	
95	White Pepper	Food	
96	Yellow Dye	Colorant	
97	Yellow Glow	Colorant	
98	Yellow Pigment	Colorant	
99	Yellow Toner	Colorant	
100	Zinc Oxide	Skincare	

Table 2.1: Powder List. Powder names, categories, and legends for segmentation labels are listed.



(a) Thick/Thin Powders (b) Common Materials

Figure 2.6: **Patches** example. Thin powders are put on the same black background material. Patches are manually cropped for thick powders, thin powders, bare background, common materials, and white patch.

(a) Background Image (b) Image with Powder (c) GT Powder Mask

Figure 2.7: **Scenes** example. The ground truth mask is obtained by background subtraction and manual annotation.

metal, etc. All patches are imaged 4 times under different light sources (Set A). To study thin powder appearances, we also imaged thin powder samples on a constant background. As shown in Fig. 2.6 (a), thick powders, thin powders, and a bare background patch are captured in the same field of view.

Scenes (Tab. 2.3) includes cluttered backgrounds with or without powder. Ground truth powder masks are obtained via background subtraction and manual editing (Fig. 2.7). Each powder in *Patches* appears 12 times in *Scenes*. In Tab. 2.3, scenes captured with light sources Set A are for training or validation, while the others are for testing. *Scene-bg* only has background images, while the others have both backgrounds and images with powder. *Scene-sl-train* and *Scene-sl-test* are larger datasets of scenes with powder that include only selected bands (explained in Section 2.3.3).

Dataset ID	Target	Light Sources	Num Patches
Patch-thick	100 thick powders	Set A	400
Patch-thin	100 thin powders	Set A	400
Patch-common	100 common materials	Set A	400

Table 2.2: **Patches**. 100 thick and thin powders, and 100 common materials are imaged under light sources Set A.

Dataset ID	Light Sources	Num SWIR Bands	Num Scenes	N Powder Instances
Scene-bg	Set A	961	64	0
Scene-val	Set A	961	32	200
Scene-test	Set B	961	32	200
Scene-sl-train	Set A	34	64	400
Scene-sl-test	Set B	34	64	400

Table 2.3: **Scenes**. Each powder appears 12 times. *Scene-sl-train* and *Scene-sl-test* include bands selected by NNCV, Grid Sampling, MVPCA [31], and Rough Set [164].

2.3.3 Nearest Neighbor Based Band Selection

Capturing all 961 bands costs 12min, forcing us to select a few bands for capturing a larger variation of powders/backgrounds. Band selection can be formulated as selecting a subset B_s from all bands B_a, optimizing a pre-defined score. We present a greedy method optimizing a Nearest Neighbor Cross Validation (NNCV) score. Let N_s be the number of bands to be selected. Starting from $B_s = \emptyset$, we apply the same selection procedure N_s times. In each iteration, we compute the NNCV score of $B_s \cup b$ for each band $b \notin B_s$. The band b maximizing the score is selected and added to B_s. Pseudocode is in Algorithm 1.

To calculate the NNCV score, we compute the mean value of each patch in *Patch-thick* and *Patch-common* (Tab. 2.2) to build a dataset with 101 classes (background and 100 powders), and perform leave-one-out cross validation. Specifically, for each data point x in the database, we find its nearest neighbor $NN(x)$ in the database with x removed, and treat the class label of $NN(x)$ as the prediction of x. The score is the mean class accuracy.

The distance in nearest neighbor search is calculated on RGBN bands and SWIR bands in $B_s \cup b$. Because the number of SWIR bands changes during selection, after selecting 2 bands, we propose to compute cosine distances for RGBN and SWIR bands separately and use the mean value as the final distance. We call this the **Split Cosine Distance**.

We extend the *Scenes* dataset by capturing only the selected bands. *Scene-sl-train* and *Scene-sl-test* in Tab. 2.3 include 34 bands selected by 4 methods (9 bands per method, dropping duplicates): (1) NNCV (ours) as described above, (2) Grid Sampling uniformly samples the 2D voltage space, (3) MVPCA [31] maximizes band variances, and (4) Rough Set [164] optimizes

Algorithm 1 NNCV Band Selection

Input: Number of SWIR bands to be selected N_s

Output: Selected SWIR bands B_s

$B_s \leftarrow \emptyset$

$B_n \leftarrow$ all SWIR bands

for $i = 1 : N_s$ **do**

  for each $b \in B_n$ **do**

    $score_b \leftarrow$ mean class accuracy of nearest neighbor cross validation using RGBN and $B_s \cup \{b\}$ bands

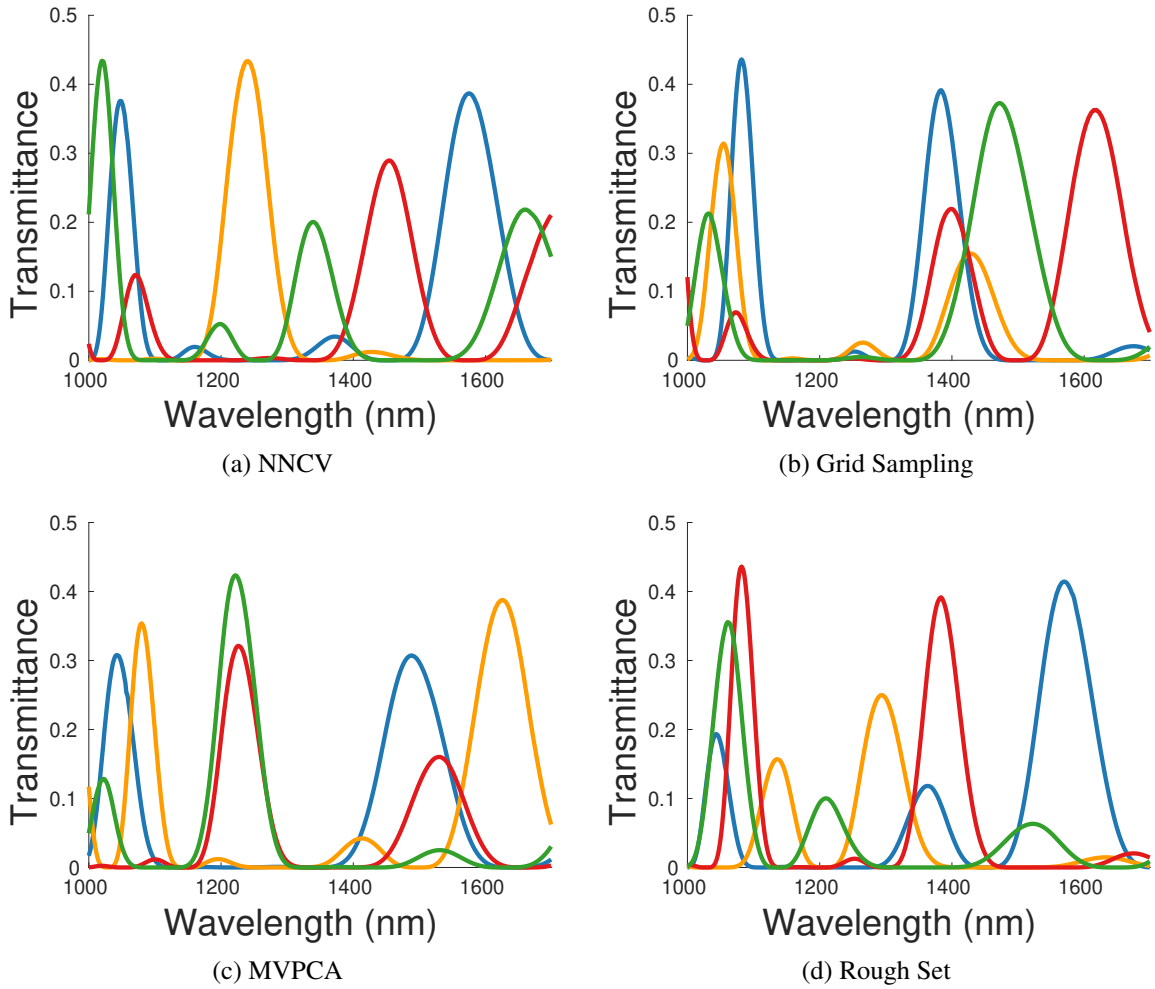
  end for

  $b \leftarrow \underset{b \in B_n}{\operatorname{argmax}}\, score_b$

  $B_s \leftarrow B_s \cup \{b\}$

  $B_n \leftarrow B_n - \{b\}$

end for



(a) NNCV

(b) Grid Sampling

(c) MVPCA

(d) Rough Set

Figure 2.8: Theoretical spectral transmittance of 4 selected bands (different colors).

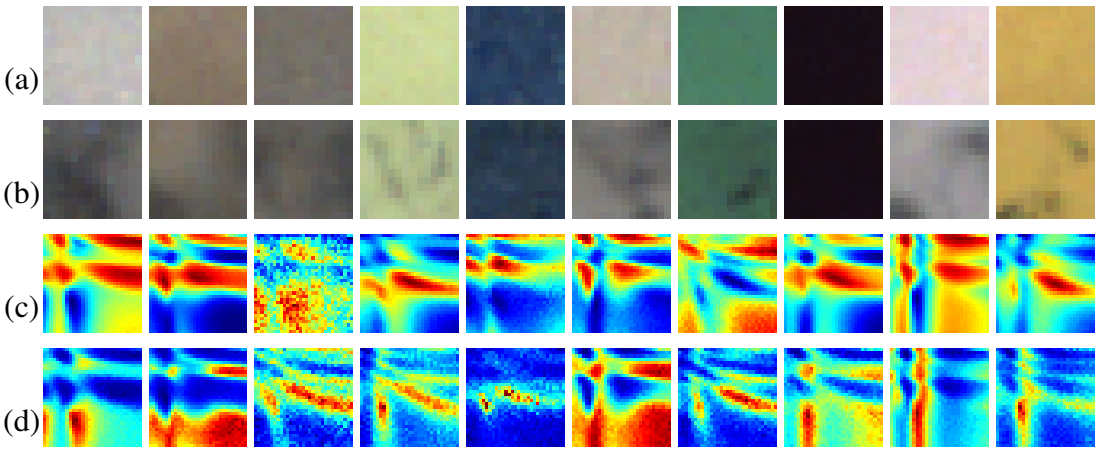


Figure 2.9: Examples of (a) thick powder RGB, (b) thin powder RGB, (c) SWIR signature, and (d) κ signature. The two signatures of many powders are negatively correlated.

a separability criterion based on rough set theory. See Fig. 2.8 for theoretical spectral transmittances of the selected bands.

2.4 The Beer-Lambert Blending Model

Powder appearance varies across different backgrounds and thicknesses. Even with fewer selected bands, capturing such data is hard. Thus, we propose a simple yet effective blending model for data synthesis. Since powders are translucent, the model takes powder color, background color, and thickness into consideration.

2.4.1 Model Description

The model is a per-channel alpha blending where α follows the Beer-Lambert law. Let I_c, A_c and B_c be the intensity of channel c of thin powder, infinitely thick powder (no background visible), and background, respectively. Let x be the powder thickness, and κ_c be the attenuation coefficient related to the powder rather than the background. Then:

$$I_c = (1 - e^{-\kappa_c x})A_c + e^{-\kappa_c x}B_c \tag{2.1}$$

Letting $\eta = e^{-x}$, the model can be rewritten as:

$$I_c = (1 - \eta^{\kappa_c})A_c + \eta^{\kappa_c}B_c \tag{2.2}$$

2.4.2 From Kubelka-Munk Model to Beer-Lambert Blending Model

The Beer-Lambert Blending model can be deduced from the Kubelka-Munk model [112] via approximation. The channel subscript c is ignored below.

Let R, R_∞, and R_g ($0 < R, R_\infty, R_g < 1$) be the absolute reflectance of thin powder, infinitely thick powder, and background, and S be the scattering coefficient. The Kubelka-Munk model is:

$$R = \frac{R_\infty^{-1}(R_g - R_\infty) - R_\infty(R_g - R_\infty^{-1})e^{Sx(R_\infty^{-1} - R_\infty)}}{(R_g - R_\infty) - (R_g - R_\infty^{-1})e^{Sx(R_\infty^{-1} - R_\infty)}} \tag{2.3}$$

Let $\kappa = S(R_\infty^{-1} - R_\infty)$, Equation 2.3 can be re-written as:

$$R = \frac{(1 - R_\infty^2)(R_g - R_\infty)}{(R_\infty R_g - R_\infty^2) - (R_\infty R_g - 1)e^{\kappa x}} + R_\infty \tag{2.4}$$

Since $0 < R_\infty, R_g < 1$, we assume R_∞^2 and $R_\infty R_g$ are small enough to be ignored. The approximate model is:

$$R = \frac{R_g - R_\infty}{e^{\kappa x}} + R_\infty = (1 - e^{-\kappa x})R_\infty + e^{-\kappa x}R_g \tag{2.5}$$

Under constant shading L, $I = LR$, $A = LR_\infty$, $B = LR_g$. Then we obtain the Beer-Lambert Blending Model:

$$I_c = (1 - e^{-\kappa_c x})A_c + e^{-\kappa_c x}B_c \tag{2.6}$$

Since $\kappa = S(R_\infty^{-1} - R_\infty)$, it indicates that κ is negatively correlated to A if the powder scattering coefficient is constant across channels. If we define the κ signature as a 31×31 image formed by the κ values of the 961 channels, similar to the SWIR signature defined in Section 2.3.1, the two signatures should show negative correlation if the scattering coefficient is constant across bands. In practice, 63% of the powders show a Pearson correlation less than -0.5. (Examples in Fig. 2.9)

2.4.3 Parameter Calibration

Algorithm 2 Beer-Lambert Parameter Calibration

Input: Set of thin powder pixels P; Set of RGBN channels C_1; Set of SWIR channels C_2; Thin powder intensity $I_{p,c}$ of each pixel p and channel c; Mean thick powder intensity A_c; Mean background intensity B_c

Output: Attenuation coefficients κ_c for each channel c

```
for each c ∈ C1 ∪ C2 do
    for each p ∈ P do
        t_{p,c} ← −ln((I_{p,c} − A_c)/(B_c − A_c))        # compute κ_c x_p
    end for
    κ_c ← median_{p∈P}{t_{p,c}}                          # compute κ_c median{x_p}
end for
r ← (1/|C1| Σ_{c∈C1} κ_c + 1/|C2| Σ_{c∈C2} κ_c)/2
for each c ∈ C1 ∪ C2 do
    κ_c ← κ_c / r                                        # channel normalization
end for
```

Blending	*Path-thin*		Multi-background	
	RGBN	SWIR	RGBN	SWIR
Alpha	0.028±0.018	0.028±0.020	0.023±0.021	0.022±0.024
Beer-Lambert	**0.018±0.016**	**0.016±0.016**	**0.014±0.016**	**0.012±0.018**

Table 2.4: Fitting error on *Patch-thin* and Multi-background Dataset. RMSE (mean± std) is calculated based on pixel values divided by white patch. Beer-Lambert Blending shows a smaller error than Alpha Blending.

The parameter κ_c can be calibrated by a simple procedure using a small constantly shaded thick powder patch, a thin powder patch, and a bare background patch. The calibration is done by calculating $\kappa_c x$ for each thin powder pixel and normalizing it across pixels and channels (see Algorithm 2). Let P be the set of pixels in the thin powder patch, C_1 be the set of RGBN channels (RGB + NIR), and C_2 be the set of SWIR channels. Let $p \in P$ be a thin powder pixel and $c \in C_1 \cup C_2$ be a channel. Let $I_{p,c}$ be the thin powder intensity, and x_p be the powder thickness. Let A_c and B_c be the average intensity of the thick powder patch and the background patch. Then, we first compute $\kappa_c x_p = -\ln(\frac{I_{p,c}-A_c}{B_c-A_c})$ for each pixel $p \in P$ according to Equation 2.1. Then we calculate $\kappa_c median\{x_p\} = median_p\{\kappa_c x_p\}$, assuming κ_c is the same for each pixel. Since the scale of κ does not matter, we simply let $\kappa_c = \kappa_c median\{x_p\}$. To make κ_c be in a convenient range, we compute the mean κ_c values for RGBN and SWIR channels separately, and normalize κ_c by dividing it by the average of the two values.

We compare the fitting error of Beer-Lambert and Alpha Blending in Tab. 2.4. For a thin patch, we search for the best thickness for each pixel and render the intensity using thick powder intensity, background, thickness and κ. We evaluate RMSE=$\sqrt{\frac{1}{nPixels \times nChannels} \sum (\frac{Rendered-Real}{WhitePatch})^2}$ for each patch in *Patch-thin*. Tab. 2.4 shows that Beer-Lambert Blending fits better than Alpha Blending.

2.4.4 Calibration on Different Backgrounds

The Beer-Lambert Blending model assumes that the attenuation coefficient κ is independent of the background. This section checks if the calibrated κ is invariant to the background used for calibration.

We choose three different backgrounds (Black Aluminum Foil, Brown Leather Hide, Sand Paper) and image a thick sample, three thin samples, and three bare backgrounds in the same filed of view for each powder. We calibrate κ values using different backgrounds. We calculate the coefficient of variation c_v (std/mean) for each powder and each channel. Usually the data is considered low variance if $c_v < 1$. Fig. 2.10 shows the histogram of mean c_v values for RGBN and SWIR channels separately. About 95% of the powders show $c_v < 0.3$, which means that κ calibrated with different backgrounds has a very low variance.

We also calculate the fitting error on this multi-background dataset. As shown in Tab. 2.4, Beer-Lambert Blending has a smaller error than Alpha Blending.

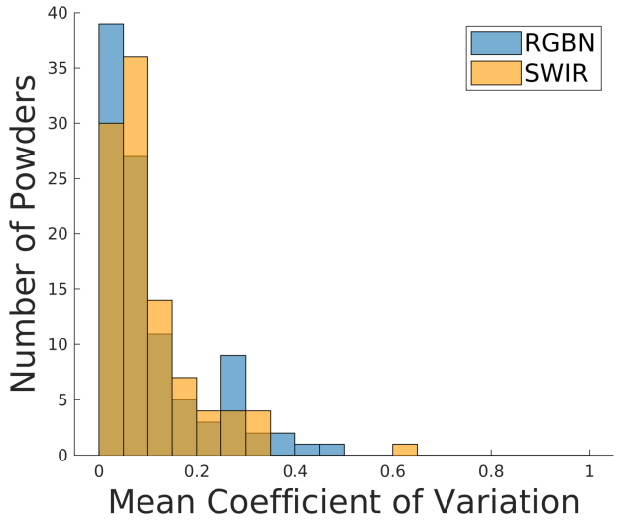


Figure 2.10: Histogram of coefficient of variation c_v of κ calibrated using different backgrounds. Usually the data is considered low variance if $c_v < 1$. About 95% of the powders show $c_v < 0.3$, which means that κ calibrated with different backgrounds has a very low variance.

2.5 Synthesizing Powder against Background Data

Since real data are limited or hard to capture, we propose to render powder against background images. We synthesize a thick powder image with thickness map, and combine it with a real or synthetic background via Beer-Lambert Blending. We use the NYUDv2 [190] dataset and $Patches$ for image-based rendering. Illustration is in Fig. 2.11.

Background Synthesis: NYUDv2 provides RGB images with segmentation labels. We randomly crop an RGB region and its segmentation, and assign a random common material patch from *Patch-common* (Tab. 2.2) to each segmentation class. The synthetic background is obtained by filling the segments with the assigned patch, using image quilting [47] or resizing and cropping. The shading map of the RGB region is estimated via intrinsic image decomposition [127].

Powder Synthesis: Kovacs *et al*. [111] provides a method to estimate smooth shading probability. Its output heatmap looks similar to powder thickness map. We apply the method to images from NYUDv2 to obtain thickness maps. We treat the pixel values (between 0 and 1) in the heatmap as $1 - \eta$ in Equation 2.2. We use the same method as rendering backgrounds to render thick powder images for pixels with positive thicknesses, using patches from *Patch-thick*.

Finally, a random synthetic background and a synthetic powder mask are blended using Equation 2.2, with shading applied. The label is obtained by thresholding $1 - \eta$ at 0.1.

2.6 Implementation Details

1000 powder masks and 1000 backgrounds are rendered. We use the DeepLab v3+ [37] net, taking RGBN and 4 SWIR bands selected by NNCV as input. We train the model from scratch using AdamWR [143] on rendered data, and fine-tune on rendered powders against real backgrounds from *Scene-bg* and *Scene-sl-train* and pure real data from *Scene-sl-train*. *Scene-val* is

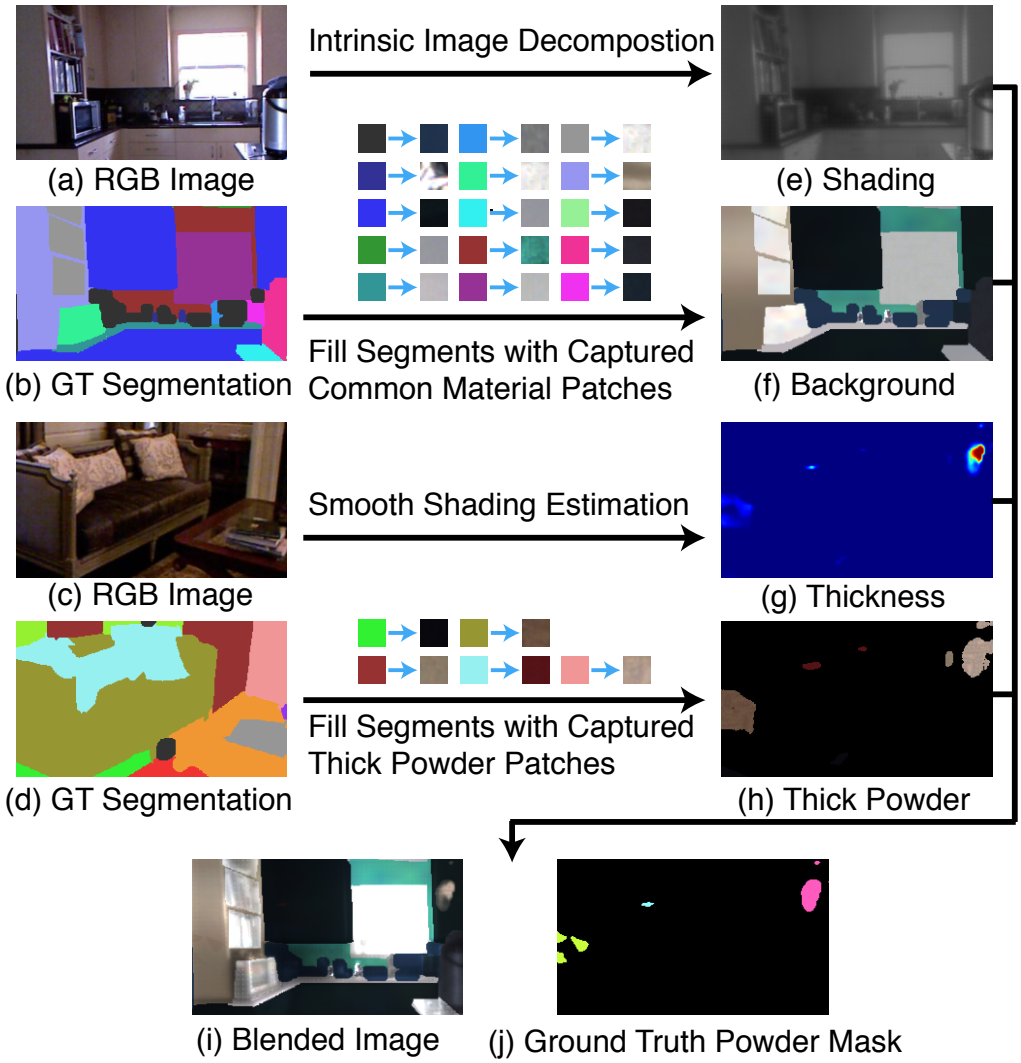


Figure 2.11: Powder against background data synthesis. (a)(c) are RGB regions from NYUDv2 [190], and (b)(d) are their segmentation labels. We obtain the shading (e) via intrinsic image decomposition, and the background image (f) by filling segments in (b) with patches from *Patch-common*. We obtain the powder thickness map (g) via smooth shading estimation, and the thick powder image (h) by filling segments in (d) with patches from *Patch-thick*, only for pixels with positive thicknesses. The final image (i) is obtained by blending background (f) and thick powder (h) using Equation 2.2 with (g) as $1-\eta$, and applying shading (e). The ground truth (j) is obtained by thresholding thickness (g).

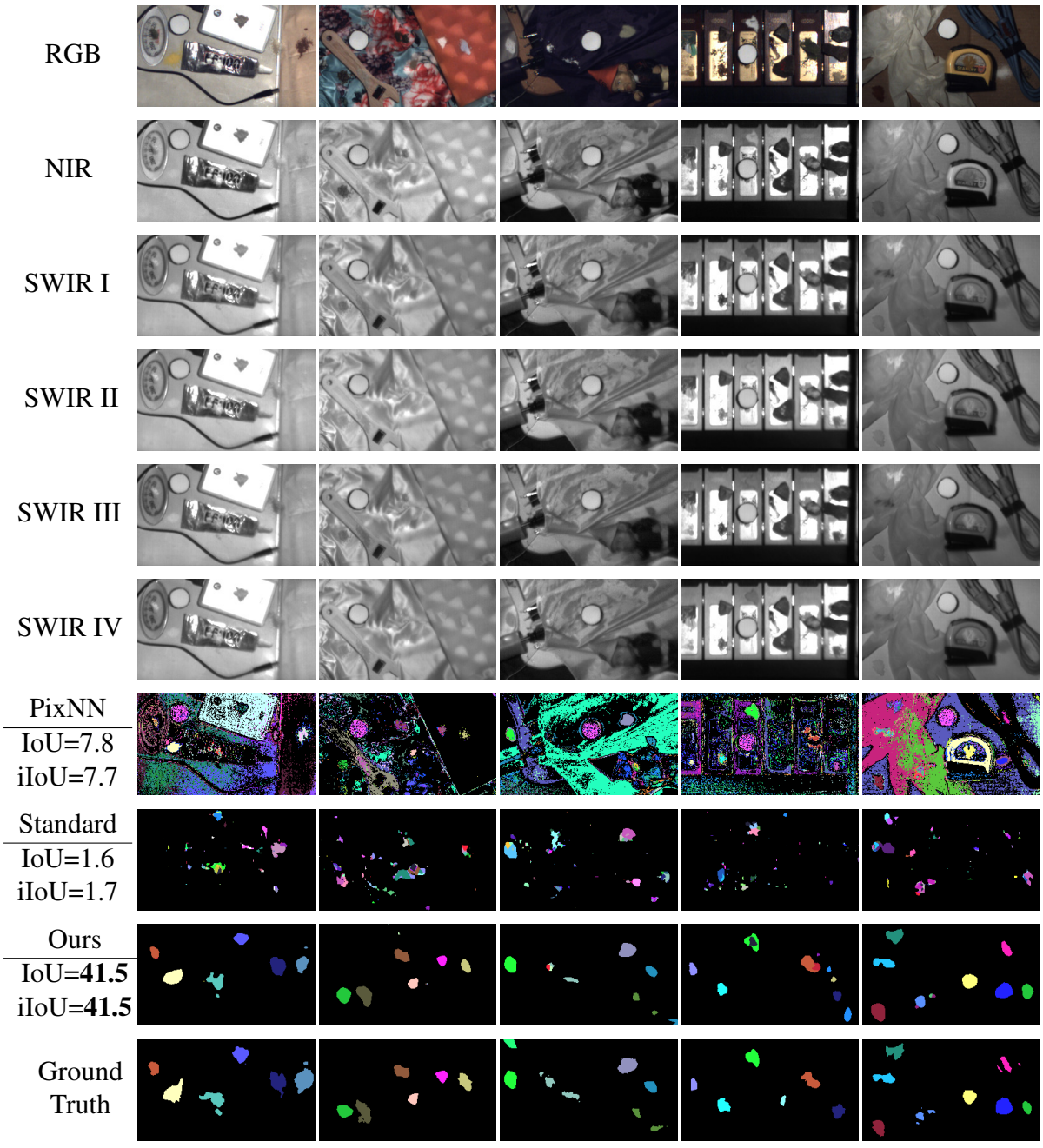


Figure 2.12: Comparisons on *Scene-test* with Per-pixel Nearest Neighbor (PixNN) and Standard Semantic Segmentation (Standard). Black color denotes background while the others denote different powders. Our method performs much better. Band selection and data synthesis lead to huge improvement over simply training on limited real data with all bands.

Blending	Unextended		Extended	
	IoU	iIoU	IoU	iIoU
No Blending	9.7	10.0	29.3	29.9
Alpha Blend	30.2	30.2	39.3	39.5
Beer-Lambert	**36.8**	**37.0**	**42.7**	**42.2**

Table 2.5: Comparison of blending methods. Beer-Lambert Blending is superior than other methods.

for validation. This training procedure can be seen as a progressive domain adaption.

Specifically, we use group normalization [224] instead of batch normalization [82] in DeepLab v3+ network [37]. CRF [36] postprocessing is used. We use the AdamWR [143] optimizer and cosine annealing with warm restart scheduler. We set initial restarting period= 8, and $T_{mult} = 2$. Thus, the scheduler restarts at 8, 24, 56, 120, and 248 epochs. We use batch size = 8 and weight decay = 1e-4 for all experiments.

We first train the model from scratch on synthetic powders against synthetic backgrounds with initial learning rate = 1e-3 for 248 epochs. In each iteration, we find a random synthetic background for the current powder mask, and blend them to render a scene. We call it an epoch when it goes through all powder masks once. Then we fine-tune it on synthetic powders against real backgrounds from *Scene-bg* and *Scene-sl-train* with initial learning rate = 1e-4 for 56 epochs. We finally fine-tune it on real powders against real backgrounds from *Scene-sl-train* with initial learning rate = 5e-5 for at most 56 epochs. Model selection is done according to the performance on validation set.

2.7 Experimental Analysis

The algorithm should distinguish between backgrounds and powders, leading to a 101-class semantic segmentation task (background+100 powders). We train a deep net using synthetic data and limited real data for this task. We show that our method is superior by comparing with baselines, and that the Beer-Lambert Blending is necessary via ablation study.

Evaluation Metrics: We report mean intersection over union (IoU) and mean instance-level intersection over union (iIoU) borrowed from Cityspaces [42]. We define the pixels with the same label in the same image as an instance.

Comparison with Baselines: We compare on *Scene-test* with two baselines: Per-pixel Nearest Neighbor (PixNN) finds per-pixel nearest neighbor in a database including mean patch values from *Patch-thick* and *Patch-common*. Standard Semantic Segmentation (Standard) trains DeepLab v3+ [37] on pure real data from *Scene-val* with RGBN and 961 SWIR bands. In Fig. 2.12, our method significantly outperforms the two baselines.

Blending Methods: We conduct two types of experiments to compare different blending methods: (1) **Unextended** experiments do not include *Scene-sl-train* in training, and evaluate on *Scene-test* only. (2) **Extended** experiments include *Scene-sl-train* in training and evaluate on a

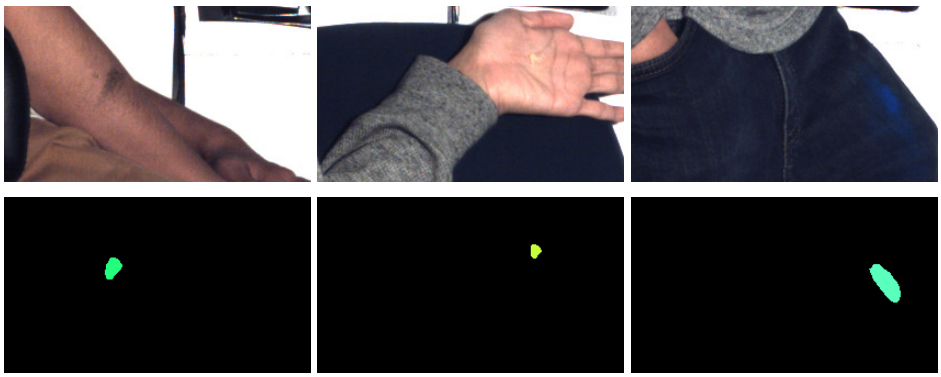


Figure 2.13: Powder recognition on arm, palm and jeans. The model is fine-tuned on ten human images with rendered powder. We vote for majority class in each connected component, and preserve components with confidence ≥ 0.95.

dataset merging *Scene-test* and *Scene-sl-test*. Tab. 2.5 show that Beer-Lambert Blending is better than other settings.

Presence/Absence Test: Security applications often care about the presence/absence of a specific powder rather than its exact mask. Thus, adjusting the confidence threshold, we plot the ROC curve and PR curve for this 2-class classification task in Fig. 2.15, showing the significant superiority of our method over the baselines.

Failure Cases: Fig. 2.14 shows failure cases where the algorithm misdetects some small objects as powders and misses some powders on cluttered backgrounds.

Application: It takes about 3s for capturing 4 SWIR channels, which could be used in time-sensitive applications (*e.g.* scenes with human in Fig. 2.13).

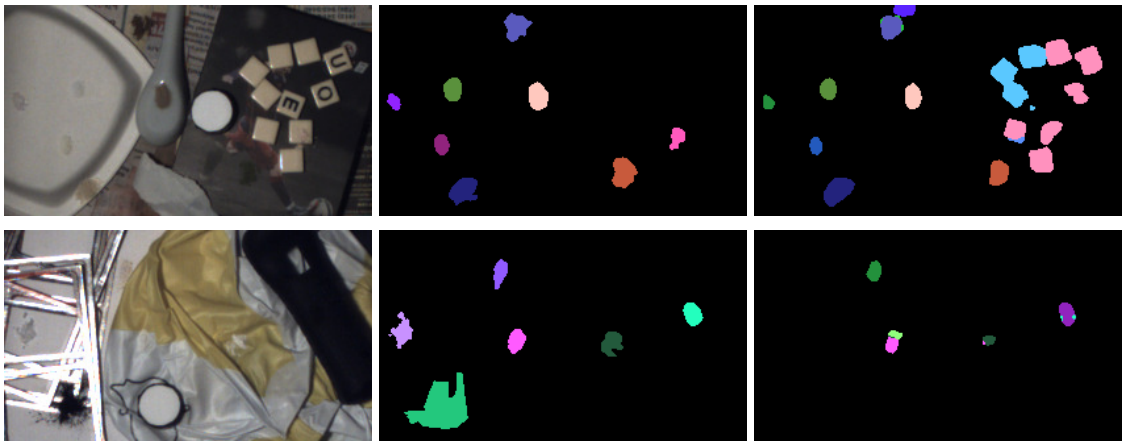
2.8 Limitations

One limitation of our work is that the accuracy may not be sufficient for a safety application that demands near perfect detection of dangerous powders. It may be improved by (a) adding more data to reduce false positives on backgrounds and/or (b) considering a wider spectral range including mid-wave IR ($2\sim5\mu$m).

Besides, the proposed blending model assumes that the powder appearance is caused by diffuse reflection. However, some powder samples, including metallic ones and glowing ones, do not follow this assumption. Special models should be used to describe the appearances of such powder samples.

2.9 Conclusion

We achieve over 40% mean IoU on powder recognition without known location. This performance is strong considering the fine-grained 101-class recognition problem. Even if powder recognition may not achieve perfect accuracy using solely visual cues, a visual recognition sys-



(a) RGB (b) Ground Truth (c) Prediction

Figure 2.14: Failure cases. Row 1 misdetects small square objects; Row 2 misses powders on cluttered background.

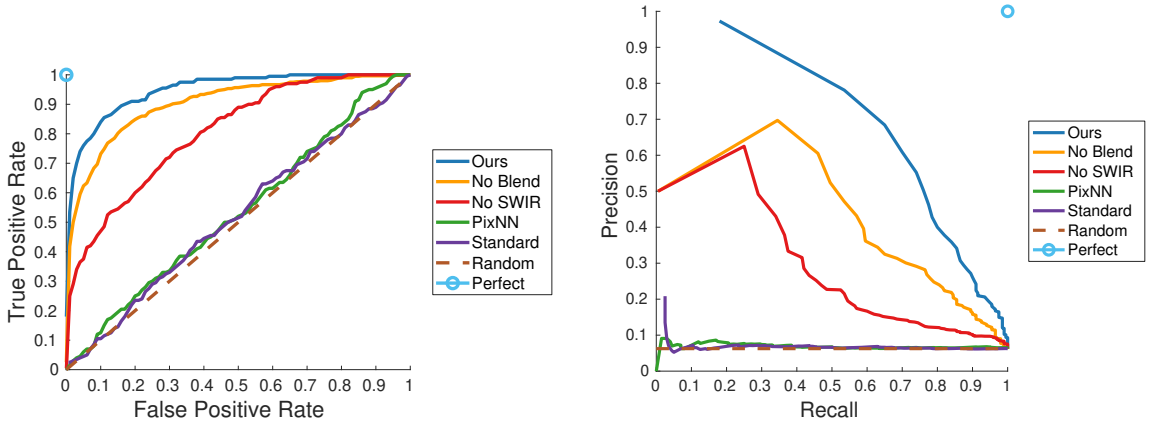


Figure 2.15: ROC curve and PR curve on *Scene-test*. Incorporating band selection and data synthesis, our method outperforms Per-pixel Nearest Neighbor (PixNN) and Standard Semantic Segmentation (Standard).

tem can eliminate most candidates, and the top-N retrievals can be further tested via other means (microscopic, chemical).

In summary, to provide an example where synthetic ground truth supervision from material-aware appearance model can be used for training deep networks, we present an approach of recognizing translucent powders via synthetic data generation. We build an appearance model for rendering translucent powder on background images, taking powder color, background color, and thickness into consideration. By training via ground truth supervision on the synthetic data based on this model and a very small amount of real data, the deep network achieves a reasonable accuracy.

Chapter 3

Photometric Supervision from Appearance Model

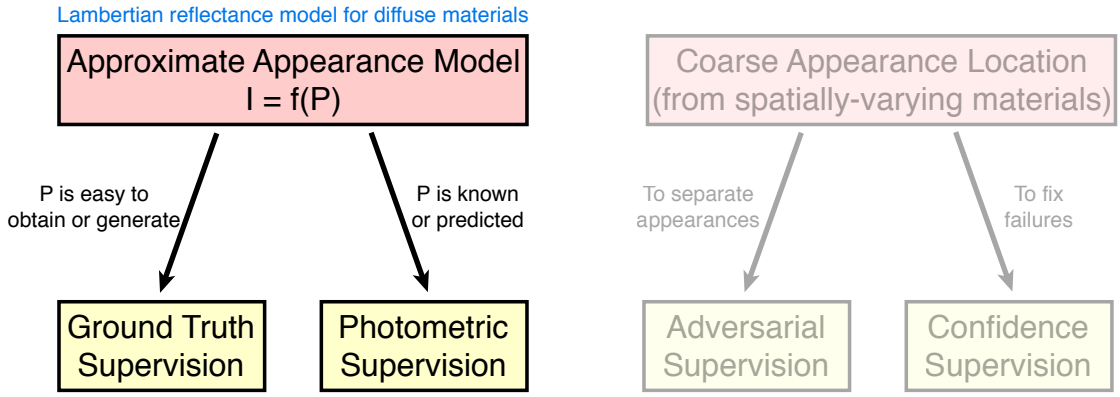
In the last chapter, we show an example of using material-aware appearance model in the form of ground truth supervision for training deep networks. This is driven by synthetic data generation. To handle the domain shift problem caused by synthetic data, Chapter 2 finetunes the model on a small amount of real data. However, ground truth annotations on real data are not always available. In such case, other supervision signals are required to train deep networks. Photometric supervision is one common choice. Thus, in this chapter, we present a method for human reconstruction, as an example of using photometric supervision from appearance model for adaptation on real data, after the pre-training on synthetic data. The supervision signals are illustrated in Fig. 3.1.

Specifically, we pick human, which is generally diffuse, as example, and reconstruct the albedo texture and detailed geometry from an RGB-D video, given the knowledge of lighting and a 3D human model. To mitigate the domain shift between synthetic data and real data, we first train the deep model with synthetic data and then optimize it on real data via self-supervised learning. The Lambertian appearance model is used to build the photometric equation between albedo, normal and intensity. Because the information provided by photometric supervision is inaccurate and inadequate, additional priors (human model, temporal priors) are used for better reconstruction.

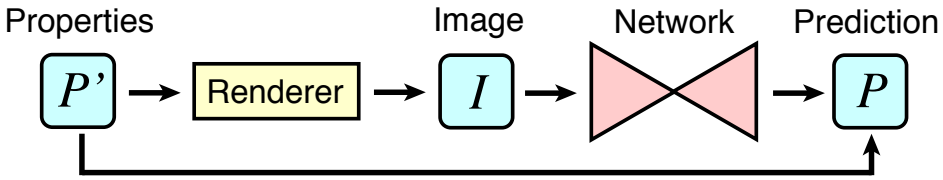
3.1 Application: Human Reconstruction from RGB-D Video

An essential component of VR communication, modern game and movie production is the ability to reconstruct accurate and detailed human geometry with high-fidelity texture from real world data. This allows us to re-render the captured character from novel viewpoints. This is challenging even when using complex multi-camera setups [41, 93, 147, 207]. Recent works such as Tex2Shape [10] and Textured Neural Avatars [189] (TNA) have shown how to reconstruct geometry and texture respectively using nothing but a single RGB image/video as input.

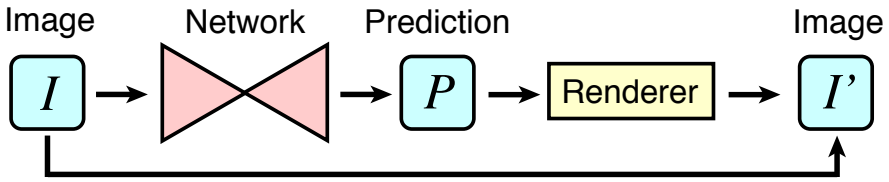
Fig. 3.2 shows examples of novel viewpoint synthesis using Tex2Shape for geometry reconstruction and TNA for texture estimation. Both Tex2Shape and TNA incorporate image synthesis



(a) Framework



(b) Ground Truth Supervision



(c) Photometric Supervision

Figure 3.1: The human reconstruction task uses appearance model information from material awareness for ground truth and photometric supervisions. We first train deep models with synthetic data and then adapt them on real data via self-supervised learning. Both synthetic data generation and self-supervised learning rely on the Lambertian appearance model.

(a) RGB (b) Tex2Shape [10]+TNA [189] (c) Ours

Figure 3.2: Tex2Shape [10] + TNA [189] vs. our result. The mesh with texture is rendered from different viewpoints. Our approach reconstructs more detailed geometry, such as the moving jacket, as well as more accurate texture.

into deep learning. However, Tex2Shape is a single view method trained only on synthetic images without adaptation to real data. Thus the domain shift problem is not handled: It generates the rough shape of the actor but misses some of the finer geometric details that appear in the real data (Fig. 3.2 (b)), and often hallucinates the incorrect deformation memorized from its training data especially in occluded parts. TNA is based on analysis by synthesis. By assuming "constant color" without material awareness, it ignores the lighting effect. The estimated texture contains the baked-in lighting of the original input sequence. Besides, due to small geometric misalignments, the estimated texture is blurry.

To address these issues, we introduce a novel framework to reconstruct both significantly higher quality mesh and texture from a real world video (see Fig. 3.2 (c)). Our model takes an RGB video, a corresponding environment map, and a per-frame coarse mesh as inputs, and produces a per-frame fine mesh and a high-resolution texture shared across the whole video that can be used for free-viewpoint rendering. The coarse mesh is a parametric human model obtained by 3D tracking from an RGB-D camera [209].

To mitigate the domain shift between synthetic data and real data, we first train our model on synthetic images and then adapt on a short real video clip with photometric supervision. Lambertian assumption is made in both two stages. This material assumption significantly reduces the rendering complexity in loss calculation because a simple mathematical approximation can be built via spherical harmonics lighting [169]. Because the appearance model is inaccurate and cannot provide enough information for reconstruction, we incorporated temporal priors and a human model as additional knowledge into the pipeline.

Concretely, for texture generation, we parameterize the texture using a CNN and optimize it on real data by comparing the rasterized images with a limited number of selected key albedo images. Our design offers three benefits: no shading and texture mixing, less geometric misalignment leading to less blur, and built-in CNN structure prior for noise and artifact removal [206].

For mesh reconstruction, we propose to first pre-train a displacement map prediction model on synthetic images with supervision, and later optimize it on a real sequence in a self-supervised manner using photometric perceptual loss and spatiotemporal deformation priors to obtain detailed clothing wrinkles even for occluded parts.

Experiments show that the proposed method provides clear texture with high perceptual quality and detailed dynamic mesh deformation in both the visible and occluded parts. The resulting mesh and texture can produce realistic free-viewpoint rendering (Fig. 3.16) and relighting results (Fig. 3.17).

3.2 Related Work

Human Shape Reconstruction. The key to human shape reconstruction is to incorporate human priors to limit the solution space. Template-based methods [67, 227] obtain human geometry by deforming the pre-scanning model. Model-based methods [24, 77, 84, 94, 117, 160, 171, 228, 247] fit a parametric naked-body model [142] to 2D poses or silhouette. While these methods estimate the coarse shape well, the recovered surface geometry is usually limited to tight clothing only [23]. To tackle this problem, [234, 235] combine depth fusion [157, 158] and human priors [142] and show highly accurate reconstruction in visible parts but not occluded regions. With multiple images, [7, 8, 9] model clothing by deforming a parametric model [142] to obtain an animatable avatar, which enables powerful VR applications. However, the clothing details are inconsistent across frames, making the re-targeting result not faithful to the observation. Some methods treat clothing as separate meshes, providing strong possibilities for simulation, but are limited to a single clothing [114], pre-defined categories [20], or mechanical properties [236]. Recently, single image methods utilize deep learning for recovering detailed shapes, including UV space methods [10, 114], volumetric methods [244], implicit surface [78, 174], and method combining learning and shading [249]. They provide excellent details in visible regions, but hallucinate invisible parts rather than using temporal information for faithful reconstruction. In contrast to the above methods, we exploit photometric and spatiotemporal deformation cues to obtain detailed mesh, even in occluded regions.

Human Texture Generation. The key of texture generation is to fuse information from multiple images. Sampling based methods [7, 8] sample colors from video frames and merge them together. TNA [189] uses photometric supervision from rendering. These methods work well for videos with limited deformation but fail when the misalignment caused by large clothing deformation is significant. Single view methods [63, 159] avoid the problem of fusing multi-view information by hallucinating the occluded part. Yet, the hallucinated texture may not match the real person. Different from these methods, our method handles large deformation and provides high quality albedo texture.

Face Reconstruction. Face modeling is closely related to body modeling but with limited self-occlusion. Methods using photometric cues reconstruct detailed geometry and albedo via self-supervision [200, 201]. Deep learning also provides the opportunity for learning face geometry from synthetic data [175] or both synthetic data and real data [178]. These methods achieve high

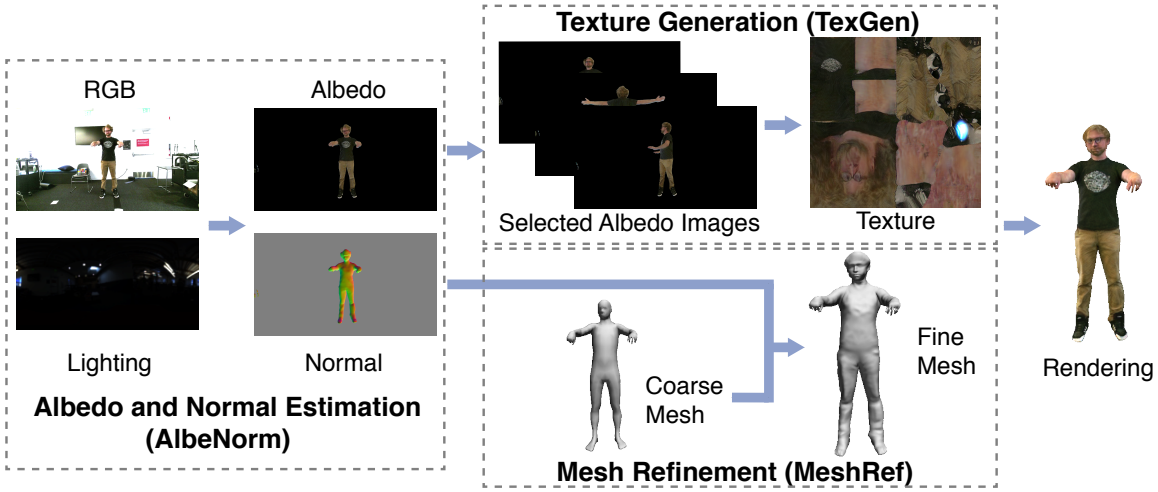


Figure 3.3: Framework overview. Our method consists of three modules: Albedo and Normal Estimation (AlbeNorm) pre-precosses RGB images to estimate albedo and normal; Texture Generation (TexGen) selects key albedo frames and recovers a texture map; Mesh Refinement (MeshRef) takes a coarse mesh and a normal image as input and outputs a fine mesh. We pre-train AlbeNorm and MeshRef on synthetic data. Then given a short clip of the real video, we optimize TexGen to obtain texture and finetune MeshRef via self-supervision. Finally, we run AlbeNorm and MeshRef on the whole video for fine meshes.

quality results but cannot be trivially extended to the full body, especially for occluded parts.

3.3 Method

As in Fig. 3.3, our framework consists of three modules: Albedo and Normal Estimation (AlbeNorm), Texture Generation (TexGen), and Mesh Refinement (MeshRef). AlbeNorm takes an RGB image and the lighting, represented using Spherical Harmonics [169], and estimates texture and geometry information in the form of albedo and normal images. This is used consecutively to refine the texture and geometry estimates: TexGen selects albedo key frames and generates a high-resolution texture map from them. MeshRef takes a coarse mesh from RGB-D tracking and a normal image and estimates a refined mesh. Ground truth data for these tasks is naturally scarce. However, we observe that (1) synthetic data can be used to train the AlbeNorm and MeshRef. In synthetic settings we can use datasets with detailed person models to obtain highly detailed geometry estimates; (2) TexGen and MeshRef can be finetuned on a short sequence via self-supervised learning, using perceptual photometric loss and the spatiotemporal deformation priors in a *self-supervised* manner. This makes training on large annotated video datasets obsolete. While we train AlbeNorm using only synthetic data, the model empirically generalizes well to real data. The final results are a single high-resolution full-body texture for the whole sequence and fine body geometry predicted at every frame. We describe our method in detail in the following sections.

3.3.1 Albedo and Normal Estimation

The cornerstone for our method is a good albedo and normal estimation: the normal is key to recover detailed geometry in MeshRef and the albedo is the key to estimate clear texture in TexGen. To extract albedo and normals, under the usual assumptions of Lambertian materials, distant light sources, and no cast shadows, we can fully represent the geometry and color of an image using a normal image and an albedo image. The normal encodes the local geometry information, and together with the incident illumination, it can be used to generate a shading image. The albedo encodes the local color and texture. The decomposition into shading and albedo is typically not unique, as we can potentially explain texture changes through normal changes. This is where the AlbeNorm module comes into play: to prevent shading from 'leaking' into texture and the albedo gradients from being explained as geometry changes, we use the module to decouple the two components. Unlike [7], we resolve the scale ambiguity with the known lighting.

The AlbeNorm module uses a CNN to predict albedo and normal images. The inputs to this module are a segmented human image [35, 165] and the incident illumination represented as Spherical Harmonics [169]. Knowing the lighting information, we omit the scene background and process the masked human region. Concretely, let A_p and A_g be the predicted and ground truth albedo images, N_p and N_g be the predicted and ground truth normal images, and M the human mask, respectively. Then, our supervised loss L_{AN} with weights λ_a^{an} and λ_n^{an} is:

$$L_{AN} = \lambda_a^{an}||(A_p - A_g) \cdot M||_1 + \lambda_n^{an}||(N_p - N_g) \cdot M||_1, \tag{3.1}$$

To faithfully recover the color and texture for rendering applications, the albedo and normal should be consistent. In other words, the image synthesized using the albedo and normal should match the original image. However, as shown in Fig. 3.4, due to the domain gap between real and synthetic data, the synthesized image (g) does not have a similar appearance as the original one (a). Another way to obtain consistent albedo is to use the normal N_p and the input illumination to estimate the shading [169] (c), and estimate the albedo (e) by dividing the image (a) by this normal estimated shading. This albedo (e) is consistent with the estimated normal N_p, and thus has the correct color and global scale. Yet it does not have a "flat" appearance, which means there is residual shading information included due to incorrectly estimated normals. The estimated albedo A_p in (d) on the other hand correctly factored out the shading, but is not consistent with the normal image. To consolidate the two estimates, we modify (d) by taking the color and scale from (e), to obtain an albedo (f) which is consistent with the normal image and at the same time has a "flat" appearance.

Concretely, let I be the per-pixel intensity of R,G,B channels: $I = (R + G + B)/3$, and $med(I)$ be the median intensity within human mask. We first take the color from (e) as $R' = I_d/I_e \times R_e$ and globally scale it to (e) as $R = med(I_e)/med(I') \times R'$. B and G are obtained similarly. The resulting albedo (f) is consistent with the normal image (b) and the newly synthesized image (h) better matches the original image (a).

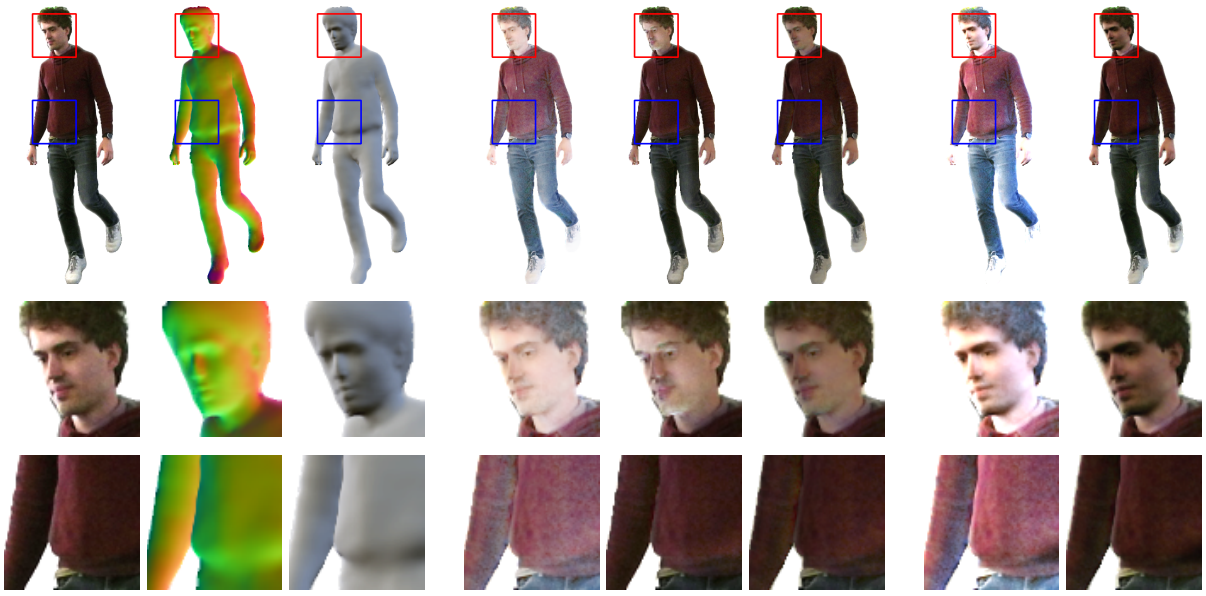


Figure 3.4: Intermediate results of AlbeNorm. (a) original RGB image; (b) predicted normal; (c) calculated shading; (d) albedo directly from CNN; (e) albedo from dividing RGB by shading; (f) final albedo; (g) rendering using (d); (h) rendering using (f). The final albedo (f) includes less shading information than (e) (e.g., the face region), and (h) resembles the original RGB (a) better than (g).

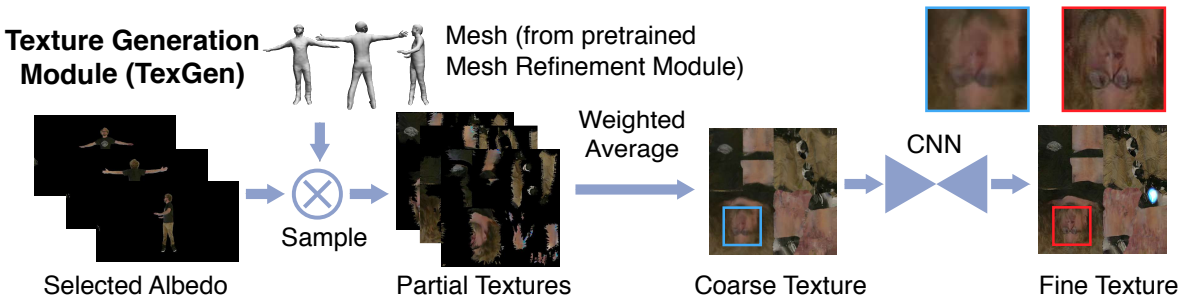


Figure 3.5: Texture generation module (TexGen). TexGen selects K albedo images and converts them into partial textures. A coarse full texture is constructed by averaging the partial textures with weights. The texture is then refined by a CNN optimized from scratch for each video. The supervision comes from rasterizing the texture to the image space and comparing it with the input albedo images.

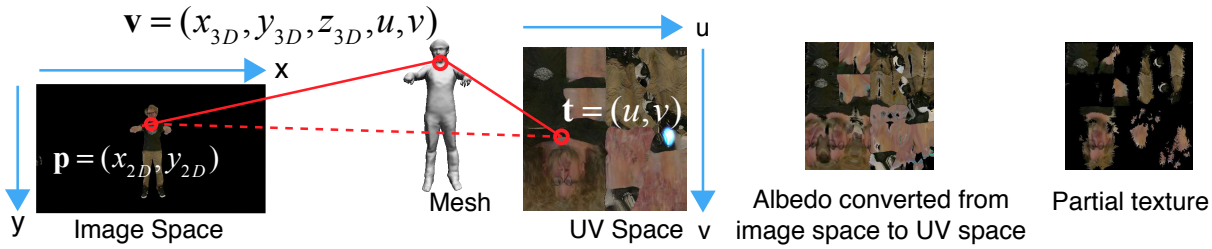


Figure 3.6: The mesh bridges image space and UV space. Assume point $\mathbf{v}$ on the mesh is associated with its 3D position (x_{3D}, y_{3D}, z_{3D}) and UV coordinates (u, v). The corresponding point $\mathbf{p} = (x_{2D}, y_{2D})$ in image space can be obtained via camera projection. It also corresponds to point $\mathbf{t} = (u, v)$ in UV space. To convert features from image space to UV space, for each $\mathbf{t}$, we first obtain its 3D position (x_{3D}, y_{3D}, z_{3D}) by barycentric interpolation, project it to image coordinates (x_{2D}, y_{2D}), and finally sample features from the image.

3.3.2 Texture Generation

TexGen is used to encode and refine the person texture. Inspired by TNA [189], the texture is obtained by optimizing the photometric loss between rendered images and the original ones. We propose to (1) use albedo instead of the original image to prevent shading leaking into texture, (2) select keyframes to mitigate geometric misalignment, and (3) use a CNN to parameterize the texture to reduce noise and artifacts. We assume the availability of MeshRef (Sec. 3.3.3) pre-trained on synthetic data. Fig. 3.5 shows the TexGen pipeline.

UV Mapping. We follow the common practice in graphics where texture is stored using a UV map [21]. This map unwraps a mesh into 2D space, called UV space. Each pixel $\mathbf{t} = (u, v)$ in UV space corresponds to a point $\mathbf{v}$ on the mesh, and thus maps to a pixel $\mathbf{p} = (x_{2D}, y_{2D})$ in the image via projection. Specially, the 3D position is defined by the barycentric interpolation of the vertices of the face where the point is on. With the 3D position, we can project it to the image space of a calibrated camera. Thus, we can sample image features and convert them into UV space. Fig. 3.6 shows an example of converting albedo to UV space. It is further converted into a partial texture by masking with visibility. To calculate visibility, as shown in Fig. 3.7, we rasterize an image with UV coordinates, then sample that to UV space, and compare with the correct UV coordinates. The pixels whose sampled UVs are consistent with its position in UV space are visible. By masking the sampled albedo with visibility, we obtain the partial texture.

Key Frame Selection. We aim to extract a sharp texture from a short video sequence. Inherently, this is difficult because of misalignments. We aim to address these issues through selection of a few, particularly well suited frames for this task. Our selection should cover the whole body using a small number (K) of albedo frames based on the visibility of the partial texture image. More concretely, let V_i be the visibility UV map for the i-th selected frame and V_0 be the visibility map of the rest pose mesh. Since most salient parts (e.g., faces) are visible in the rest pose, we first select the frame closest to the rest pose by minimizing $||V_1 - V_0||_1$. We then greedily add the i-th frame by maximizing the total visibility $||\max_{j=1}^{i} V_j||_1$, until K frames are selected. We also assign a sampling frequency weight of $W_i = 1/K + w_i / \sum_{i=1}^{K} w_i$, where $w_i = ||V_i - \max_{j=1}^{i-1} V_j||_1$, to every i-th frame. These weights bias the training to more visible

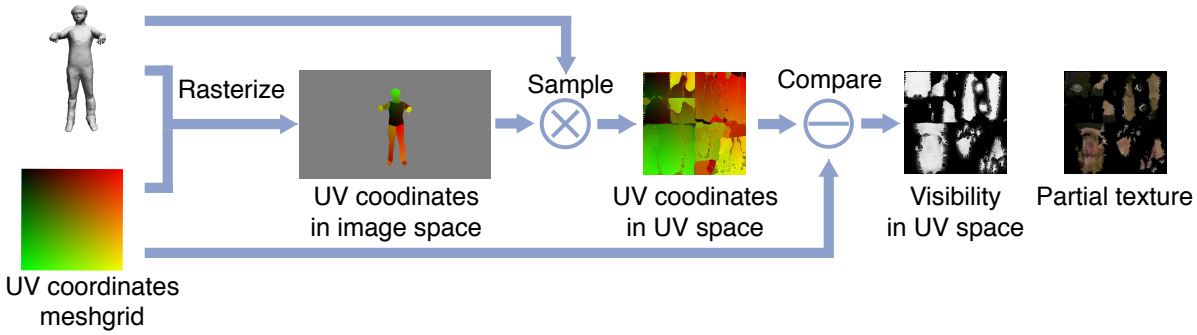


Figure 3.7: Visibility and partial texture generation. By rasterizing UV coordinates to image space and sample it back to UV space, we obtain a UV coordinates map where only the visible parts are ”correct”. By comparing it with the ground truth UV meshgrid, we obtain the visibility map in UV space. We further calculate partial texture by masking the sampled albedo.

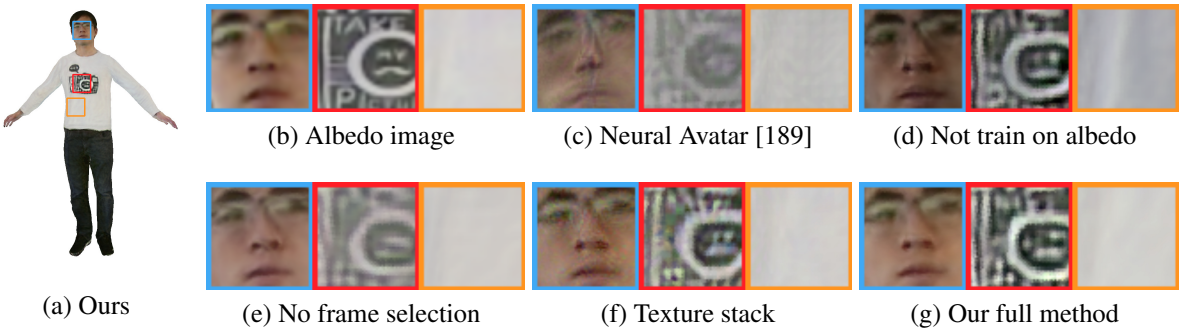


Figure 3.8: Rasterized albedo using generated texture. (b) is the albedo from AlbeNorm, which can be seen as “ground truth”. Training on the original image rather than albedo (d) causes the texture to include shading. No frame selection (e) makes the result blurry. Using a texture stack instead of a CNN (f) creates a noisy face. (c) is TNA [189] (trained on original images using a texture stack, without frame selection). These issues are addressed by our full method (a)(g).

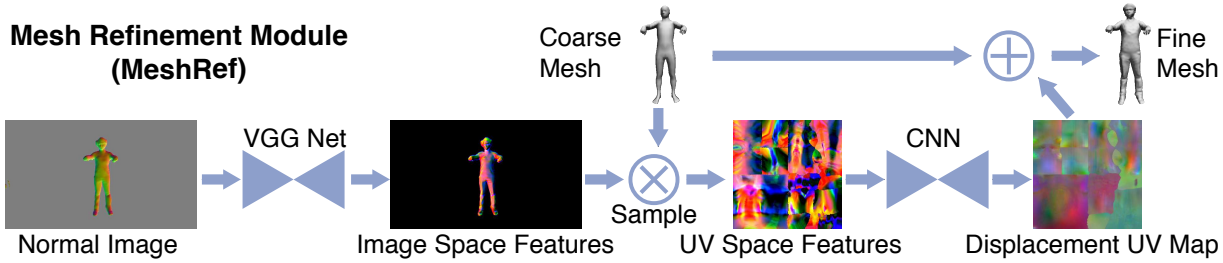


Figure 3.9: Mesh refinement pipeline (MeshRef). MeshRef first extracts features from a normal image, then convert those features to UV space. The UV space features are then sent to a CNN to predict a 3D displacement map. We obtain the fine mesh by adding the displacements to the coarse mesh. This module is first trained using synthetic data with ground truth and later self-adapted on a short real sequence via photometric and temporal losses.

frames and speed up the convergence. In practice, we add two adjacent frames with $w = w_i/2$ of the i-th frame for denoising. Fig. 3.8 shows the benefit of our selection scheme which leads to more detailed and accurate reconstructions.

Texture Refinement CNN. From the key albedo frames, we generate partial textures and obtain a coarse texture using a weighted average, where the weight is $w_i V_i$. The coarse texture is processed by a CNN for generating a fine texture, using a deep image prior [206] for denoising and artifact removal (see faces in Fig. 3.8 (f) and (g) for the benefit of our CNN-based texture paramaterization over the texture stack of TNA [189]). The loss comes from rasterizing the texture to the image space and comparing it with the selected albedo images. The gradients are also back-propagated to the mesh and thus MeshRef for a better alignment. This mesh adjustment scheme is a crucial component of the TexGen module.

Concretely, let R be the albedo image rasterized using SoftRas [140], A be the albedo image from AlbeNorm, and M be the human mask from segmentation [35, 165]. We use an L_1 photometric loss and a perceptual loss [90] to compare R and A within M. We further regularize the mesh deformation by an L_1 loss between the Laplacian coordinates [195] of the current vertices and the vertices from the initial pre-trained MeshRef model. Our total loss L_{TG} is written as:

$$L_{TG} = \lambda^{tg}_{L1}||(R - A) \cdot M||_1 + \lambda^{tg}_{pct} l_{pct}(R \cdot M, A \cdot M) + \lambda^{tg}_{lap} \sum_{i \in V} ||\mathbf{v}'_{p,i} - \mathbf{v}'_{o,i}||_1, \tag{3.2}$$

where V is the vertex index set and $\mathbf{v}'_{p,i}$ and $\mathbf{v}'_{o,i}$ are the Laplacian coordinates of the i-th predicted vertex and original vertex, respectively. $l^{tg}_{pct}(x, y)$ computes an adaptive robust loss function [16] over the VGG features [192] as perceptual loss [90]. λ^{tg}_{L1}, λ^{tg}_{pct}, and λ^{tg}_{lap} are weights. In practice, we empirically limit the deformation of small structures such as the head and hands by locally disabling gradients, because of possible large mesh registration errors in these regions.

3.3.3 Mesh Refinement

The MeshRef module is used to refine the coarse mesh. Fig. 3.9 gives an overview of its design. Inspired by the effectiveness of predicting human shape deformation in UV space [10], MeshRef

converts the image features into UV space to predict 3D displacement vectors. Our design takes the normal map from AlbeNorm and extracts VGG features [192] to obtain a better encoding, before converting the features to UV space. In addition to VGG [192] features, we further augment the features by including vertex position information. Specifically, we can rasterize coarse vertex position into both image space and UV space, to become a vertex position image I_{vp} and a vertex position map T_{vp}. We append I_{vp} to the input of VGGNet, and append T_{vp} to the input of the UV space CNN.

To learn human shape priors, we pre-train MeshRef on a synthetic dataset with supervision. However, due to the domain gap, the pre-trained model does not perform well on real data. Thus, after obtaining the texture from TexGen, we adapt MeshRef on a real sequence using a photometric loss between the textured mesh and the original image. We also apply a motion prior loss [208] to enhance short-term temporal consistency. Since these losses cannot provide tight supervision for invisible regions, we further use a deformation loss to propagate the deformation from frames where those regions are visible to frames where they are not. This model is trained on batches of consecutive video frames.

Supervised Training on Synthetic Images. We supervise the 3D displacement maps using L_1 and SSIM losses and regularize the 3D vertices using a Laplacian loss. Let D_p and D_g be the predicted and ground truth displacement maps and $DSSIM = (1 - SSIM)/2$ be the structural dissimilarity function [218]. Our loss L_{MR1} is defined as:

$$L_{MR1} = \lambda_{L1}^{mr1}||D_p - D_g||_1 + \lambda_{ssim}^{mr1}DSSIM(D_p, D_g) + \lambda_{lap}^{mr1}\sum_{i\in V}||\mathbf{v}'_{p,i} - \mathbf{v}'_{g,i}||_1, \tag{3.3}$$

where $\mathbf{v}'_{p,i}$ and $\mathbf{v}'_{g,i}$ are Laplacian coordinates defined similar to Eq. 3.2, and λ_{L1}^{mr1}, λ_{ssim}^{mr1}, and λ_{lap}^{mr1} are the weights between different losses.

Self-supervised Training on Real Video Data. For self-supervised training, we render the images using the SoftRas differentiable renderer [140] and spherical harmonics lighting [169] and compare with the original images. Our self-supervised loss is defined as:

$$\begin{aligned} L_{MR2} =& \lambda_{pct}^{mr2}L_{pct} + \lambda_{sil}^{mr2}L_{sil} + \lambda_{temp}^{mr2}L_{temp} + \\ & \lambda_{pos}^{mr2}L_{pos} + \lambda_{lap}^{mr2}L_{lap} + \lambda_{deform}^{mr2}L_{deform}, \end{aligned} \tag{3.4}$$

where $L_{pct}, L_{sil}, L_{temp}, L_{pos}, L_{lap}, L_{deform}$ are the perceptual loss, the silhouette loss, the motion consistency loss, the vertex position loss, Laplacian loss, deformation propagation loss, and $\lambda_{pct}^{mr2}, \lambda_{sil}^{mr2}, \lambda_{temp}^{mr2}, \lambda_{pos}^{mr2}, \lambda_{lap}^{mr2}, \lambda_{deform}^{mr2}$ are their corresponding weights, respectively. We introduce the losses below. For simplicity, we present the losses for one frame, omitting the summation over all frames.

Perceptual Loss. Let R be the rendered image, I be the original image, M_R be the rasterized silhouette and M_I be the segmented human mask, and $M = M_R \cdot M_I$. The loss is defined as $L_{pct} = l_{pct}(R \cdot M, I \cdot M)$, where l_{pct} is the robust perceptual loss function [16, 90].

Silhouette Loss. This loss compares the rasterized silhouette and the segmented human mask is defined as $L_{sil} = ||(M_R - M_I) \cdot C||_1$, where C is the confidence map given by the segmentation algorithm [35, 165].

Motion Consistency Loss. Let t be the current frame index and $\mathbf{v}_{p,i}^{(t)}$ be the position of the i-th vertex in frame t. Our motion consistency loss favors constant velocity in adjacent frames [208]

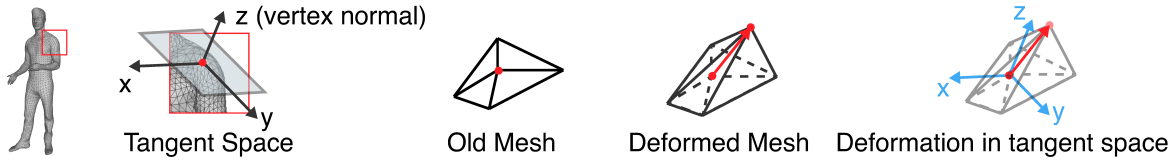


Figure 3.10: Deformation in tangent space. At a local point, the z-axis points to the vertex normal direction and the x and y axes complete the orthogonal basis, forming a local coordinate system. We use this coordinate system to represent the vertex deformation. This representation is invariant to pose change, propagating deformation of the same vertex in different frames.

(a) Source frame (b) Target (c) No propagation (d) Propagation

Figure 3.11: Effect of deformation propagation. (a) is a source frame propagating deformation to frame (b) where the back is not visible. The back of (b) is reconstructed without (c) and with (d) deformation propagation. The one with propagation shows more clothing details.

and is written as $L_{temp} = \sum_{i \in V} ||\mathbf{v}_{p,i}^{(t-1)} + \mathbf{v}_{p,i}^{(t+1)} - 2\mathbf{v}_{p,i}^{(t)}||_1$, where V is the set of vertex indices.

Vertex Position Loss. This loss prevents large deformation from the original position predicted by the model pre-trained on synthetic data and is defined as $L_{pos} = \sum_{i \in V'} ||\mathbf{v}_{p,i} - \mathbf{v}_{o,i}||_2$, where $\mathbf{v}_{p,i}$ and $\mathbf{v}_{o,i}$ are the positions of the i-th predicted vertex and original vertex, and V' be the set of visible vertex indices.

Laplacian Loss. This loss is not only applied to visible vertices but also head and hand vertices regardless of their visibility because noisy deformation of these vertices can significantly affect the perceptual result, and is defined as $L_{lap} = \sum_{i \in V} ||(\mathbf{v}'_{p,i} - \mathbf{v}'_{o,i}) \cdot u_i||_1$, where $\mathbf{v}'_{p,i}$ and $\mathbf{v}'_{o,i}$ are the Laplacian coordinates of the i-th predicted and original vertices, and u_i be the weight of the i-th vertex. We set $u_i = 100$ for head and hand, 1 for other visible vertices, and 0 for the rest.

Deformation Propagation Loss. To reconstruct an vertex invisible in the current frame, we find a visible counterpart in the set of keyframes computed in Sec. 3.3.2 and propagate the deformation from it. This is similar in spirit to the canonical shape model [158]. However, because the human in the source frame and target frame may have different poses, we can not simply force the deformation in the global coordinates to be similar. We adopt the local tangent space [118] (Fig. 3.10) to solve this problem.

Let $\mathbf{d}_i^{(s)}$ and $\mathbf{d}_i^{(t)}$ be the deformation in tangent space of the i-th source vertex visible in one of the selected keyframes and occluded target vertex at the current frame. The deformation loss is defined as $L_{deform} = \sum_{i \in V''} ||\mathbf{d}_i^{(t)} - \mathbf{d}_i^{(s)}||_1$, where V'' is the set of invisible vertex indices in

Type	Components
inconv	[Conv3×3 + ReLU + InstanceNorm]×2
down	[Conv3×3 + ReLU + InstanceNorm]×2 + MaxPool2×2
up	Upsample + [Conv3×3 + ReLU + InstanceNorm]×2
outconv	Conv1×1

Table 3.1: Network components. We use ReLU [154] for activation, and Instance Normalization [205] for normalization

Name	Type	Input	Output Channels
inc	inconv	RGB+SH lighting	64
down1	down	inc	128
down2	down	down1	256
down3	down	down2	512
down4	down	down3	512
up1a	up	down4, down3	256
up2a	up	up1a, down2	128
up3a	up	up2a, down1	64
up4a	up	up3a, inc	64
outca (Normal Output)	up	up4a	3
up1b	up	down4, down3	256
up2b	up	up1b, down2	128
up3b	up	up2b, down1	64
up4b	up	up3b, inc	64
outcb (Albedo Output)	outconv	up4b	3

Table 3.2: Network architecture of AlbeNorm CNN

target frame. $\mathbf{d}_i^{(s)}$ does not receive gradients, and is stored in a buffer updated when the source frame is sampled during training. In practice, we extend V'' to include head vertices, to enhance head rigidity. Our deformation propagation scheme provides more realistic details on invisible surfaces as shown in Fig. 3.11.

3.4 Implementation Details

In this section, we describe how our system is implemented, including details of the network and the human model.

Deep Networks. CNNs are based on U-Net [172], optimized using Adam [108]. The full-frame image resolution is 960×540 with the human region resolution around 200×430. Texture resolution is 512×512.

All the CNNs are U-Net sharing similar architectures, except for the VGG16 Network [192]

Name	Type	Input	Output Channels
inc	inconv	Coarse Texture	64
down1	down	inc	128
down2	down	down1	256
down3	down	down2	512
down4	down	down3	512
up1	up	down4, down3	256
up2	up	up1, down2	128
up3	up	up2, down1	64
up4	up	up3, inc	64
outc	outconv	up4	3

Table 3.3: Network architecture of TexGen CNN

Name	Type	Input	Output Channels
inc	inconv	feat0	64
down1	down	inc, feat1	128
down2	down	down1, feat2	256
down3	down	down2, feat3	512
down4	down	down3, feat4	512
up1	up	down4, down3	256
up2	up	up1, down2	128
up3	up	up2, down1	64
up4	up	up3, inc	64
outc	outconv	up4	3

Table 3.4: Network architecture of MeshRef CNN. “feat0”, “feat1”, “feat2”, “feat3”, “feat4” are features converted from VGGNet input, conv1_2, conv2_2, conv3_3, and conv4_3 features

in MeshRef module. See Tab. 3.1 for the shared components and Tab. 3.2, 3.3, and 3.4 for architectures of AlbeNorm, TexGen, and MeshRef CNNs. Specially, the CNN in TexGen predicts the residual between the coarse texture and the fine texture.

We use $K = 30$ for the number of selected frames, and $\lambda_a^{an} = 1, \lambda_n^{an} = 1, \lambda_{L1}^{tg} = 20, \lambda_{pct}^{tg} = 1, \lambda_{lap}^{tg} = 10, \lambda_{L1}^{mr1} = 1, \lambda_{ssim}^{mr1} = 1, \lambda_{lap}^{mr1} = 20, \lambda_{pct}^{mr2} = 1, \lambda_{sil}^{mr2} = 100, \lambda_{temp}^{mr2} = 10, \lambda_{pos}^{mr2} = 10, \lambda_{lap}^{mr2} = 10, \lambda_{deform}^{mr2} = 10$ for the loss weights. We use learning rate 10^{-5} for pretraining AlbeNorm, 10^{-4} for pretraining MeshRef, 3×10^{-4} for optimizing TexGen, and 5×10^{-5} for finetuning MeshRef. We use batch size 4 for pretraining AlbeNorm, 1 for pretraining MeshRef, 1 for optimizing TexGen, and 3 for finetuning MeshRef (as a triplet for motion smoothness loss). VGGNet is trained from scratch with MeshRef CNN, and kept fixed during finetuning. To speed up finetuning, we use a smaller image size 480×270 for photometric losses, but the image features are from the 960×540 original image.

Human Model. The full-body human model is a variant of SMPL [142]. The original SMPL model has about 7k vertices, which is too large for the coarse mesh representation, and insufficient to represent fine details such as clothing wrinkles. Thus, we construct a body model with two levels of resolution: the coarse level has 1831 vertices and 3658 faces, and the fine level has 29,290 vertices and 58,576 faces obtained by subdividing the coarse mesh topology. The vertices of the coarse mesh are a subset of the vertices of the fine mesh and share identical vertex indices. Both representations also share a unique skeletal rig that contains 74 joints. This design reduces the overhead for generating the coarse mesh, and preserves the fine mesh capability to represent geometric details.

3.5 Experimental Analysis

In this section, we provide information about the datasets and analyze the experimental results.

3.5.1 Datasets

Our method requires lighting information, which is not provided by most public datasets. Thus we capture and render our own data to perform experiments.

Synthetic Images for Pre-training. We synthesize 18,282 images using 428 human 3D scans from RenderPeople[1] and Our Dataset under the lighting from Laval Dataset [55]. Examples are in Fig. 3.12 and Fig. 3.13. Our Dataset was captured with a 3dMD scanner and contains 48 subjects. We registered the fine level Human Model to the 3D scans using non-rigid ICP [122], initialized with a 3D pose estimator [24]. To generate the coarse mesh, Gaussian noise scaled by a random factor sampled from a uniform distribution is added to the pose and shape parameters, and the position of the character. The registered model can be set in an arbitrary pose with a skeletal rig. We render the 3D scans into images of various poses sampled from the Motion Capure data. No video sequences are synthesized due to its high computational demand. Our final dataset contains coarse meshes, fine meshes, displacement maps, environment maps, RGB images, albedos, normals, and human masks.

[1] http://renderpeople.com/

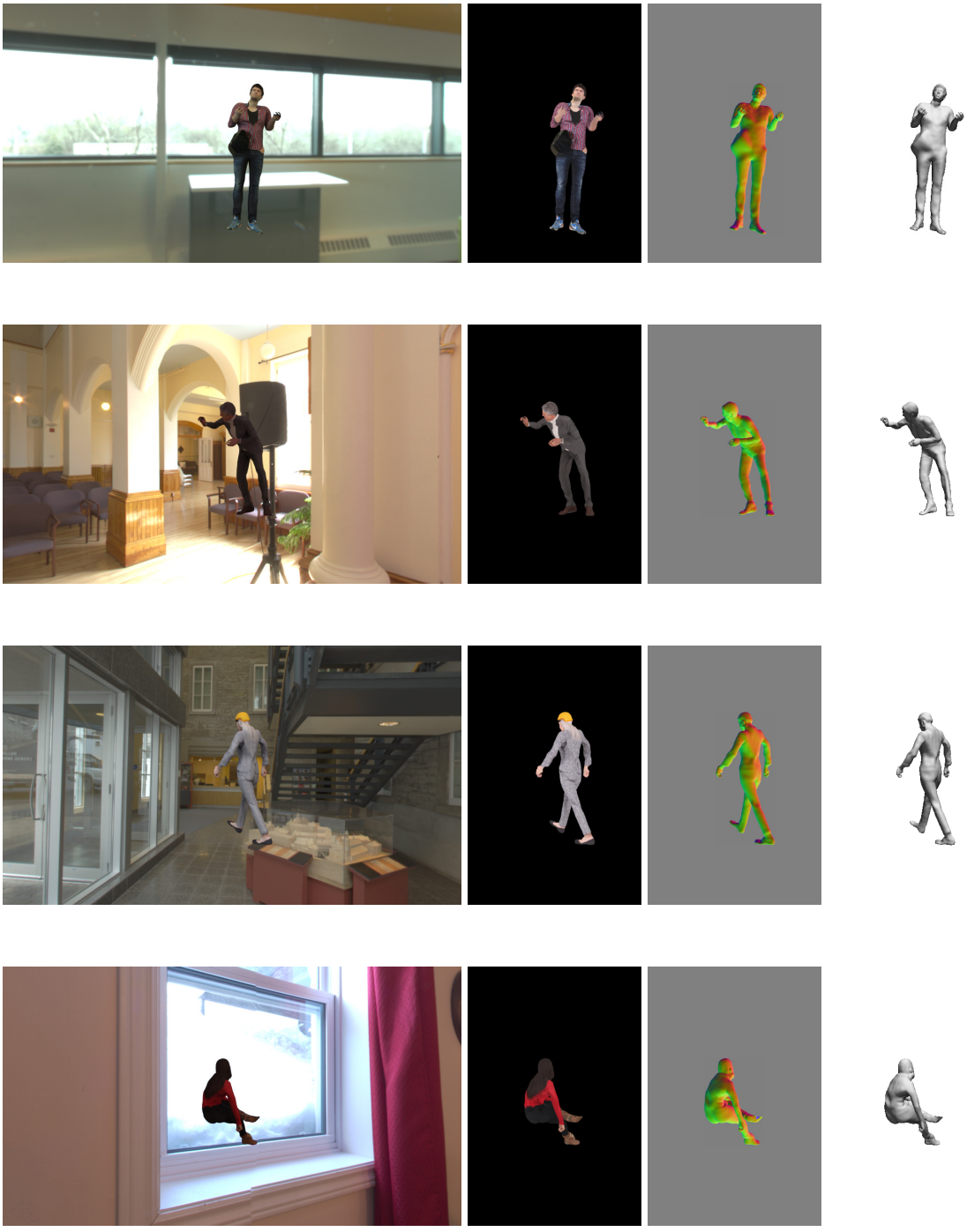


Figure 3.12: Example synthetic training data. The albedo and normal images are cropped to focus on human region.

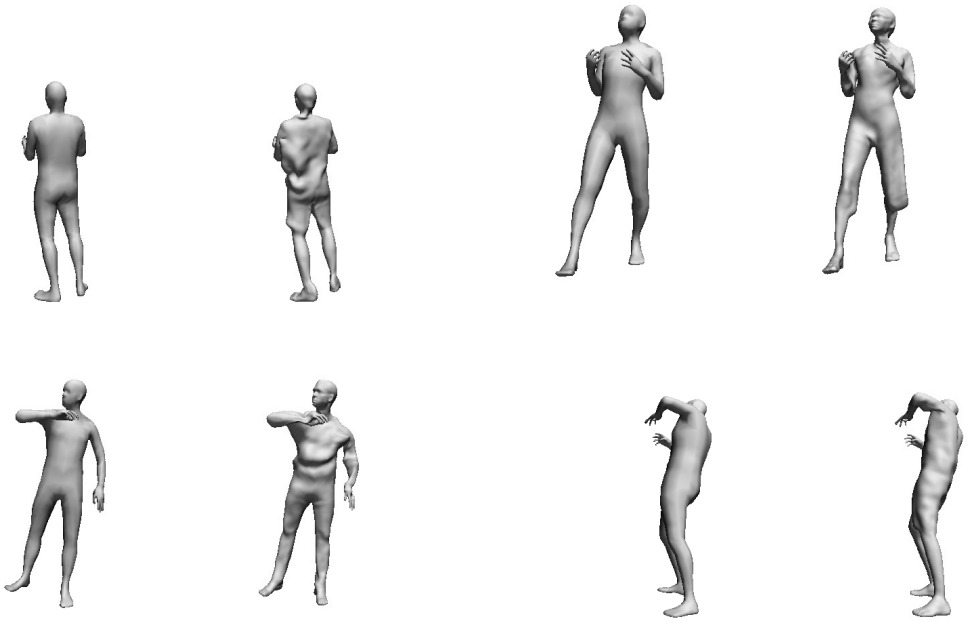


Figure 3.13: Example coarse and fine mesh pairs for pre-training MeshRef Module.

Synthetic Videos for Quantitative Evaluation. We synthesize 6 videos with ground truth measurements that contain dynamic clothing deformation for higher realism. Our clothing is modeled as separate meshes on top of human body scans as in DeepWrinkles [114]. However, we obtain deformation by physics-based simulation. We use the human bodies from AXYZ dataset [1] and the lighting from HDRI Heaven [4]. The videos represent subjects performing different motions such as walking or dancing and have about 3.8k frames each. In each video, we use about half of the frames for model adaptation and do inference on the rest. We treat the naked body as coarse mesh and the clothed body as fine mesh.

Real Videos for Qualitative Evaluation. We capture 8 videos ($\sim$4min each) using a Kinect along with lighting captured using a ThetaS. The cameras are geometrically and radiometrically calibrated. We use the first 2k frames for model adaption and infer on the whole video. We obtain the coarse mesh in real-time by solving an inverse kinematic problem to fit the posed body shape to the 3D point cloud and detected body keypoints [29] using an approach similar to [209].

3.5.2 Results

Texture. We compare our texture with a sampling-based method (SBM) [8] variant and a Textured Neural Avatars (TNA) [189] variant re-implemented with our Human Model, which map between image and UV spaces using the mesh. We render albedo images on synthetic videos, and evaluate average RMSE within valid mask and MS-SSIM [218] within human bounding box over subsampled videos. Our method outperforms SBM and TNA on (RMSE, MS-SSIM):

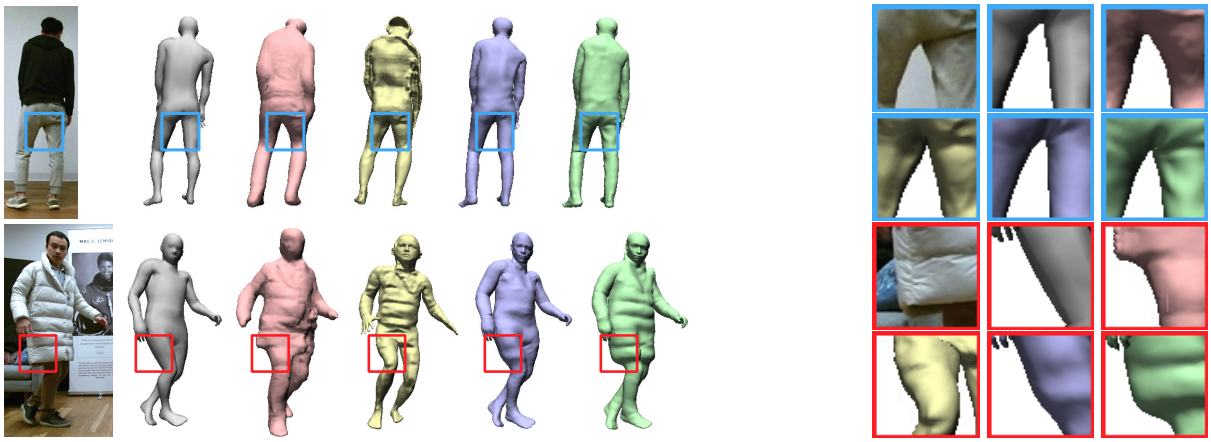


Figure 3.14: Comparing mesh with coarse mesh (grey), DeepHuman [244] (red), HMD [249] (yellow), Tex2Shape [10]. Our method (green) outperforms them in shape and details: DeepHuman is coarse in head and hands. HMD has artifacts in head and body regions. Tex2Shape does not obtain realistic wrinkles.

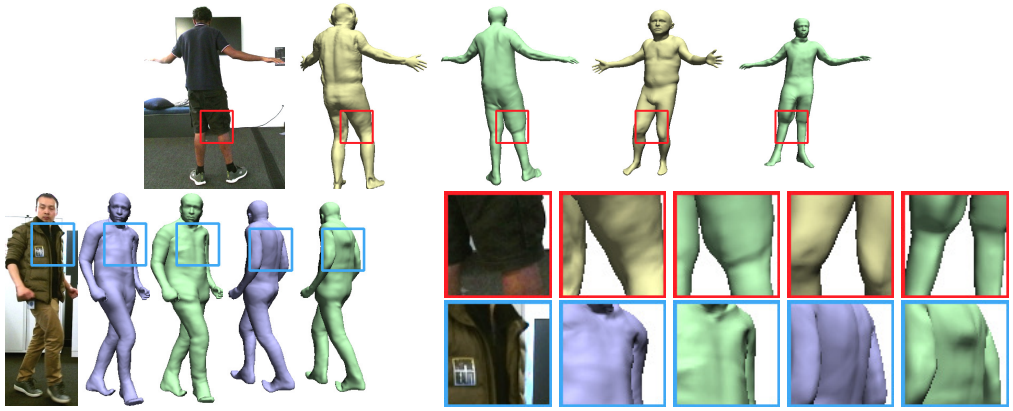


Figure 3.15: Viewing from front and back. Tex2Shape (blue), HMD (yellow), Ours (green). Ours captures the shape of occluded jacket hood and shorts deformation.

(0.124, 0.800) for SBM, (0.146, 0.809) for TNA, and (**0.119, 0.831**) for ours, respectively. See Fig. 3.8 qualitative results.

Mesh. We compare our method with DeepHuman [244], HMD [249], and a variant of Tex2Shape [10] trained on our synthetic images for our Human Model, predicting 3D displacements on top of the posed coarse mesh, using our network and loss settings. For fairness, we compare with both the original version and the variants of DeepHuman and HMD where the initial mesh is replaced by our coarse mesh. In Fig. 3.14, while DeepHuman and HMD provide unrealistic heads and Tex2Shape fails to produce faithful clothing details, our method is shape-preserving and generates better fine geometry. We also recover the geometry of the shorts and jacket in the occluded regions (Fig. 3.15). Quantitatively, we evaluate on the synthetic videos by rasterizing 2D normal images. The metrics are silhouette IoU, RMSE, and MS-SSIM[218] within human mask/bounding box. Tab. 3.5 shows that our method outperforms the others on all metrics.

Method	IoU	RMSE	MS-SSIM
DeepHuman	0.650	0.399	0.421
DeepHuman variant	0.779	0.309	0.587
HMD	0.667	0.417	0.684
HMD variant	0.790	0.344	0.779
Tex2Shape variant	0.926	0.192	0.857
Ours (no fine-tuning on video)	0.940	0.186	0.857
Ours (replace input normal by RGB)	0.928	0.190	0.852
Ours (no VGG feature)	0.932	0.185	0.865
Ours (no deformation propagation)	**0.941**	0.174	0.869
Our full method	**0.941**	**0.173**	**0.870**

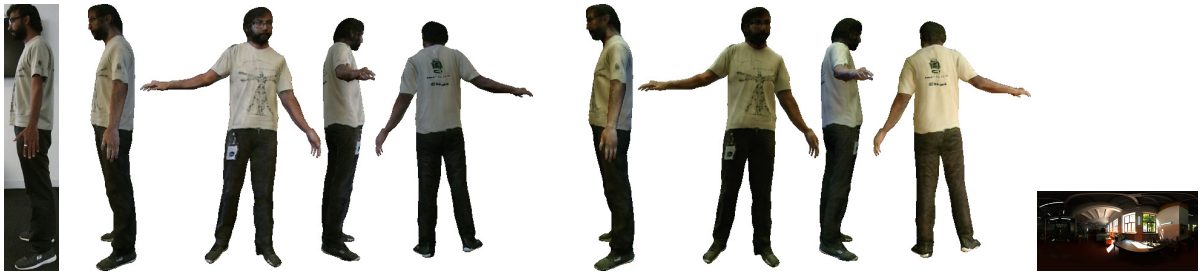
Table 3.5: Evaluation of mesh reconstruction. Silhouette IoU, rasterized normal RMSE and MS-SSIM are listed. Our method significantly outperforms the compared methods. The ablation study shows the key designs are crucial

Figure 3.16: Rendering results. The right-most scene is from synthetic video. The rendering has both high fidelity and perceptual quality, from different viewpoints. The clothing wrinkles, logo, and text are clearly recovered.

Ablation Study. We quantify the effect of domain finetuning, replacing the normal image by RGB image in MeshRef, removing the VGGNet, and removing the deformation propagation scheme in Tab. 3.5. Evidently, the first three components are crucial and ignoring them hurts the performance. As expected, removing deformation propagation has little effect because it focuses mainly on the occluded regions (see Fig. 3.11 for its qualitative effect).
Applications. We show rendering from novel viewpoints (Fig. 3.16) and relighting in a different environment (Fig. 3.17) using our outputs. The results have clear textures and realistic shading variation around clothing wrinkles.

3.6 Limitations

One key limitation of our method is that we assume known lighting. However, lighting is often unavailable in real applications. Besides, we rely on a spherical camera to capture the lighting and represent it using only low frequency Spherical Harmonics. Exploring recent techniques using a single narrow FOV image to estimate high frequency environmental lighting [55, 178]



Rendering (original lighting) Relighting

Figure 3.17: Free-viewpoint rendering and relighting. The detailed pattern on the shirt is clearly reconstructed. The shading varies in the clothing wrinkles implying that the wrinkles are correctly estimated as geometry rather than texture.

together with subtle lighting cues from human appearance [233] is an exciting future direction to tackle this problem. In addition, because we model clothing details as vertex deformation, topology changes cannot be well represented. Modeling clothing using separate meshes [236] or implicit functions [174] might be a possible way to solve this problem.

3.7 Conclusion

In summary, as an example of using photometric supervision from material-aware appearance models, we introduce a deep learning method for recovering texture and geometry of a dynamic human from an RGB-D video. By assuming diffuse materials, we build the mathematical relationship between albedo, geometry, and image as supervision signal. We significantly improve the reconstruction quality by mitigating the domain shift problem via adapting on a short real clip and incorporating human priors and temporal priors.

One possible future direction is to train the model with photometric supervision on many real videos concurrently to learn a generic shape prior, as it could allow faster adaptation to new sequences or even require no adaptation at all.

Chapter 4

Adversarial Supervision from Appearance Location

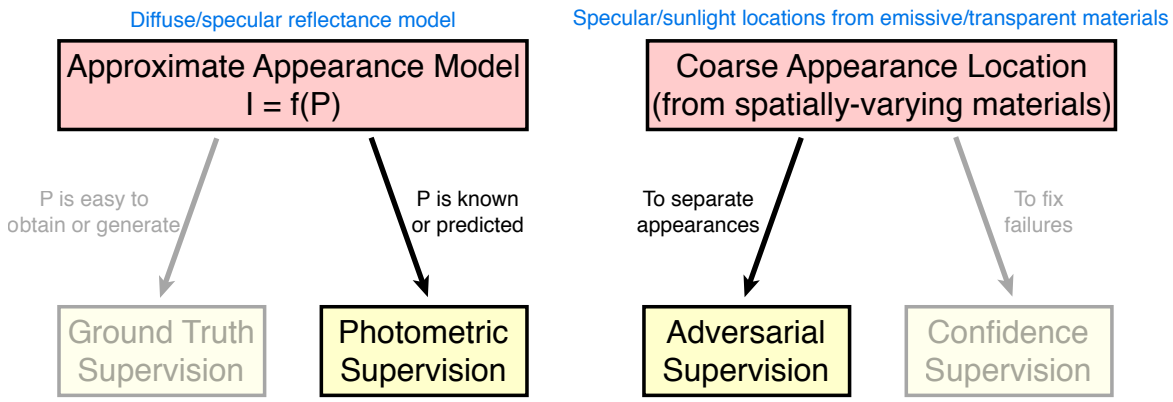
In previous chapters, we have explored supervision signals from material-aware appearance models. As mentioned in Chapter 1, appearance location information from spatially varying materials also provides signals that can be used for supervising the training of deep networks. A common target is to separate or remove special appearances. In this chapter, we introduce an approach based on adversarial supervision to separate specular reflection and direct sunlight components from floor appearances, using appearance location information from the awareness of emissive and transparent materials. Here, the appearance location information can be seen as a coarse version of the prediction target, which is a typical application scenario of adversarial supervision.

Specially, we describe a new appearance decomposition method for diffuse-specular separation and direct sunlight detection on the planar floor regions from 360° panoramic images. Our system is weakly supervised, assuming known room layouts and material semantics that can be either automatically inferred or quickly provided through human annotation. It uses a GAN-based approach to extract specular reflection or direct sunlight, and enhances the resolution of the decomposition result using a guided-filtering-like technique. Our system is capable of photo-realistic virtual furniture insertion, which is important for AR applications such as gaming and virtual staging in real estate. We show the results on a large dataset of real captured panoramas of empty homes to validate our method.

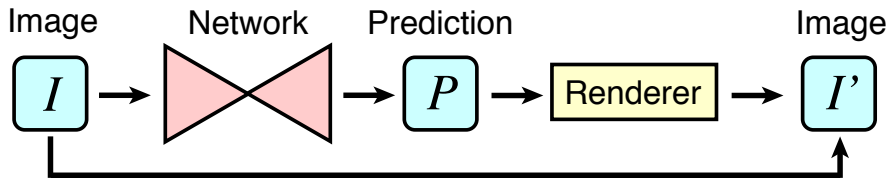
4.1 Application: Floor Appearance Decomposition for Object Insertion

Remote home shopping is becoming more popular, and effective tools to facilitate virtual home tours are much needed. Examples of such tools include virtual staging and interior design: how would furniture fit in a home of interest, and what would the home look like with a redesign?

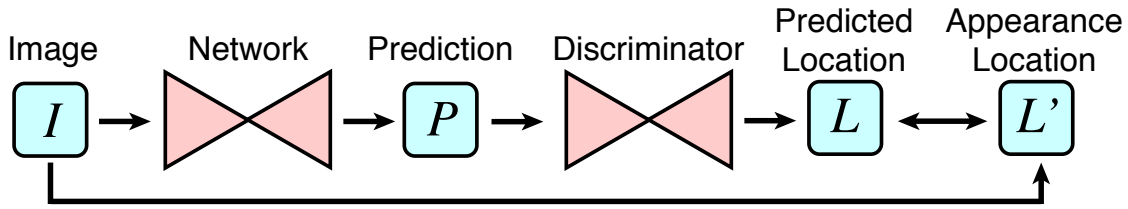
Inserting virtual furniture into images of rooms in a photo-realistic manner is nontrivial. Multiple complex shading effects should be taken into consideration. Such complex interactions are shown in Fig. 4.2; the insertion induces effects such as occlusion, sunlight cast on objects, and soft and hard shadows. Hence, the furniture insertion task requires not only the estimation of the



(a) Framework



(b) Photometric Supervision



(c) Adversarial Supervision

Figure 4.1: The floor decomposition task uses appearance model and appearance location information from material awareness for photometric and adversarial supervision. Using the coarse masks of specular and sunlight regions obtained from room layout and material semantics as supervision signals, we accurately separate the specular and sunlight components.

Figure 4.2: Room furnishing example (top: empty, bottom: furnished). Multiple realistic effects are rendered, including the shadows under the table, sunlight on the sofa and table, and occlusion of specular reflection by the small plant and chair. The tall plant creates soft and hard shadows by blocking the diffuse skylight and the directional sunlight.

indoor environment map that can vary spatially to adapt to the geometry of the scene [56, 130], but also the separation of diffuse and specular components, estimation of sun direction, and detection of direct sunlight. This highly ill-posed problem requires some form of prior to resolve the ambiguity.

This task is challenging due to the lack of ground truth training data. The ground truth annotations for appearance decomposition tasks either require significant human labor [19], or specialized devices in a controlled environment [64, 182], which is hard to be extended to a large scale. Previous approaches [51, 130, 132] rely on synthetic data for supervised training. However, there is the issue of domain shift and the costs of designing and rendering scenes [131].

By contrast, it is easier to annotate the ground truth layout geometry and semantics, in the spirit of the early human-in-the-loop approaches by [95, 237], which facilitates the advancement in layout estimation [167, 197, 250] and semantic segmentation [34, 141, 212] algorithms. We also capitalize on the existence of such data; we propose an approach for weakly-supervised appearance decomposition, assuming known layout geometry, in the form of the 3D position and segments of floor, ceiling and walls, and semantic labels for windows, doors and lamps. We demonstrate its effectiveness on diffuse-specular separation and direct sunlight removal for the planar floor regions.

Our approach consists of three steps: (1) approximate region proposal for specular reflections and direct sunlight using the input layout geometry and semantics as a coarse supervision signal, (2) a novel GAN-based method to extract specular reflections and direct sunlight, and (3) the use of the high-res input RGB image as a guidance to enhance the low-res decomposition results. In addition, useful side information (*e.g.* sun direction) can also be estimated.

To evaluate our system, we experiment on a large dataset of 591 unfurnished houses with various floor materials (wood, carpet, concrete, tile, etc.). Our system is able to effectively decompose the floor appearance into multiple components. It enables photo-realistic object insertion

applications (Fig. 4.24) by combining blocked specular and sunlight effects rendered separately. We validate our method and justify our design decisions through qualitative and quantitative analyses on specular reflection removal, sun direction estimation, and direct sunlight detection.

4.2 Related Work

Inverse Rendering: The goal of inverse rendering is to estimate various physical attributes of a scene (e.g., geometry, material properties and illumination) given one or more images. Intrinsic image decomposition estimates reflectance and shading layers [199]. Other methods attempt to recover scene attributes with simplified assumptions: single object [15, 104], faces [177, 188], texture-like materials [5], near-planar surfaces [129], or general scenes with human-in-the-loop assistance [95, 237]. Data-driven methods require large amounts of annotated data, usually synthetic images [130]. The domain gap can be reduced by fine-tuning on real images using self-supervised training via differentiable rendering [176]. In our work, we model complex illumination effects on real 360° panoramas of empty homes. Similar to [95], we believe that layouts and semantic segmentation are easier to collect and train good models for. By contrast, diffuse and specular annotations are harder continuous signals, coupled in a nontrivial, spatially-varying way.

Illumination Estimation: Many approaches represent indoor lighting using HDR maps (or its spherical harmonics). Some estimate lighting from an LDR panorama [49, 61], a perspective image [55, 193] or object appearance [60, 161, 219]. Recent approaches [58, 130] extend this representation to multiple positions, enabling spatially-varying estimation. Others [56, 89, 96] estimate parametric lighting by modeling the position, shape, and intensity of light sources. Zhang *et al*. [238] combine both representations and estimate a HDR map together with parametric light sources. However, windows are treated as the source of diffuse skylight without considering directional sunlight. We handle the spatially-varying high-frequency sun illumination effects, which is usually a challenging case for most methods.

Some techniques estimate outdoor lighting from outdoor images. Early methods [115, 116] use analytical models to describe the sun and sky. Recently, deep learning methods [74, 75, 240] regress the sun/sky model parameters or outdoor HDR maps by training on large-scale datasets. However, they use outdoor images as input, where the occlusion of the sunlight by interior walls is not relevant.

Specular Reflection Removal: There are two main classes of specular reflection removal techniques. One removes specular highlights on objects. Non-learning based approaches usually exploit appearance or statistical priors to separate specular reflection, including chromaticity-based models [6, 180, 198, 231], low-rank model [65], and dark channel prior [103]. Recently, data-driven methods [185, 225] train deep networks in a supervised manner. However, the reflection on floors is more complex than highlights, because it may reflect window textures and occupy a large region.

The second class removes reflections from a glass surface in front of the scene. Classical methods use image priors to solve this ill-posed problem, including gradient sparsity [11, 119], smoothness priors [124, 210], and ghosting cues [186]. Recently, deep learning has been used for this task [51, 121, 211, 220, 242] and achieved significant improvements by carefully designing

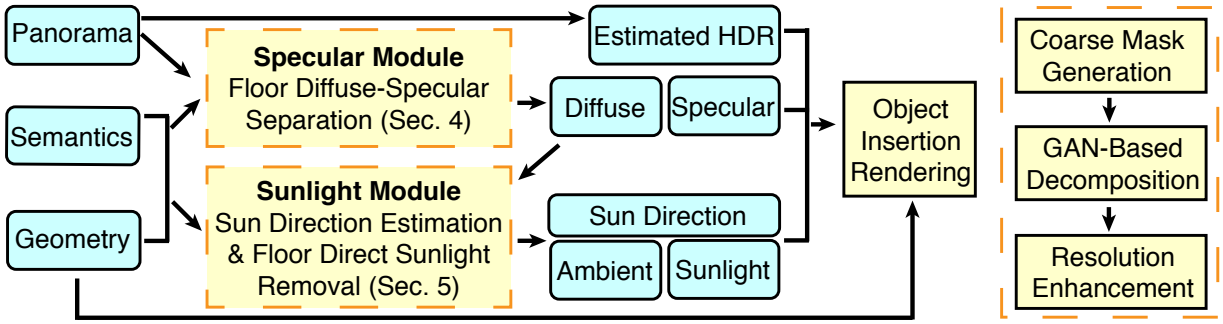


Figure 4.3: Our system has two main modules: Specular module for diffuse-specular separation and Sunlight module for sun direction estimation and sunlight removal. They share a similar weakly-supervised pipeline shown on the right. Their outputs, together with an estimated HDR environment map, are used for rendering realistic object insertion results.

network architectures. However, they mostly use supervised training, requiring large amounts of data with ground truth.

	Specular Module	Sunlight Module
Train	Single panorama	Multiple panoramas on the same floor
	Semantic segmentation	Semantic segmentation
	Layout geometry and camera pose	Layout geometry and camera pose
Test	Single panorama	Multiple panoramas on the same floor
	Floor mask	Semantic segmentation
		Layout geometry and camera pose

Table 4.1: Inputs required for different modules and stages.

4.3 System Overview

Our system for object insertion is shown in Fig. 4.3. The input is one or more panoramic image(s) with known semantics (segments of windows, lamps, floor, ceiling, walls) and layout geometry (3D positions of floor, ceiling and walls, camera pose). The system consists of two main modules: Floor Diffuse-Specular Separation (Specular module) to handle specularities, and Sun Direction Estimation and Floor Direct Sunlight Removal (Sunlight module) for handling sunlight effects. Their outputs, together with an estimated HDR map, are used to simulate complex object-scene effects including soft and hard shadows, blocked specular reflection and sunlight, and sunlight cast onto objects.

The Specular module and Sunlight module share a similar 3-step weakly-supervised pipeline as shown at the right side in Fig. 4.3: (1) coarse specular or sunlight mask generation using semantics and geometry, (2) GAN-based method for specular separation and sunlight estimation, and (3) guided spatial resolution enhancement method. In our current work, we consider the specular reflection and sunlight *on the floor only*, which we assume to be a single plane.

Figure 4.4: 3D visualization of floor layouts. The textures come from projecting panoramas to perspective views. Black artifacts around the centre of each room are tripods.

Split	Houses	Floors	Primary Panoramas	All Panoramas
Train	467	815	10,243	23,056
Test	124	214	2,785	6,902
Total	591	1,029	13,028	29,958

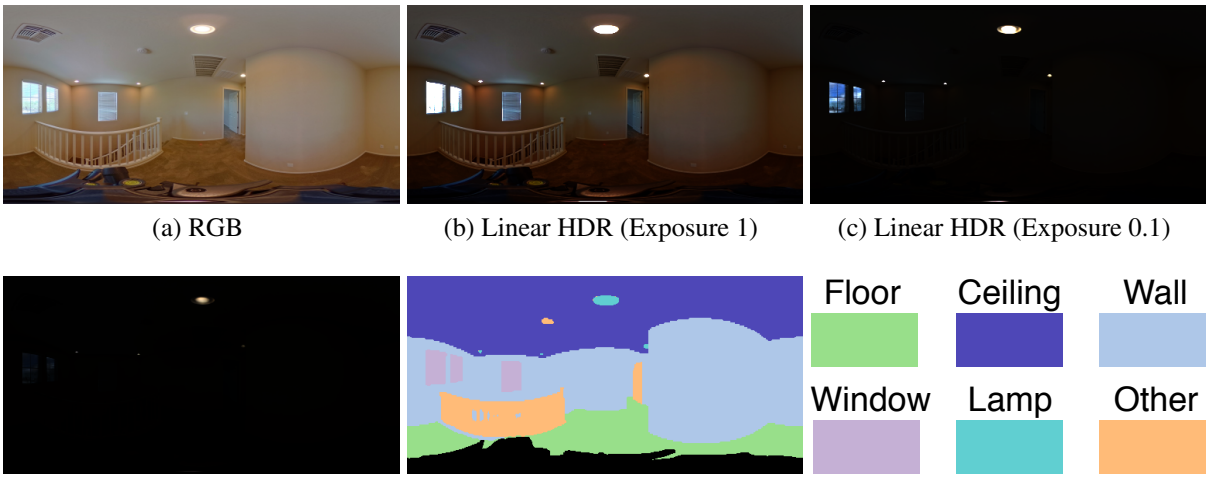
Table 4.2: Dataset statistics and train-test split. Primary panoramas are panoramas with camera pose.

4.4 Dataset Details and Preprocessing

To validate our method, we use a subset of the Zillow Indoor Dataset (ZInD) [43], including 591 unfurnished houses captured using 360° cameras under day-light with indoor lights turned on, where each home consists of $1\sim4$ floors. Around $20\sim30$ 360° panoramic images are captured for each floor. The split of training and test sets is listed in Tab. 4.2. The panoramas are tone-mapped, from which we estimate HDR maps. The dataset provides floor plan annotation with approximate ceiling height and the camera poses of around 40% panoramas (called "primary" panoramas). Given the layout and primary panoramas, we are then able to generate textured meshes for the houses (Fig. 4.4). To obtain semantic segmentation, we adapt a pre-trained model [212] to panoramic views. For Specular module, we use a single panoramic image. For Sunlight module, we use the images of different rooms on the same floor, assuming they share the same sun direction. Below we describe the generation of HDR maps and semantic segmentation.

4.4.1 HDR Map Estimation

We train a deep network for estimating the HDR map from a tonemapped image (inverse tonemapping). Concretely, given a tonemapped image, we first "linearize" it assuming a 2.2 sRGB gamma to obtain a "linearized" LDR image I_{ldr}. To obtain a HDR image I_{hdr}, we use a deep network to predict the residue in log space $I_{res} = ln(1 + I_{hdr} - I_{ldr})$. This network is trained in a supervised



(a) RGB (b) Linear HDR (Exposure 1) (c) Linear HDR (Exposure 0.1)

(d) Linear HDR (Exposure 0.01) (e) Semantic Segmentation (f) Legends

Figure 4.5: Dataset preprocessing result. (b) (c) (d) show the estimated HDR map with different exposure levels. (e) is the semantic segmentation result, where the tripod is masked as black pixels.

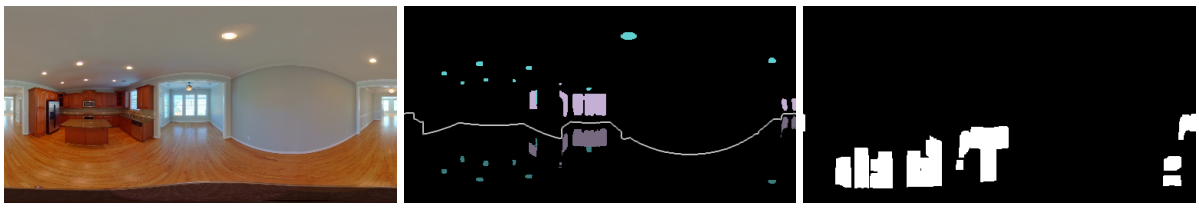
manner using Laval Indoor Dataset [55] and HDRI Heaven dataset [4]. The loss is a L2 loss weighted by the cosine of the altitude angle of each pixel.

To prevent artifacts during inference, this inverse tonemapping operation only applies to bright regions, which are usually saturated. We define a soft bright mask $M = max(0, tanh(I_{hdr} - 1))$ and obtain the final HDR map via alpha blending $I_{final} = M \cdot I_{hdr} + (1 - M) \cdot I_{ldr}$. See Fig. 4.5 for an example HDR map visualized with different exposures.

4.4.2 Semantic Segmentation

To obtain semantic segmentation of a panoramic image, we use a perspective-to-panoramic transfer learning technique. Specifically, we obtain a HRNet model [212] pre-trained on a perspective image dataset ADE20K [245] and treat it as the Teacher Model. Then we use the same model and weights to initialize the Student Model, and adapt it for panoramic image segmentation. To supervise the Student Model, we sample perspective views from the panoramic image. Let I_{pano} be the original panoramic image, Φ be the sampling operator, f_{tea} be the Teacher Model function, and f_{stu} be the Student Model function. The transfer learning loss L_{trans} is defined as the cross entropy loss between $f_{tea}(\Phi(I_{pano}))$ and $\Phi(f_{stu}(I_{pano}))$. To regularize the training, we prevent the Student Model from deviating from the Teacher Model too much by adding a term L_{reg} defined as the cross entropy loss between $f_{tea}(I_{pano})$ and $f_{stu}(I_{pano})$. The total loss is $L = w_{trans}L_{trans} + w_{reg}L_{reg}$. w_{trans} and w_{reg} are weights given by the confidence of the Teacher Model prediction $f_{tea}(\Phi(I_{pano}))$ and $f_{tea}(I_{pano})$.

In our experiments, we define 7 classes: floor, ceiling, wall, window, lamp, door, and other. In Fig. 4.5 we merge "door" into "other" because we do not use door segmentation in this paper. Because ADE20K has a different set of label definitions, we manually define the mapping from ADE20K labels to our labels. Specifically, the ADE20K definition of lamps usually includes



(a) Original Image (b) Lighting Reflection (c) Coarse Mask

Figure 4.6: Coarse specular reflection mask generation. As in (b), the specular reflection area on a mirror floor is obtained by reflecting the light source (window and lamp) segments. The coarse specular area (c) is generated by further jittering the lamp height and dilating the mask.

the whole lamp, while we only care about the bright bulb. Thus, we only treat the bright parts with $I_{hdr} > 2$ as lamps, while defining the rest parts as "other". To make the prediction result consistent with the layout geometry, we apply several rules to improve the segmentation result, including: semantic windows cannot be on the layout floor, the bottom half of the image cannot be the semantic ceiling, and the semantic ceiling cannot be on the layout floor. When these conflicts happen, we trust the layout geometry. The tripod region is segmented by training HRNet on 462 images from the training set where we create tripod annotations on. We ignore the tripod regions in our experiments.

4.5 Floor Diffuse-Specular Separation

To separate diffuse and specular components from a floor image without ground truth supervision, we propose to first obtain coarse reflection masks from room semantics and layouts (Sec. 4.5.1), and then use a GAN-based approach to learn to decouple the spatially-varying diffuse and specular signals (Sec. 4.5.2) in a weakly-supervised manner. These two steps handle low resolution panoramas (256×512) due to the limited computing resources. We further present a method to enhance the resolution of the separation result with the guidance of the original high-resolution RGB image (Sec. 4.5.3). This 3-step process is designed to solve this highly ill-posed problem. Despite possible specular residues in the intermediate diffuse image, the final result is of high perceptual quality.

4.5.1 Coarse Specular Mask Generation

The key to learning without ground truth is to find appropriate supervision signals that are available "for free" from the data generation pipeline, or we can cast our problem as another simpler task. In our case, we believe that layout, and windows and lamps segmentation are easier, discrete signals for which we can collect human annotations and train good models. However, diffuse and specular annotations are harder continuous signals, since they are coupled in a non-trivial, spatially-varying way.

Our key observation is simple, yet effective, that *the possible specular area, on the floor plane, is the mirror reflection of the light sources*, in a similar spirit to [237]. Here, we define light sources as emissive materials like indoor lamps, and transparent materials like windows that

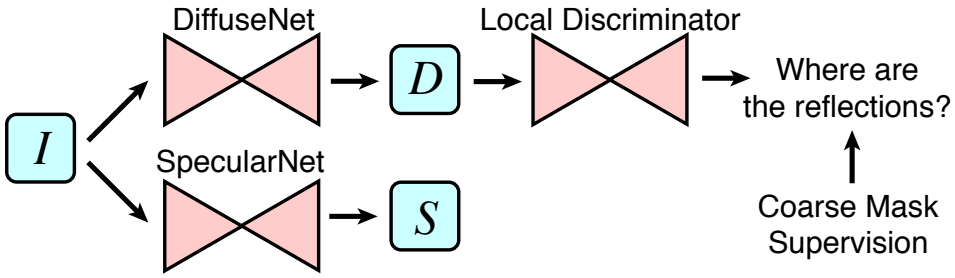


Figure 4.7: Architecture for diffuse-specular separation. Two networks predict diffuse and specular components separately. A local discriminator finds the reflections by looking at the diffuse component, while DiffuseNet tries to fool it.

allow outdoor lighting to illuminate the room. In other words, we propose to find coarse masks of specular reflections using *the input layout geometry and material semantics* as a supervision signal.

Specifically, the input is a single RGB panorama with known room layout (3D positions and segments of floor, ceiling, and walls), camera pose (inferred from the layout), and semantic segmentation (windows and lamps), while the output is a coarse mask of specular regions (Fig. 4.6 (c)). As shown in Fig. 4.6 (b), we can *automatically* locate both indoor and outdoor light sources by *coarse segmentation* of lamps and windows using deep learning [212]. If the floor was a mirror, the reflection area could be obtained by calculating the mirror reflection of light sources on the floor.

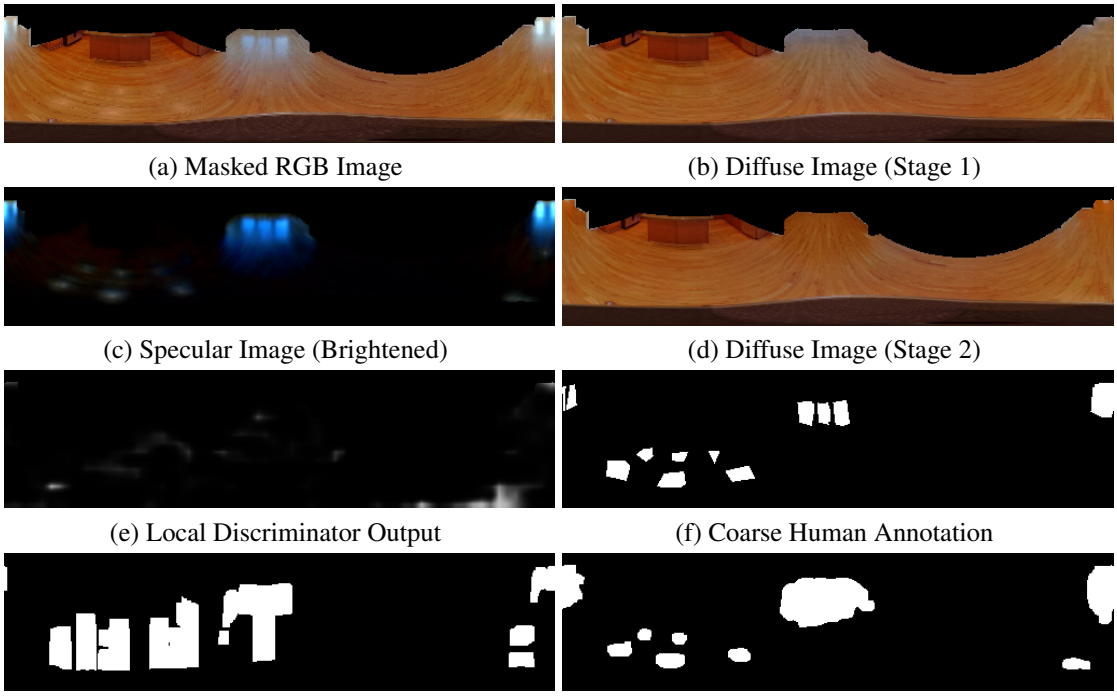
Since the only *known* geometry information is on the floor, ceiling, and walls, in order to project the lamp to the floor, we assume the lamp is lower than the ceiling and jitter its height to handle the depth uncertainty. Specifically, we sample 30 possible heights within an 1-meter range. Furthermore, rough floor materials have a larger blurred reflection area than the mirror material. Thus, as in Fig. 4.6 (c), we further dilate the mask for a better coverage. We set the dilation range to be 3 pixels for windows and 6 pixels for lamps.

The mask may include areas that are not specular at all (*e.g.* carpet) or ignore reflections caused by occluded light sources. We show how to use this coarse supervision to accurately separate specular reflection in Sec. 4.5.2.

4.5.2 GAN-Based Diffuse-Specular Separation

The coarse specular mask provides possible reflection areas, but is usually not very accurate (Fig. 4.8 (g)). To spatially refine the specular component, we introduce a simple yet effective prior: *After removing specular reflection, an algorithm, trained to detect specular reflections, should not be able to find the reflection areas in the image.*

We propose a GAN-based method to implement this idea. As shown in Fig. 4.7, our goal is to separate an image I (masked with floor region) into diffuse component D and specular component S. Two deep networks (DiffuseNet and SpecularNet) are used to predict D and S, respectively. See Fig. 4.8 for examples of I, D, and S. During training, we require the coarse specular mask from geometry and semantics as supervision signal; during inference, the floor region of the RGB panorama is the only input required.



(a) Masked RGB Image (b) Diffuse Image (Stage 1)

(c) Specular Image (Brightened) (d) Diffuse Image (Stage 2)

(e) Local Discriminator Output (f) Coarse Human Annotation

(g) Coarse Specular Mask (h) Fine Specular Mask

Figure 4.8: Intermediate results of diffuse-specular separation. Two networks take a masked image (a) and predict the diffuse (b)/(d) and specular (c) components separately. A discriminator tries to find reflection areas (e) by looking at (b)/(d). In Stage 1, the discriminator is supervised by the coarse mask (g). In Stage 2, we obtain the fine mask (h) by binarizing (c) for supervision. The Stage 2 result (d) removes the specular residue in the Stage 1 result (b). The color tints of different light sources are also removed in (d).

We train the networks in two stages. The first stage estimates the specular component by training DiffuseNet and SpecularNet together using the coarse mask. The second stage freezes SpecularNet and trains DiffuseNet using a fine mask that is generated by binarizing the estimated specular component. The second stage is for achieving a more realistic diffuse image with a stronger supervision signal.

Stage 1: Specular Image Estimation

Algorithm 3 Losses for Specular Image Estimation

Input: Original Image I, Floor Mask F, Coarse Reflection Mask M (Fig. 4.7 (e))
Output: Loss for DiffuseNet and SpecularNet L_g, Loss for LocalDiscriminator L_d

$I' \leftarrow I \cdot F$ # Masked image, Fig. 4.7 (a)
$D \leftarrow$ DiffuseNet(I') $\cdot F$ # Diffuse image, Fig. 4.7 (b)
$S \leftarrow$ SpecularNet(I') $\cdot F$ # Specular image, Fig. 4.7 (c)
$P \leftarrow$ LocalDiscriminator(D)$\cdot F$
Tries to find specular regions, Fig. 4.7 (d)
$L_{recon} \leftarrow L1Loss(D + S, I)$, in region F
$L_{diff} \leftarrow L1Loss(D, I)$, in region $(1 - M) \cdot F$
$L_{adv} \leftarrow CrossEntropy(P, 1 - M)$, in region M
$L_g \leftarrow \lambda_r L_{recon} + \lambda_d L_{diff} + \lambda_a L_{adv}$
$L_d \leftarrow CrossEntropy(P, M)$, in region F

In Stage 1, the loss L_g for training DiffuseNet and SpecularNet consists of a reconstruction loss L_{recon}, a diffuse region loss L_{diff}, and an adversarial loss L_{adv}:

$$L_g = \lambda_r L_{recon} + \lambda_d L_{diff} + \lambda_a L_{adv}, \tag{4.1}$$

where λ_r, λ_d, and λ_a are constants selected by inspecting training set results. The loss is applied to the floor region. We omit the floor mask below for convenience. See Algorithm 3 for pseudo code considering the floor mask. With $Loss_1$ below we indicate L1 Loss.

Reconstruction Loss: Assuming I is radiometrically calibrated, the image can be represented as $I = D+S$. Thus, a reconstruction loss can be naturally constructed as $L_{recon} = Loss_1(D+S, I)$. In practice, we found that this loss term also works for uncalibrated images.

Diffuse Loss: Let M be the coarse specular mask generated in Sec. 4.5.1. Then D should be identical to I for regions outside M. This is implemented as $L_{diff} = Loss_1(D, I)$ applied to pixels not covered by M.

Adversarial Loss: To ensure that the reflections are removed, we use a local discriminator network to find reflection areas by observing D. Let P be the output of the discriminator (Fig. 4.8 (e)). The discriminator is supervised by M using cross entropy loss function $L_d = Loss_c(P, M)$. To fool the discriminator, an adversarial loss $L_{adv} = Loss_c(P, 1 - M)$ is applied to region M.

For object insertion, the quality requirement of diffuse image is higher than the specular image because when objects occlude the specular reflection, the dffuse component is the one left

(a) Original Image (b) Specular Reflection (c) Diffuse Image

Figure 4.9: Diffuse-specular separation results. Specular reflections are 2× brightened. Our method handles both strong and weak reflections, from windows and lamps.

on the floor. As in Fig. 4.8, the specular image (c) is visually acceptable while the Stage 1 diffuse image (b) still has observable specular residues. This issue is handled in Sec. 4.5.2 by using a stronger supervision signal.

Stage 2: Diffuse Image Recovery

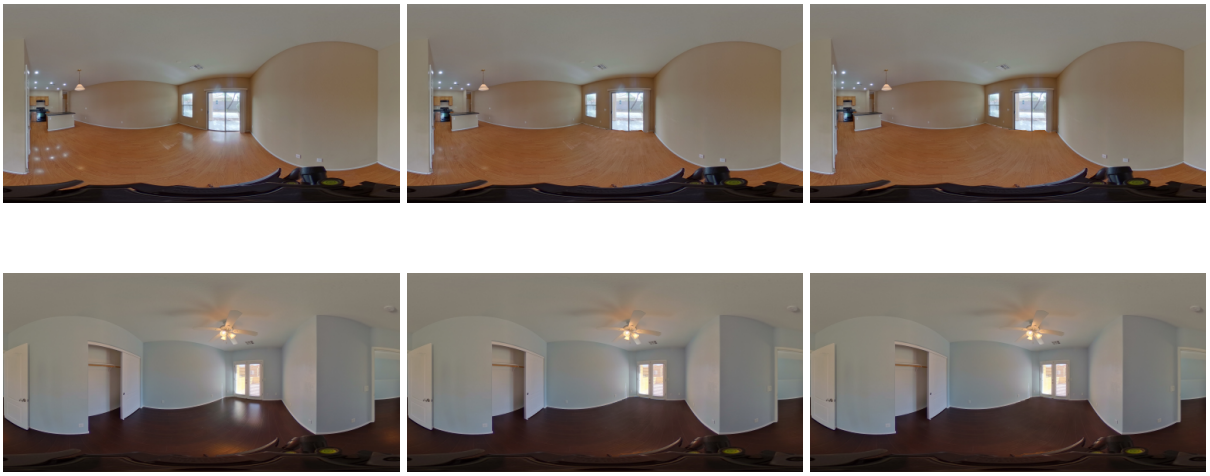
Algorithm 4 Losses for Diffuse Image Recovery (without on-the-fly data synthesis)

Input: Original Image I, Floor Mask F

Output: Loss for DiffuseNet L_g, Loss for LocalDiscriminator L_d

$I' \leftarrow I \cdot F$
$D \leftarrow$ DiffuseNet$(I') \cdot F$
$S \leftarrow$ SpecularNet$(I') \cdot F$
$P \leftarrow$ LocalDiscriminator$(D) \cdot F$
$M' \leftarrow$ Binarized S by thresholding intensity
$M' \leftarrow$ Dilate(M') # Fine mask, Fig. 4.7 (f)
$L_{recon} \leftarrow L1Loss(D + S \cdot M', I)$, in region F
$L_{diff} \leftarrow PerceptualLoss(D + S \cdot M', I)$, in region $(1 - M') \cdot F$
$L_{excl} \leftarrow ExclusionLoss(D, S)$, in region M'
$L_{adv} \leftarrow CrossEntropy(P, 1 - M')$, in region M'
$L_g \leftarrow \lambda_r L_{recon} + \lambda_d L_{diff} + \lambda_e L_{excl} + \lambda_a L_{adv}$
$L_d \leftarrow CrossEntropy(P, M')$, in region F

To improve the quality of the diffuse component, a better supervision signal is required for a stronger discriminator. Since Stage 1 provides a specular component with a better shape, from which a fine specular mask M' (Fig. 4.8 (h)) can be extracted by thresholding and dilating the specular image S. The intensity threshold is 0.05 and the dilation range is 2 pixels. In this stage, the DiffuseNet is trained with M', while SpecularNet from Stage 1 is frozen.



(a) Original Image (b) without On-the-Fly Synthesis (c) with On-the Fly Synthesis

Figure 4.10: On-the-fly data synthesis removes specular residues. See the lamp reflections in Row 1 and the window reflection in Row 2.

The loss consists of reconstruction loss L_{recon}, diffuse region loss L_{diff}, adversarial loss L_{adv}, exclusion loss L_{excl}, and on-the-fly data synthesis loss L_{syn}:

$$L_g = \lambda_r L_{recon} + \lambda_d L_{diff} + \lambda_a L_{adv} + \lambda_e L_{excl} + \lambda_s L_{syn}. \tag{4.2}$$

See Sec. 4.8 for the weights λ_r, λ_d, λ_a, λ_e, and λ_s. See Algorithm 4 for pseudo code.

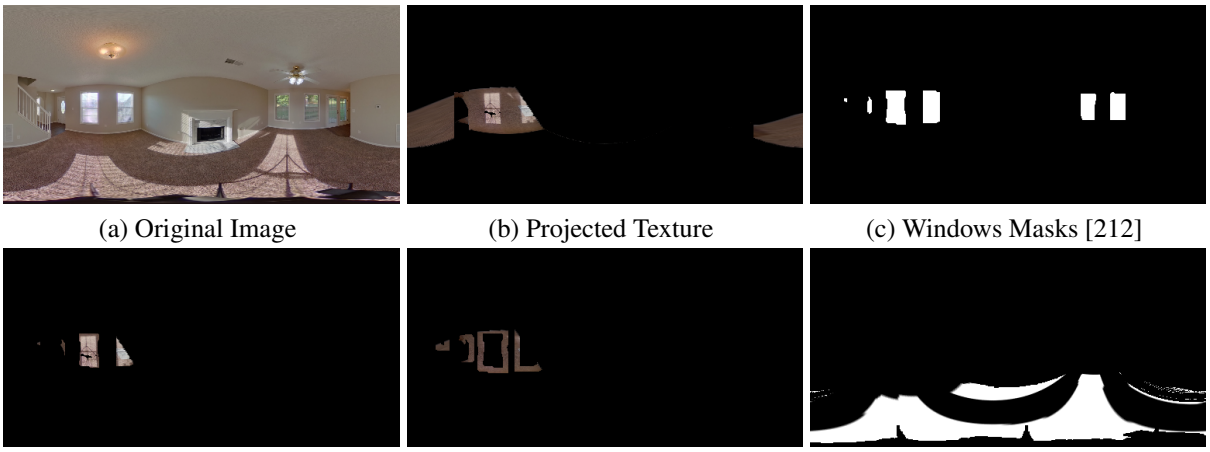
The reconstruction loss L_{recon} and the adversarial loss L_{adv} are the same as in Stage 1; however, the coarse mask M is replaced by its fine version M'. Instead of $Loss_1$, we use perceptual loss [90] $L_{diff} = Loss_p(D + S \cdot M', I)$ to enhance visual quality. The exclusion loss [242] $L_{excl} = Loss_e(D, S)$ applied to region M' is for minimizing the correlation between the gradients of D and S.

On-the-Fly Data Synthesis: Even with the four loss terms described above, the diffuse image does occasionally contain specular residues (Fig. 4.10). Thus, we propose an on-the-fly data synthesis method to generate training data by combining the predicted diffuse component D_1 of one image and the specular component S_2 of another image. The synthetic image $I_s = D_1 + S_2$ is fed to DiffuseNet and a supervised loss $L_{syn} = Loss_1(D, D_1) + Loss_p(D, D_1)$ is applied. Specifically, when $S_2 = 0$, this loss prevents the DiffuseNet from removing anything from a diffuse image, which implicitly encourages the network to remove the specular reflection completely in one step. We implement this loss by combining two images in the same minibatch (similar to mixup [239]) or one image and its horizontally flipped version. Ablation study in Sec. 4.8 shows the effectiveness of this loss.

Fig. 4.9 shows representative results with low resolution 256×512, due to limited computing. We now present a method to increase the resolution to 1024×2048 in Sec. 4.5.3.

4.5.3 Spatial Resolution Enhancement

Unlike in common super resolution methods [80, 125, 137] hallucinating unknown pixels, the original high-resolution RGB image is given in our case. The high-resolution diffuse image



(a) Original Image (b) Projected Texture (c) Windows Masks [212]

(d) Windows Regions (e) Surround-Windows (f) Coarse Sunlight

Figure 4.11: Intermediate results of coarse sun direction and direct sunlight estimation. Given a candidate sun direction $\mathbf{d}$, we project the floor texture to the walls (b) and obtain the windows regions (d) and surround-windows regions (e). The score of $\mathbf{d}$ is the difference between the mean intensities of (d) and (e). After finding the $\mathbf{d}$ with the optimal score, we project (c) back to the floor plane to obtain the coarse sunlight mask (f).

should be consistent with the original image. This scenario is studied by some previous works [17, 79, 144], but on modalities that are spatially smooth (*e.g.* depth map).

Our method is inspired by Double DIP [54], an unsupervised method for separating two layers from a single image. We extend it to the scenario where the low-res versions of the two layers are available. Given the high-resolution *input* RGB image I_{hr}, low-resolution *inferred* diffuse D_{lr} and specular S_{lr} images, we recover high-resolution diffuse D_{hr} and specular S_{hr} images by optimizing on the three images only. Specifically, two networks taking a noise image as input are used to predict D_{hr} and S_{hr}. The loss L consists of high-resolution reconstruction L_{recon} and exclusion L_{excl} losses, low-resolution diffuse L_{diff} and specular L_{spec} losses, and a low-resolution gradient loss L_{grad}:

$$L = \lambda_r L_{recon} + \lambda_e L_{excl} + \lambda_d L_{diff} + \lambda_s L_{spec} + \lambda_g L_{grad}. \tag{4.3}$$

See Sec. 4.8 for the weights λ_r, λ_d, λ_s, λ_e, and λ_g.

Losses $L_{recon} = Loss_1(D_{hr} + S_{hr}, I_{hr})$ and $L_{excl} = Loss_e(D_{hr}, S_{hr})$ are is defined similarly as in Sec. 4.5.2. Let D_{ds} and S_{ds} be the down-sampled versions of D_{hr} and S_{hr}, respectively. Then $L_{diff} = Loss_1(D_{ds}, D_{lr})$ and $L_{spec} = Loss_1(S_{ds}, S_{lr})$ force the down-sampled images to be similar to their low-resolution counterparts. To prevent the specular structure leaking to D_{hr}, the gradient loss $L_{grad} = Loss_1(\nabla D_{ds}, \nabla D_{lr})$ is introduced to preserve the diffuse structure. We weight L_{grad} based on the intensity of S_{lr} to focus on strong reflections. Let S be the specular intensity. We define $\alpha = max(0, min(1, (S - 0.005)/(0.05 - 0.005)))$. The weight is obtained by blurring α with a Gaussian kernel ($\sigma = 5$). This weighting strategy is only for Specular module. Fig. 4.14 (a)(b)(c) shows an example result, where the details of the floor are well recovered in the specular regions.

4.6 Sun Direction Estimation and Floor Direct Sunlight Removal

After removing specular reflection, we remove direct sunlight on the floor while estimating the sun direction. A similar pipeline as Sec. 4.5 is adopted: we first obtain coarse sunlight masks from material semantics and room layouts, and then use the GAN-based method to accurately recover the sunlight component. The guided resolution enhancement is applied as post-processing. The sun direction is also obtained in the pipeline. This pipeline can be applied for a single panorama or multiple panoramas sharing the same sun direction. We first explain the single image case and then extend to multiple images. Currently we only handle floor regions because the simple geometry allows a computationally efficient implementation. We will extend to walls in the future.

Algorithm 5 Coarse Sun Direction Estimation and Sunlight Mask Generation (single image version)

Input: Original Image I (Fig. 4.11 (a)), Floor Mask F, Window Mask W (Fig. 4.11 (c)), Room Layout

Output: Sun Direction $\mathbf{d}^*$, Sunlight Mask M_c

for $\mathbf{d} = (\theta, \phi)$ in possible sun directions **do**
  $V \leftarrow$ Floor mask projected to wall according to $\mathbf{d}$
  $T \leftarrow$ Floor texture (sampled from I) projected to wall according to $\mathbf{d}$ # Fig. 4.11 (b)
  $W_i \leftarrow W \cdot V$
  $T_i \leftarrow T \cdot W_i$ # Window region, Fig. 4.11 (d)
  $m_i \leftarrow$ Mean intensity of T_i in region W_i
  $W_s \leftarrow (dilate(W) - W) \cdot V$
  $T_s \leftarrow T \cdot W_s$ // Surround-window region, Fig. 4.11 (e)
  $m_s \leftarrow$ Mean intensity of T_s in region W_s
  $s(\mathbf{d}) \leftarrow m_i - m_s$
end for
$\mathbf{d}^* \leftarrow \operatorname{argmax} s(\mathbf{d})$
$M_s \leftarrow$ Window mask projected to floor according to $\mathbf{d}^*$ # Sunlight mask, Fig. 4.11 (f)

4.6.1 Coarse Sun Direction and Sunlight Estimation

To coarsely estimate the sun direction, we use the simple observation that *sunlight enters a room through transparent materials, creating bright shading on the floor.* Such transparent materials usually include glass windows and doors. For simplicity, we only consider windows. If we project the floor texture back to the wall based on sun direction, the windows should match the bright textures. The optimal sun direction should maximize the matching score.

Given sun direction $\mathbf{d}$, we calculate its score as follows (Algorithm 5): As shown in Fig. 4.11 (b), we first project the floor texture back to the wall according to the sun direction $\mathbf{d}$. By masking it with window region (c), we obtain (d). We also obtain a 7-pixel-wide surround-window region (e) via morphological operations. The score is defined as the difference between the mean intensities of the valid regions in (d) and (e). The optimal direction is obtained by maximizing the score. In practice, we adopt a coarse-to-fine strategy for searching $\mathbf{d}^*$. We first search elevation angle θ with step size $= 5^\circ$ and azimuth angle ϕ with step size $= 10^\circ$ to obtain an approximate direction. A better direction is then searched in a small range (10° for θ and 20° for ϕ) centered at the approximate result with step size $= 1^\circ$. We extend this approach to multiple images with the same sun direction by summing the scores of each image. In our experiments, we use the panoramas on the same floor for this calculation.

With the sun direction, the coarse direct sunlight mask (Fig. 4.11 (f)) is obtained by projecting the window mask (c) to the floor. It acts as the supervision signal for accurate sunlight estimation in Sec. 4.6.2.

4.6.2 GAN-Based Sunlight Refinement

Similar to Sec. 4.5.2, the coarse sunlight mask acts as the supervision signal for the GAN-based method to accurately estimate the sunlight component. Unlike specular reflections modeled in an additive way [133, 179], we model the direct sunlight effects as per-pixel scale factors based on multiplicative modeling [187, 237]. Let A be the ambient image after removing sunlight, and B be the ratio between direct sunlight and ambient light. Then the image formation model of diffuse image is $D = A \cdot (1 + B)$. We define $C = 1/(1 + B)$ so that $0 < C < 1$. Similar to Sec. 4.5.2, two networks are used to predict A and C respectively, but replacing I by D, D by A, S by C, and reconstruction loss by $L_{recon} = Loss_1(A/C, D)$. After estimating the sunlight-to-ambient ratio $B = 1/C - 1$, we refine the sun direction by running the algorithm in Sec. 4.6.1 again, with I replaced by B.

Because the sunlight region is usually saturated in the tone-mapped LDR ZInD panoramas and provides little information for recovering the ambient image (image without sunlight), we fill the sunlight regions using image inpainting [123]. The inpainting mask is obtained by thresholding and dilating the sunlight scale. A finer C is calculated as the division between the inpainted image and the diffuse image. This works well for empty homes with the same texture on the floor, but it might fail when we work around different floor textures or if there were other furniture presented. As inpainting algorithms improve, we can plug-and-play those in our pipeline.

Specifically, to inpaint the image, we train RFRNet [123] on our training set with masks randomly generated by projecting windows to the floor using random sun directions. The mask generated from one panorama is also applied to other images. Random dilation and noise are also applied for data augmentation. We adopt a two-stage inpainting method: We first inpaint it with an initial mask. Then we calculate a finer mask by comparing the inpainting result and the original image. Finally we inpaint with the fine mask again. Concretely, we follow the procedure described below: First, we obtain the inverse sunlight scale C by running the inference pass of the GAN network. In practice, we found that the specular reflection removal method occasionally removes direct sunlight. If the inference is done on the diffuse image, those removed sunlight could not be detected. Thus, we train the GAN method on diffuse images but infer on the original

(a) RGB Image (b) Initial Mask (c) Initial Inpainting Result

(d) Diffuse Image (e) Final Mask (f) Final Inpainting Result

Figure 4.12: Intermediate results of inpainting

RGB images to detect sunlight. Then, we propagate the inverse sunlight scale C from other panoramas on the same floor to the current view. A discount factor $cos^2\theta$ is used for weighting the propagated inverse sunlight scale C, where the θ is the altitude angle of the pixel in the current view. After that, we obtain the inpainting mask by thresholding and dilating the inverse sunlight scale. The threshold is set to be 0.95 for C and the dilation range is 3 pixels. After inpainting, we calculate the difference between the original image and the inpainted images, and remove pixels which are brighter after inpainting from the mask. We further dilate this mask by 1 pixel and send it to the inpaining network again. Fig. 4.12 shows an example of the inpainting process. (b) is the inital mask by thresholding the sunlight scale C. (c) is the initial inpainting result. (e) and (f) are the final mask and inpainting result. It is a challenging case because of the distorted tile patterns in a panoramic view.

Fig. 4.13 shows example results with resolution 256×512. To increase the spatial resolution to 1024×2048, we adopt the same approach as Sec. 4.5.3 by replacing the reconstruction loss by the sunlight image formation model. Fig. 4.14 (d)(e)(f) shows a representative resolution enhancement result. The high-resolution ambient image is consistent with the original image without sunlight residues.

4.7 Implementation Details

We use U-Net [172] for DiffuseNet and SpecularNet, and FCN [141] for discriminator, optimized with Adam [108]. See Tabs. 4.3, 4.4, and 4.5 for the building blocks of the networks, the generator architecture, and the discriminator architecture, respectively. DiffuseNet and SpecularNet share the same generator architecture. Specifically, in Stage 1 of diffuse-specular separation, the DiffuseNet uses a residual representation and predicts the difference between the original image and diffuse image. We found that it helps preserve the sparsity of specular image. In Stage 2, we input a 3-channel noise image together with the masked RGB image to the DiffuseNet to provide

Type	Components
inconv	[Conv3×3 + Activation + Normalization]×2
down	[Conv3×3 + Activation + Normalization]×2 + MaxPool2×2
up	Upsample + [Conv3×3 + Activation + Normalization]×2
outconv	Conv1×1

Table 4.3: Network components

Layer Name	Type	Input	Output Channels
inc	inconv	input image	64
down1	down	inc	128
down2	down	down1	256
down3	down	down2	512
down4	down	down3	512
up1	up	down4, down3	256
up2	up	up1, down2	128
up3	up	up2, down1	64
up4	up	up3, inc	64
outc	outconv	up4	3

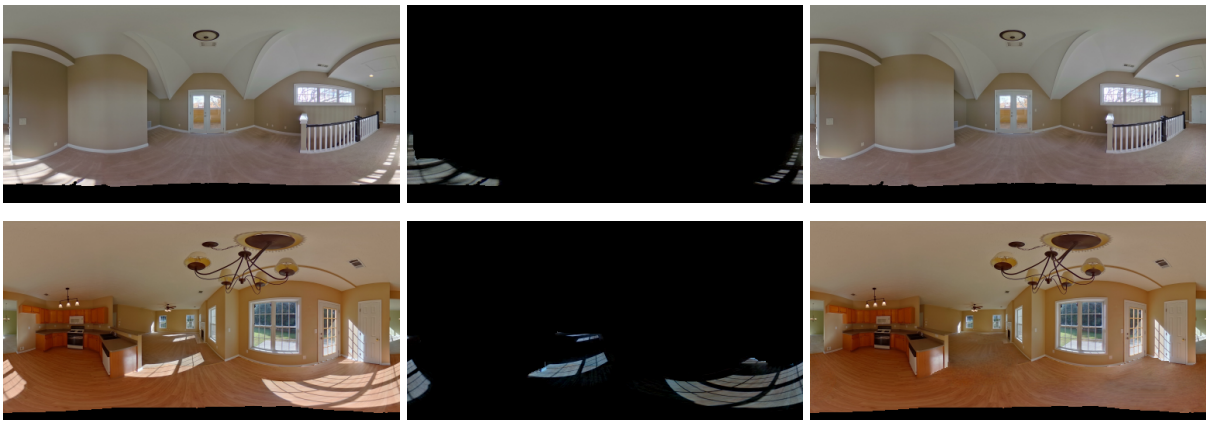
Table 4.4: Generator network architecture. We use ReLU [154] for activation, and Instance Normalization [205] for normalization.

information for it to hallucinate textures in areas with strong reflections. In sunlight detection, we additionally apply a "sigmoid" operation to the network output, making the predicted inverse sunlight scale C be between 0 and 1.

The GAN networks use learning rate 10^{-4}, weight decay 10^{-5}, and batch size 16. For Diffuse-Specular Separation, $\lambda_r = \lambda_d = 1$, $\lambda_a = 2$, $\lambda_e = 0.4$, $\lambda_s = 0.1$. For Direct Sunlight Detection, because fewer images have direct sunlight than specular reflection, we increase the adversarial loss to $\lambda_a = 20$. The discriminator loss is also multiplied by 10. We determine the epoch to stop training by inspecting the visual result on the training set. Usually it takes about 50 epochs for Stage 1 and 100 epochs for Stage 2. For sunlight detection, it usually takes about 30 epochs. The resolution enhancement networks use learning rate 10^{-4}, weight decay 10^{-5} and batch size 1. We empirically set $\lambda_r = 3$, $\lambda_e = 0.01$, $\lambda_d = \lambda_s = 1$, $\lambda_g = 20$. The optimization usually converges in 2000$\sim$3000 iterations. In our experiment, we optimize for 3000 iterations. The GAN takes $\sim$2 days for training and 0.05s for inference. Resolution enhancement converges in $\sim$20 min.

Layer Name	Type	Input	Output Channels
inc	inconv	input image	64
down1	down	inc	128
down2	down	down1	256
down3	down	down2	256
outc	outconv	down3	1

Table 4.5: Discriminator network architecture. We use Leaky ReLU [145] for activation, and Batch Normalization [82] for normalization.



(a) Diffuse Image (b) Direct Sunlight (c) Ambient Image

Figure 4.13: Floor direct sunlight removal results. Sunlight image (b) is the difference between (a) and (c). Our method handles direct sunlight with various shapes.

(a) High-Res RGB (b) Low-Res Diffuse (c) High-Res Diffuse

Sampled from (a) Sampled from (b) Sampled from (c)

(d) High-Res RGB (e) Low-Res Ambient (f) High-Res Ambient

Sampled from (d) Sampled from (e) Sampled from (f)

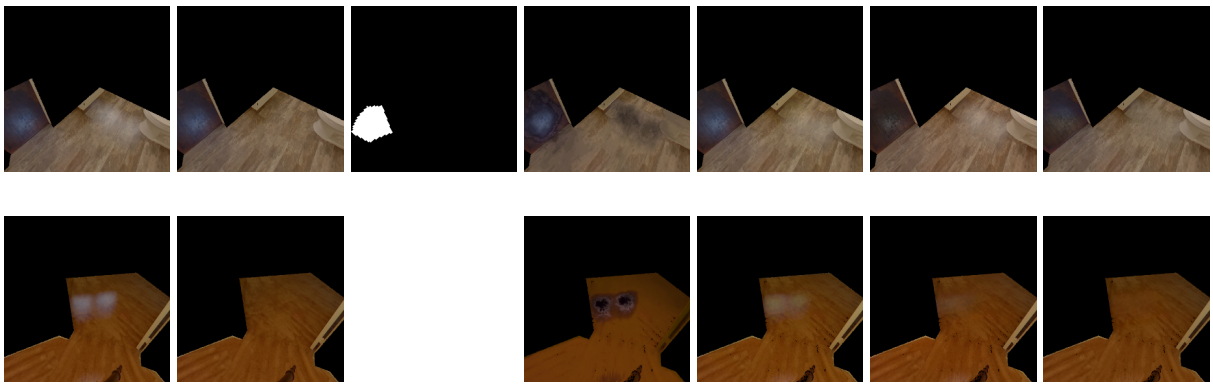
Figure 4.14: Resolution enhancement results. (a)(b)(c) show an example of enhancing specular removal result. The high-res diffuse image (c) increases the resolution of (b), with details consistent with the original image (a). Similarly, (d)(e)(f) show an example where the resolution of sunlight removal result is faithfully enhanced. Note that we focus on the floor region so the sunlight on the walls is outside the scope of the current implementation and thus is not removed.

(a) RGB (b) Human GT (c) Coarse Mask (d) Trained w/o GAN (e) Our Full Method

Figure 4.15: Detection of specular reflection. Our full method outperforms the method trained using coarse masks as labels without GAN, meaning the effectiveness of the proposed GAN-based approach.

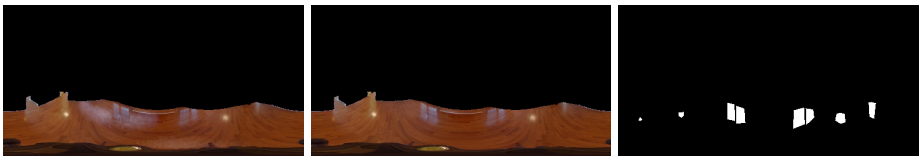
(a) Image (b) Yang *et al*. (c) IBCLN (d) ERRNet (e) Ours

Figure 4.16: Qualitative comparison of specularity removal with Yang *et al*. [231], ICBLN [121] and ERRNet [220]. Our method successfully removes specular reflections, while the compared methods cannot fully remove them.

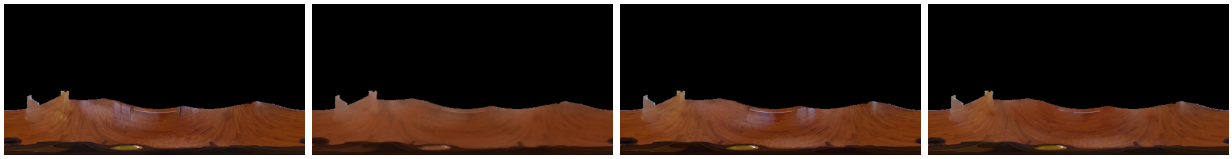


(a) RGB Image (b) GT Diffuse (c) Validity (d) Yang *et al*. (e) IBCLN (f) ERRNet (g) Ours

Figure 4.17: Comparison on synthetic perspective images with (d) Yang *et al*. [231], (e) IBCLN [121], and (f) ERRNet [220]. The manually annotated validity mask excludes specular regions in the ground truth diffuse image (b) for quantitative evaluation.



(a) RGB Image (b) GT Diffuse (c) Validity Mask



(d) Only Stage 1 (e) Merge two stages (f) No on-the-fly synthesis (g) Our full method

Figure 4.18: Ablation study on synthetic panoramic images. The manually annotated validity mask excludes specular regions in the ground truth diffuse image (b) for quantitative evaluation.

4.8 Experimental Analysis

Specular Reflection Detection: We manually annotate specular regions on the 2,785 test panoramas for quantitative analysis. The annotation is coarse (Fig. 4.7 (f)) because there is no clear boundary of specular regions. For a fair comparison, we report the mean IoU with respect to the best threshold for binarization for each method. To show the GAN-based method introduces priors for refining the coarse mask, we train a network with the same architecture as SpecularNet in a standard supervised manner (without GAN), using the coarse mask as labels, resulting mean IoU=19.5%. We also compare with the coarse mask itself (IoU=9.7%). Our full method (IoU=**40.3%**) outperforms the others, demonstrating the effectiveness of GAN. See Fig. 4.15 for an example result.

Specular Reflection Removal: We compare with a bilateral filtering based method [231] and two supervised deep learning methods IBCLN [121] and ERRNet [220]. We use the pre-trained models provided by the authors. Because they are trained on perspective images, we train our method from scratch on perspective views sampled from the panoramic images for a fair comparison. As in Fig. 4.16, our method removes specular reflections on different materials, while the other methods do not completely remove them. It could be because our method has been trained on a similar distribution, whereas the pre-trained models have been trained on most likely different domains. This shows the advantage of training on scenes without ground truth, which would be impossible with the supervised methods.

For quantitative comparison, we render specular reflections on the floor using the estimated environment maps with random floor roughness. We combine a random RGB image from the test set and a random synthetic specular image to generate 1,000 test images (resolution 256×256). As shown in Fig. 4.17, we mask the annotated specular regions defined for "Specular Reflection Detection" evaluation when calculating metrics to avoid the confusion between synthetic and real reflections. Tab. 4.6 lists comparative performance numbers in terms of PSNR and MS-SSIM[217]. We also define images with mean specular intensity ranking top 10% in the test data as "strong reflection" and report the performance in Tab. 4.6. Our method performs better on

Method	All Testdata		Strong Reflection	
	PSNR	MS-SSIM	PSNR	MS-SSIM
Yang *et al.*	26.19	0.7515	25.90	0.6646
IBCLN	31.95	0.9217	30.50	0.8544
ERRNet	31.60	0.9271	30.75	0.8756
Ours	**34.95**	**0.9365**	**33.99**	**0.9011**

Table 4.6: Quantitave comparison of specular reflection removal on synthetic perspective images with Yang *et al.* [231], IBCLN [121], and ERRNet [220]. Our method outperforms the others, especially for strong reflections.

Method	All Testdata		Strong Reflection	
	PSNR	MS-SSIM	PSNR	MS-SSIM
Only stage 1	26.96	0.9393	21.14	0.9016
Merge two stages	27.68	0.9380	**23.66**	0.9108
No on-the-fly synthesis	28.75	0.9520	22.28	0.9171
Our full method	**29.06**	**0.9534**	22.78	**0.9219**

Table 4.7: Ablation study of specular reflection removal on synthetic panoramic images. The full method outperforms other choices, showing the benefits of the proposed two-stage approach with on-the-fly data synthesis.

both intensity and structural similarity.

We also conduct a quantitative ablation study (Fig. 4.18, Tab. 4.7) on 1,000 panoramic images of resolution 256×512 rendered in the same way as the perspective ones. The other variants are: only doing Stage 1, merging the two stages by training on coarse mask using the loss of Stage 2, and removing on-the-fly data synthesis. The results show that the full method performs the best overall compared to the other variants. Note that although merging two stages achieves a better PSNR on strong reflections, the results are much blurrier than the full method, leading to a lower MS-SSIM score.

Our specular reflection removal method may fail on certain cases. Fig. 4.19 shows a failure case where the specular reflection of a small wall lamp (right side of the image) is not removed.

(a) RGB Image (b) Specular Reflection (c) Diffuse Image

Figure 4.19: Failure case of diffuse-specular separation. The wall lamp reflection at the right side is not removed.

(a) RGB (b) Human GT (c) Coarse Mask (d) Trained w/o GAN (e) Our Full Method

Figure 4.20: Detection of direct sunlight. Our full method outperforms the method trained using coarse masks as labels without GAN, meaning the effectiveness of the proposed GAN-based approach.

(a) Original Image (b) Direct Sunlight (c) Ambient Image

Figure 4.21: Failure case of direct sunlight removal. The weak sunlight is not detected by our method because our supervision signal is based on intensity.

(a) RGB Image (b) GT (32.73°) (c) Our Estimation (40°)

(d) Elevation=25° (e) Elevation=35° (f) Elevation=45°

Figure 4.22: Rendering with different sun elevation angles. Although the shadow of the clothes hanger moves across different sun elevations, the rendering results within 10° error show high perceptual quality. It is hard for a human to tell which one from (b) (c) (d) (e) is rendered with the correct elevation based on visual observation.

(a) High-Res RGB (b) Low-Res Diffuse (c) High-Res Diffuse

(d) Low-Res Ambient (e) High-Res Ambient

Sampled from (a) Sampled from (b) Sampled from (c) Sampled from (d) Sampled from (e)

Figure 4.23: Resolve specular-sunlight confusion. To resolve the specular-sunlight confusion, we assume the sunlight component takes all the energy while the specular component is zero in sunlight regions.

Sun Direction Estimation: The dataset provides sun elevation calculated from geolocation and time. Because the uploading time instead of the capturing time was used, the ground truth is approximate. We evaluate on floors with visible sunlight that have been annotated manually. We also report the performance on training set since the ground truth elevation is not used for training. We compare with the coarse direction without the GAN refinement in Tab. 4.8. The coarse mask shows $\sim$48% accuracy with error $\leq 10^\circ$, while the full method improve it to $\sim$60%. The numerical error can be misleading, because even if the error seems large, there is little visual difference. Fig. 4.22 shows rendering results for different sun elevations. Although the shadow of the clothes hanger moves across different sun elevations, the rendering results within 10° error are of high perceptual quality. It is difficult to tell which one from (b) (c) (d) (e) is rendered using the correct elevation.

Sunlight Detection: We manually annotate the sunlight regions on the 2,785 test panoramas and compare with the coarse mask (IoU=9.3%) and the supervised training method using the coarse mask as labels without GAN (IoU=40.2%). Our full method (IoU=**47.7%**) outperforms them, demonstrating that our GAN-based refinement helps. See Fig. 4.20 for qualitative results. Fig. 4.21 shows the failure of our sunlight detection method when the sunlight is very weak.

Resolution Enhancement: High-res images in Fig. 4.14 appear sharper with details and are consistent with the original image. Fig. 4.23 shows an challenging case where specular reflections and direct sunlight overlap. Usually the specular reflection removal method removes part

Method	Train Set (243 valid floors)		Test Set (87 valid floors)	
	Error $\leq 5°$	Error $\leq 10°$	Error $\leq 5°$	Error $\leq 10°$
Coarse	30.5%	49.4%	31.0%	47.1%
Full method	**40.3%**	**60.9%**	**34.5%**	**58.6%**

Table 4.8: Quantitative analysis of sun elevation estimation. After using the GAN-based sunlight detection method, the estimation is improved compared with the coarse direction.

Method	PSNR (dB)
Guided Deep Encoder	33.69
Ours	**39.63**

Table 4.9: Quantitative results of spatial resolution enhancement. Our method outperforms the baseline method Guided Deep Encoder [204].

of the reflection/sunlight (Fig. 4.23 (b)). To resolve this specular-sunlight confusion, we assume the sunlight component takes all the specular/sunlight energy and the specular component is zero in sunlight regions. This is based on the assumption that direct sunlight is usually much brighter than the specular reflection. It is implemented by forcing the specular component in such region to be zero in the spatial resolution enhancement step. As mentioned in the conclusion section in the main paper, the accurate separation of overlapping specular refection and direct sunlight is a potential future direction.

We also conduct a quantitative comparison of spatial resolution enhancement with Guided Deep Encoder [204], a deep-image-prior based method. We render specular reflections on floors using the same method as for the evaluation of specular reflection removal task. The result is shown in Tab. 4.9. Our method significantly outperforms Guided Deep Encoder on PSNR, as evidenced by the fewer artifacts from our method.

Furniture Insertion: We manually insert furniture models from AdobeStock[1] and render results with Mitsuba [85]. Specular and sunlight effects are rendered separately and combined together to form the final image. We found that using a constant floor roughness=0.03 for rendering the occluded specular component shows a good perceptual result. We show high resolution appearance decomposition and furniture insertion results in Fig. 4.24. The objects correctly block sunlight and specular reflections. The blocked sunlight is correctly cast on the objects.

4.9 Limitations

Our work has two main limitations. First, the confusion between specular reflection and direct sunlight cannot be resolved when they overlap with each other. As shown in Fig. 4.23, currently we assume the sunlight component takes all the specular/sunlight energy and the specular component is zero in sunlight regions, because direct sunlight intensity is usually much stronger than specular reglections. However, it is an ad hoc solution. Multi-task learning using both coarse masks for simultaneous supervision could be a potential option to solve the problem better. Sec-

[1]https://stock.adobe.com/

Figure 4.24: Appearance decomposition and object insertion results. The effects (specular + sunlight) are 1.5× brightened. The perspective samples are shown under the panoramic views of original, ambient, and rendered images. The three examples cover different flooring (wood, tile, carpet) and different effects (both specular and sunlight, specular only, sunlight only). The inserted objects create realistic appearances when blocking specular reflection and the direct sunlight. The direct sunlight is cast upon the inserted objects and further reflected by other surfaces (*e.g.* mirror in the bottom example).

ond, we currently process only the floor. Applying similar techniques for the wall and ceiling is a direction we intend to pursue. In fact, our sun direction estimation can also be applied to walls, using wall-to-wall texture projection rather than floor-to-wall projection. However, some walls have more complex textures and geometry, leading to inaccurate estimation and a higher computational overhead. We will address these issues in the future.

4.10 Conclusion

In summary, we present a 3-step weakly-supervised approach for appearance decomposition: (1) finding coarse appearance location information as supervision signal using material semantics and geometry; (2) using a GAN-based adversarial supervision method to refine the decomposition; (3) using the high-res RGB image as guidance to enhance the resolution of the result. We apply this method to both floor diffuse-specular separation and direct sunlight removal, enabling high-quality object insertion applications. This work demonstrates the effectiveness of adversarial supervision from material-aware appearance locations.

The proposed method could potentially be extended to the removal of other sparse signals (*e.g.*, a specific type of furniture), as long as the coarse mask can be easily obtained. It might even be further extended to signals that are sparse in other domains instead of the pixel domain (*e.g.*, frequency domain). These are potential future directions as well.

Chapter 5

Confidence Supervision from Appearance Location

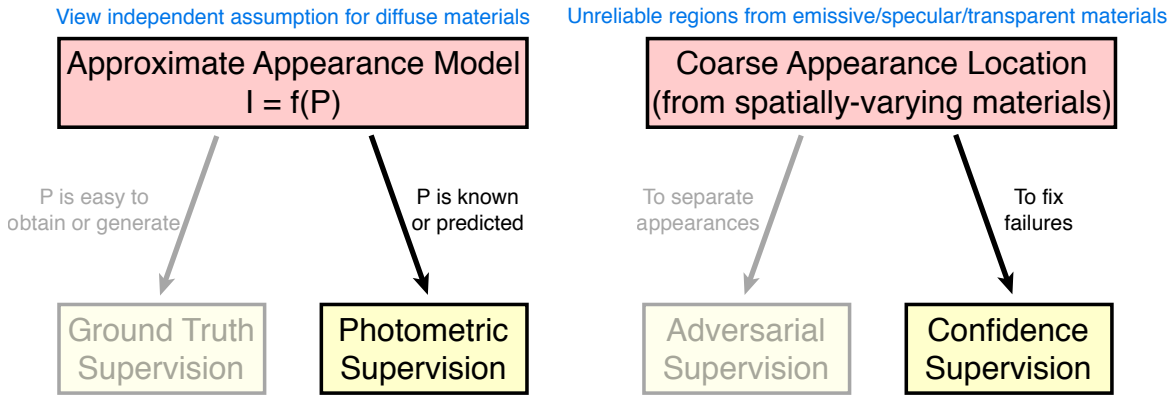
Chapter 4 demonstrates an approach of using appearance location information from material awareness to separate special appearances. However, in some other cases, we are interested in fixing failures caused by the special appearances rather than the accurate separation of them. This usually requires a weaker supervision signal. Confidence supervision often plays an role in such cases. Thus, in this chapter, we study a common scenario—road scenes and present a cross-spectral stereo matching method which fixes failures in unreliable regions using confidence supervision from non-Lambertian material location information. The connection of this approach to the framework proposed in introduction is shown in Fig. 5.1.

Specifically, we predict a disparity map from an RGB-NIR stereo pair. The deep model is trained in the form of analysis by synthesis: We synthesize the left-view image by warping the right-view based on disparity and compare it with the captured left-view image. However, this warping-based image synthesis only works for Lambertian materials. With material awareness, we can identify the uncertain predictions, and fix them by assigning low confidence and propagating disparities to them and. Currently, we can handle unreliable regions including light sources, glass, and glossy surfaces.

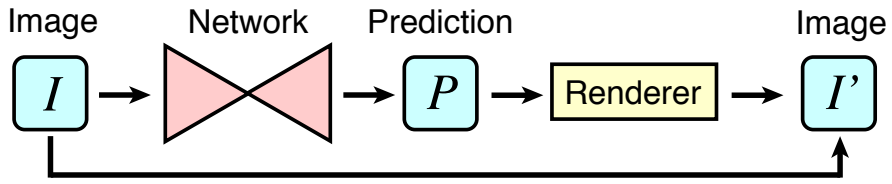
5.1 Application: Cross-Spectral Stereo Matching for Road Scenes

Cross-spectral imaging is broadly used in computer vision and image processing. Near infrared (NIR), short-wave infrared (SWIR) and mid-wave infrared (MWIR) images assist RGB images in face recognition [25, 70, 92, 120]. RGB-NIR pairs are utilized for shadow detection [173] and scene recognition [26]. NIR images also help color image enhancement [241] and dehazing [52]. Blue fluorescence and ultraviolet images assist skin appearance modeling [99]. Color-thermal images help pedestrian detection [81, 226].

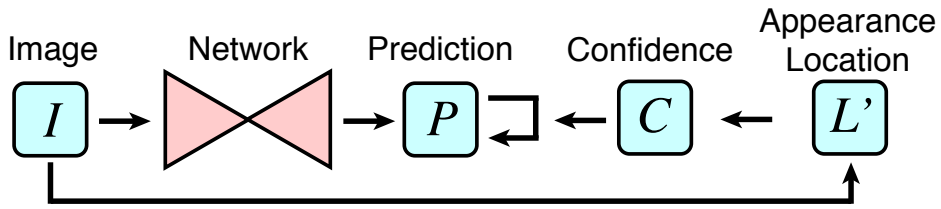
As multi-camera multispectral systems become more common in modern devices (*e.g.* RGB-NIR cameras in iPhone X and Kinect), the cross-spectral alignment problem is becoming critical since most cross-spectral algorithms require aligned images as input. Aligning images in hard-



(a) Framework



(b) Photometric Supervision



(c) Confidence Supervision

Figure 5.1: The road scene stereo matching task uses appearance model and appearance location information from material awareness for photometric and confidence supervisions. The photometric supervision based on view independent assumption fails on non-Lambertian materials. Thus, we recognize unreliable materials and assign low confidence values to them as supervision for fixing the failure in those regions.

(a) Left RGB (b) Right NIR (c) Difficult regions (d) Predicted disparity

Figure 5.2: A challenging case for RGB-NIR stereo matching and our result. Cyan box: The light source is visible in RGB but not in NIR. Yellow box: The transmittance and reflectance of the windshield are different in RGB and NIR. Red box (brightened): Some light sources reflected by the specular car surface are only visible in RGB. Our approach uses a deep learning based simultaneous disparity prediction and spectral translation technique with material-aware confidence assessment to perform this challenging task.

ware with a beam splitter is often impractical as it leads to significant light loss and thus needs longer exposure, resulting in motion blur. Stereo matching handles this problem by estimating disparity from a rectified image pair. Aligned images are obtained by image warping according to disparity. Stereo matching also provides an opportunity to obtain depth without an active projector source (as is done in the Kinect), helping tasks like detection [66] and tracking [194].

Cross-spectral stereo matching is challenging because of large appearance changes in different spectra. Figure 5.2 is an example of RGB-NIR stereo. Headlights have different apparent sizes or intensities in RGB and NIR. LED tail lights are not visible in NIR. Glass often shows different light transmittance and reflectance in RGB and NIR. Glossy surfaces have different specular reflectance. Additionally, vegetation and clothing often show a large spectral difference.

In this paper, we propose a deep learning based RGB-NIR stereo matching method in the form of analysis by synthesis without depth supervision. We use two networks to simultaneously predict disparity and remove the spectral difference. A disparity prediction network (DPN) estimates disparity maps based on an RGB-NIR stereo pair, and a spectral translation network (STN) converts an RGB image into a pseudo-NIR image. The losses are constructed by reprojecting and matching the NIR and the pseudo-NIR images, thus both the geometric and spectral differences are encoded.

Though the DPN and STN work well in many cases, certain materials cannot be handled correctly due to unreliable matching. 'Unreliable' means it is hard to find good matches due to large spectral differences, or the matches found correspond to incorrect disparities (*e.g.* matches on reflections). As shown in Figure 5.2 and 5.5, light sources in RGB may be absent in NIR, or show different apparent sizes resulting in incorrect matches. The transmitted and reflected scenes on glass and specular reflection on glossy surfaces may be matched but do not represent the real disparity. These are fundamental problems occurring often and cannot be ignored. We address the problems by using a material recognition network to identify unreliable regions, assigning low confidence values to uncertain predictions, and inferring the correct disparities from the context. The DPN loss assesses pixel confidence according to the material probability and the predicted disparity, and utilizes a confidence-weighted smoothing technique to backpropagate

more gradients to lower confidence pixels. This method significantly improves the results in unreliable regions.

We have collected 13.7 hours of RGB-NIR stereo frames covering different scenes, lighting conditions and materials. The images were captured from a vehicle driven in and around a city. Challenging cases for matching appear very frequently, including lights, windshields, glossy surfaces, clothing and vegetation. We labeled material segments on a subset of the images to train the aforementioned material recognition network. Additionally, we labeled sparse disparities on a test subset for evaluation. We experimented on this specific but important domain of driving in an urban environment and will extend it to indoor or other outdoor domains in the future. Experimental results show that the proposed method outperforms other comparable methods and reaches real-time speed. This method could be extended to other spectra like SWIR or thermal.

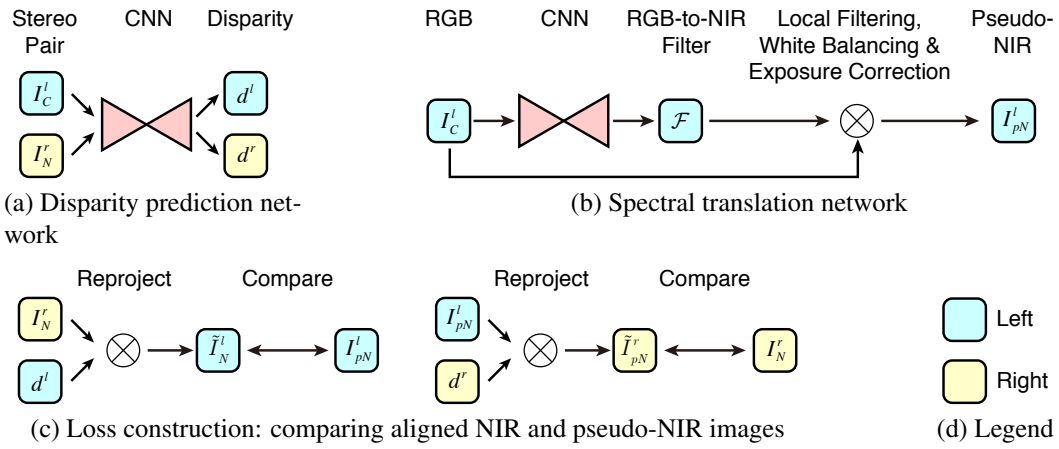
5.2 Related Work

Cross-Modal Stereo Matching: The key to cross-modal stereo matching is to compute an invariant between different imaging modalities. Chiu *et al*. [39] proposed a cross-modal adaptation method via linear channel combination. Heo *et al*. [72] presented a similarity measure robust to varying illumination and color. Heo *et al*. [73] also proposed a method to jointly produce color consistent stereo images and disparity under radiometric variation. Pinggera *et al*. [166] showed that the HOG [44] feature helps visible-thermal matching. Shen *et al*. [183] proposed a two-phase scheme with robust selective normalized cross correlation. Kim *et al*. [105] designed a descriptor based on self-similarity and extended it into a deep learning version [106]. Jeon *et al*. [87] presented a color-monochrome matching method in low-light conditions by compensating for the radiometric differences. These methods are based on feature or region matching without material awareness and are unreliable for materials such as lights, glass or glossy surfaces.

Unsupervised Deep Depth Estimation: Unsupervised depth estimation CNNs are usually trained with a smoothness prior and reprojection error. Garg *et al*. [57] proposed a monocular method with Taylor expansion and coarse-to-fine training. Godard *et al*. [62] presented a monocular depth network with left-right consistency. Zhou *et al*. [248] proposed a structure from motion network to predict depth and camera pose. Zhou *et al*. [246] presented a stereo matching method by selecting confident matches and training data. Tonioni *et al*. [202] showed that deep stereo matching models can be fine-tuned with the output of conventional stereo algorithms. All these methods deal with only RGB images rather than cross-spectral images, with no consideration for difficult non-Lambertian materials.

5.3 Simultaneous Disparity Prediction and Spectral Translation

To compensate for the appearance differences between RGB and NIR and extract disparity, we present an unsupervised scheme that trains two networks simultaneously to respectively learn disparity and spectral translation with reprojection error (Figure 5.3).



(a) Disparity prediction network

(b) Spectral translation network

(c) Loss construction: comparing aligned NIR and pseudo-NIR images

(d) Legend

Figure 5.3: Model overview. The disparity prediction network (DPN) predicts left-right disparity for an RGB-NIR stereo input. The spectral translation network (STN) converts the left RGB image into a pseudo-NIR image. The two networks are trained simultaneously with reprojection error.

5.3.1 Model Overview

Our approach consists of a disparity prediction network (DPN) and a spectral translation network (STN). The DPN design follows Godard *et al*. [62] but the input is replaced with an RGB-NIR stereo pair $\{I_C^l, I_N^r\}$, where superscripts l and r refer to the left and right images. Left-right disparity maps $\{d^l, d^r\}$ are predicted by DPN. STN translates an RGB image I_C^l into a pseudo-NIR image I_{pN}^l. Translation from NIR to RGB is not used because it is hard to add information to a 1-channel image to create a 3-channel image.

Both networks use reprojection error as the main loss. Given the right NIR image I_N^r and the left disparity d^l, we reproject the left NIR image $\tilde{I}_N^l$ via differentiable warping [83], similar to previous works [62, 113, 248]. Let $\omega(I, d)$ be the operator warping I according to disparity d, *i.e*., $\omega(I, d)(x, y) = I(x + d(x, y), y)$. Then $\tilde{I}_N^l = \omega(I_N^r, -d^l)$. Symmetrically, the warped pseudo-NIR image $\tilde{I}_{pN}^r = \omega(I_{pN}^l, d^r)$. Error is calculated between the warped NIR image $\tilde{I}_N^l$ and the pseudo-NIR image I_{pN}^l, and the warped pseudo-NIR image $\tilde{I}_{pN}^r$ and the NIR image I_N^r.

5.3.2 Disparity Prediction Network

The DPN predicts left-right disparities $\{d^l, d^r\}$ based on an RGB-NIR stereo pair $\{I_C^l, I_N^r\}$. The network structure proposed by Godard *et al*. [62] is adopted. Convolutional layers are followed by batch normalization [82] (except for output layers) and ELU [40] activation. The output disparity is scaled by a factor η for a good initialization. The loss has a view consistency term L_v, an alignment term L_a and a smoothness term L_s following Godard *et al*. [62].

$$L_{DPN} = \lambda_v(L_v^l + L_v^r) + \lambda_a(L_a^l + L_a^r) + \lambda_s(L_s^l + L_s^r) \tag{5.1}$$

For simplicity, only the left terms are described below and the right ones can be derived similarly. Multi-scale prediction is done by adding similar loss functions at four scales.

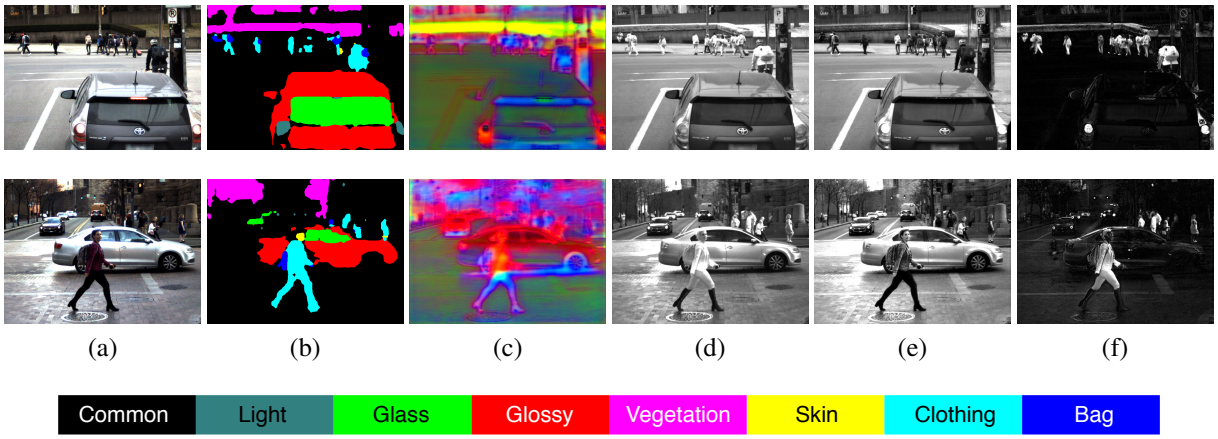


Figure 5.4: Intermediate results. (a) is the left RGB image. (b) is the material recognition result from DeepLab [33] (explained in Section 5.4.2). (c) shows the RGB-to-NIR filters corrected by exposure and white balancing. The R,G,B values represent the weights of R,G,B channels. (d) is the right NIR image. (e) is the warped pseudo-NIR image. (f) is the error between (d) and (e). Some clothing fails in spectral translation because the relationship between its RGB and NIR intensities is low. The structural similarity term in alignment loss (Equation 5.3) can partially solve this problem as long as the structure is preserved.

The view consistency term L_v^l describes the consistency of left-right disparity maps. Let N be the number of the pixels in one image, and Ω be the pixel coordinate space.

$$L_v^l = \frac{1}{N} \sum_{\mathbf{p} \in \Omega} |d^l(\mathbf{p}) - \omega(d^r, -d^l)(\mathbf{p})| \tag{5.2}$$

The alignment term L_a^l compares the intensity and structure between aligned NIR and pseudo-NIR images. Let $\delta(I_1, I_2)$ be the structural dissimilarity function [218]. Then,

$$L_a^l = \frac{1}{N} \sum_{\mathbf{p} \in \Omega} (\alpha \delta(I_{pN}^l, \tilde{I}_N^l)(\mathbf{p}) + (1 - \alpha) |I_{pN}^l(\mathbf{p}) - \tilde{I}_N^l(\mathbf{p})|) \tag{5.3}$$

where α is set to be 0.85 as suggested by Godard *et al.* [62].

The smoothness term L_s^l is edge-aware to allow noncontinuous disparity at image edges:

$$L_s^l = \frac{1}{N} \sum_{\mathbf{p} \in \Omega} ((\left| \frac{\partial d^l}{\partial x} \right| e^{-|S_x * I_C^l|} + \left| \frac{\partial d^l}{\partial y} \right| e^{-|S_y * I_C^l|})(\mathbf{p})) \tag{5.4}$$

where S_x and S_y are Sobel operators and the filtered RGB channels are averaged into one channel.

5.3.3 Spectral Translation Network

The RGB-NIR cameras are radiometrically calibrated and their varying white balancing gains (g_R for red and g_B for blue) and exposure times Δt_C and Δt_N are known. The spectral translation

network (STN) converts an RGB image I_C^l into a pseudo-NIR image I_{pN}^l via local filtering, white balancing, and exposure correction (Figure 5.3). Let $\mathcal{G}_{\boldsymbol{\theta}_1}$ be the white balancing operator with learnable parameter $\boldsymbol{\theta}_1$, and $\mathcal{F}_{\boldsymbol{\theta}_2}^{(\mathbf{p})}$ be the filtering operation with predicted parameter $\boldsymbol{\theta}_2$ for each position $\mathbf{p}$. The pseudo-NIR image is:

$$I_{pN}^l(\mathbf{p}) = \frac{\Delta t_N}{\Delta t_C} \mathcal{G}_{\boldsymbol{\theta}_1}(g_R, g_B) \mathcal{F}_{\boldsymbol{\theta}_2}^{(\mathbf{p})}(I_C^l(\mathbf{p})) \tag{5.5}$$

$\mathcal{G}_{\boldsymbol{\theta}_1}$ is a one-layer neural network learning parameters $\boldsymbol{\theta}_1 = (\theta_{11}, \theta_{12}, \theta_{13})$ with a sigmoid activation h,

$$\mathcal{G}_{\boldsymbol{\theta}_1}(g_R, g_B) = \beta h \left(\frac{\theta_{11}}{g_R} + \frac{\theta_{12}}{g_B} + \theta_{13} \right) \tag{5.6}$$

where, $\beta = 2$ is the maximum white balancing gain.

$\mathcal{F}_{\boldsymbol{\theta}_2}^{(\mathbf{p})}$ calculates a weighted sum of R,G,B channels. The weights are different for each position $\mathbf{p}$. Formally,

$$\begin{aligned} &\mathcal{F}_{\boldsymbol{\theta}_2}^{(\mathbf{p})}(I_C^l(\mathbf{p})) = \\ &\theta_{21}(\mathbf{p}) I_R^l(\mathbf{p}) + \theta_{22}(\mathbf{p}) I_G^l(\mathbf{p}) + \theta_{23}(\mathbf{p}) I_B^l(\mathbf{p}) \end{aligned} \tag{5.7}$$

where I_R^l, I_G^l, I_B^l are the three channels of I_C^l, and the weights $\boldsymbol{\theta}_2(\mathbf{p}) = (\theta_{21}(\mathbf{p}), \theta_{22}(\mathbf{p}), \theta_{23}(\mathbf{p}))$ are predicted by a filter generating network (FGN) [45]. To prevent the STN from learning disparity, we use a CNN with left-right symmetric filtering kernels. The structure of the FGN is the same as the DPN. The FGN accepts an RGB image and predicts an RGB-to-NIR filter (Figure 5.4 (c)).

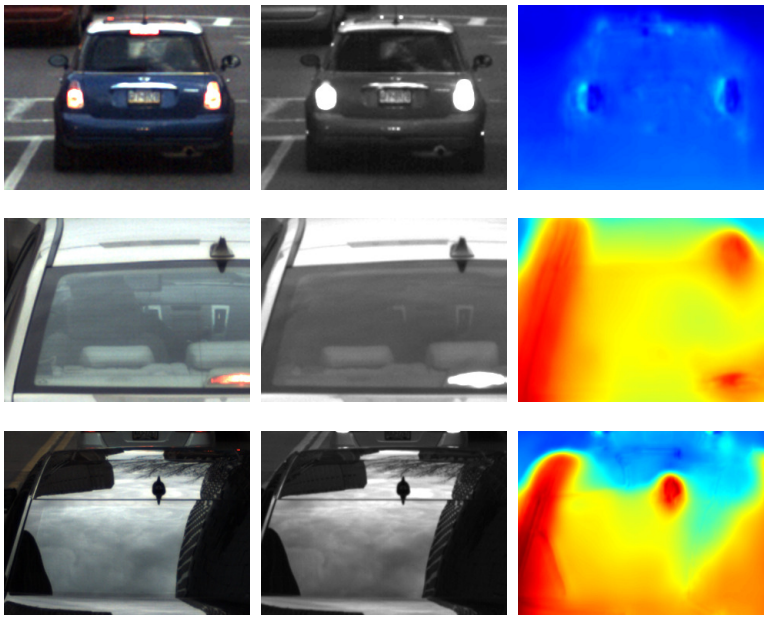
The STN loss matches the NIR and pseudo-NIR images:

$$L_{STN} = \frac{1}{N} \sum_{\mathbf{p} \in \Omega} (|I_{pN}^l(\mathbf{p}) - \tilde{I}_N^l(\mathbf{p})| + |I_N^r(\mathbf{p}) - \tilde{I}_{pN}^r(\mathbf{p})|) \tag{5.8}$$

where I_{pN}^l, $\tilde{I}_N^l$, I_N^r and $\tilde{I}_{pN}^r$ are, respectively, the pseudo-NIR image, the warped NIR image, the NIR image, and the warped pseudo-NIR image as defined in Section 5.3.1.

5.4 Incorporating Material-Aware Confidence into Disparity Prediction Network

Though the DPN and STN work well in many cases, they cannot handle certain materials including lights, glass and glossy surfaces due to unreliable matching. Matching on these materials is hard due to large spectral change (Figure 5.2) and not trustworthy because it does not represent the correct disparity (Figure 5.5). Such materials are common but difficult to identify without external knowledge. Assessing reliability by matching score or view consistency [153, 246] fails because unreliable regions may show high matching scores (Figure 5.5) and strong view consistency. A light source may show a different size in RGB and NIR and thus match at its edge instead of the center. Transmitted or reflected scenes may match perfectly but the predicted disparities do not correspond to the physical surfaces.



(a) RGB patch (b) NIR patch (c) Wrong disparity

Figure 5.5: Unreliable matching with high matching score. (c) is predicted by DPN without material awareness. Row 1: the light sources showing different sizes in RGB and NIR, and incorrectly match at the edges instead of the centers. Row 2: matching of transmitted scene does not represent the correct windshield disparity. Row 3: disparity of the reflected scene does not correspond to the car surface.

Our goal is to incorporate material-aware confidence into DPN loss (Equation 5.1) to solve this problem. We propose two novel techniques: (1) Propagate the disparity from the reliable to the unreliable regions using a new confidence-weighted smoothing technique (Section 5.4.1) and (2) Extend the DPN loss function to be material-aware by creating material-specific alignment and smoothness losses (Section 5.4.2). Section 5.4.3 discusses how to combine those two techniques to solve specific unreliable materials.

5.4.1 Confidence-Weighted Disparity Smoothing

Smoothing is a common technique to infer disparity in unreliable regions. However, a smoothness loss allows unreliable regions to mislead the reliable parts by forcing them to share similar disparity. As shown in Figure 5.6 (c), this results in the disparity at the side of the car to be misled by the wrong prediction on the windshield.

Confidence-weighted disparity smoothing uses confident disparities to "supervise" non-confident ones. Instead of fine-tuning [202] or bootstrapping [246], we change the backpropagation behavior of the smoothness loss so that it can be embedded in the DPN loss (Equation 5.12). Consider two neighbor pixels $\mathbf{p}_1$ and $\mathbf{p}_2$ with predicted disparities d_1 and d_2. A $L1$ smoothness loss is $L = |d_1 - d_2|$. Let $\mathbf{W}$ be all the parameters in the DPN, then $\frac{\partial L}{\partial \mathbf{W}} = \frac{\partial L}{\partial d_1}\frac{\partial d_1}{\partial \mathbf{W}} + \frac{\partial L}{\partial d_2}\frac{\partial d_2}{\partial \mathbf{W}}$. Assume that $\mathbf{p}_1$ is confident while $\mathbf{p}_2$ is unreliable. We want d_2 to follow d_1 without harming d_1. Let

Method	Common	Light	Glass	Glossy	Vegetation	Skin	Clothing	Bag	Mean	Time (s)
CMA	1.60	5.17	2.55	3.86	4.42	3.39	6.42	4.63	4.00	227
ANCC	1.36	2.43	2.27	2.41	4.82	2.32	2.85	2.57	2.63	119
DASC	0.82	1.24	1.50	1.82	1.09	1.59	**0.80**	1.33	1.28	44.7
Proposed	**0.53**	**0.69**	**0.65**	**0.70**	**0.72**	**1.15**	1.15	**0.80**	**0.80**	**0.0152**

Table 5.1: Quantitative results. Disparity RMSE in pixels is reported for each material. CMA [39] with searching step 0.01, ANCC [72] and DASC [105] with guided filtering [69] are tested on an Intel Core i7 6700HQ CPU. The proposed method is tested on a single NVIDIA TITAN X (Pascal) GPU. Our method outperforms the others and reaches real-time speed.

$\chi(\cdot)$ be the stopping gradient operator (a.k.a. 'detach' in PyTorch [163]) that acts as an identity mapping in the forward pass but stops gradients from being backpropagated through it in the backward pass. A confidence-aware loss is $L = |\chi(d_1) - d_2|$, preventing gradients being back-propagated through d_1. $\frac{\partial L}{\partial d_1}$ is set to be zero during backpropagation, *i.e.*, $\frac{\partial L}{\partial \mathbf{W}} = \frac{\partial L}{\partial d_2}\frac{\partial d_2}{\partial \mathbf{W}}$. This can be extended into a "soft" version. Generally, let $\mathbf{p}_1$ and $\mathbf{p}_2$ have confidences c_1 and c_2. We define relative confidences as $r_1 = \frac{c_1}{c_1+c_2}$ and $r_2 = 1 - r_1$, and the confidence-weighted loss as $L = r_1|\chi(d_1) - d_2| + r_2|d_1 - \chi(d_2)|$.

In practice, we consider a disparity map $d(x, y)$ and its known confidence $c(x, y)$ (defined in Section 5.4.3 using material). We present detailed expressions for the confidences by defining the pixel neighborhood in x and y directions. The relative confidences r^+ and r^- in $x-$direction are:

$$r^+(x, y) = \chi\left(\frac{c(x + 1, y)}{c(x + 1, y) + c(x - 1, y)}\right) \tag{5.9}$$

and $r^- = 1 - r^+$, where the $\chi(\cdot)$ prevents gradients to be backpropagted to the confidence. The confidence-weighted $L1$ smoothness loss along $x-$direction is:

$$\begin{aligned} L_x(d, c)(x, y) =& r^+(x, y)\left|\frac{\chi(d(x + 1, y)) - d(x - 1, y)}{2}\right| \\ &+ r^-(x, y)\left|\frac{d(x + 1, y) - \chi(d(x - 1, y))}{2}\right| \end{aligned} \tag{5.10}$$

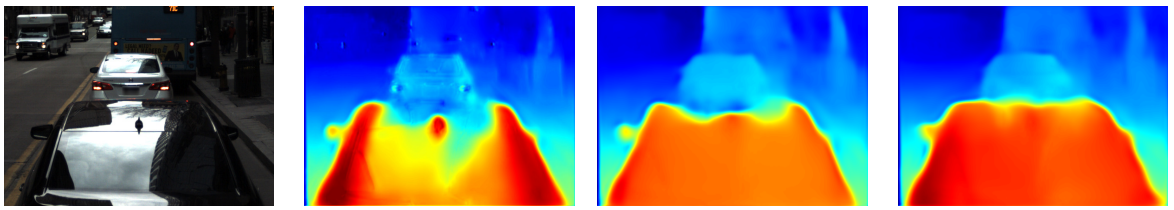
$L_y(d, c)$ is defined similarly for the $y-$direction. Then the complete confidence-weighted smoothness loss is:

$$L_{cs}(d, c) = L_x(d, c) + L_y(d, c) \tag{5.11}$$

As shown in Figure 5.6, the use of the confidence-weighted loss leads to better results than traditional smoothing.

5.4.2 Material-Aware Loss Function

A DeepLab [33] network is used to identify unreliable regions. It is trained separately and before the training of the DPN and STN networks. A set of 8 material classes $\mathcal{M} = \{$'light', 'glass',



(a) RGB (b) No material awareness (c) Smooth w/o confidence (d) Smooth w/ confidence

Figure 5.6: Comparison of smoothing with and without confidence. Smoothing without confidence makes the reliable matching around the car sides be misled by the unreliable matching on glass, which causes the predicted disparity (orange) to be smaller than the correct one (red). Introducing confidence addresses this issue.

'glossy', 'vegetation', 'skin', 'clothing', 'bag', 'common'} are predicted (Figure 5.4). 'Common' refers to any material not in the other classes. Let $\mathcal{M}^U$ be the subset of unreliable materials in $\mathcal{M}$. The DeepLab network takes a stereo pair as input and predicts left-right probabilities $\{\mu^l_m(\mathbf{p}), \mu^r_m(\mathbf{p})\}$ of each pixel $\mathbf{p}$ being material m.

To make the original DPN loss in Equation 5.1 material-aware, we introduce material-specific alignment and smoothness losses $L^l_{a,m}$ and $L^l_{s,m}$ respectively (similarly for the right terms). Thus, we re-write Equation 5.1 as:

$$\begin{aligned} L_{DPN} =& \lambda_v(L^l_v + L^r_v) \\ &+ \sum_{m\in\mathcal{M}} \lambda_{a,m}(\frac{1}{N}\sum_{\mathbf{p}\in\Omega}(\mu^l_m(\mathbf{p})L^l_{a,m}(\mathbf{p}) + \mu^r_m(\mathbf{p})L^r_{a,m}(\mathbf{p}))) \\ &+ \sum_{m\in\mathcal{M}} \lambda_{s,m}(\frac{1}{N}\sum_{\mathbf{p}\in\Omega}(\mu^l_m(\mathbf{p})L^l_{s,m}(\mathbf{p}) + \mu^r_m(\mathbf{p})L^r_{s,m}(\mathbf{p}))) \end{aligned} \tag{5.12}$$

For the reliable materials we use the same alignment and smoothness terms as in Equation 5.3 and 5.4, where the definition of confidence c is not required. For the unreliable materials, we use the confidence-weighted smoothness loss proposed in Section 5.4.1. We next describe how μ^l_m and μ^r_m are used to compute the confidence c in Equation 5.11.

5.4.3 Example Loss Terms of Unreliable Materials

Here we define the unreliable materials $\mathcal{M}^U = \{$'light', 'glass', 'glossy'$\}$ and present their loss terms.

Light Sources: Light sources like tail-lights, brake lights, bus route indicators and headlights result in unreliable matching. Thus the alignment term is $L^l_{a,light} = 0$. We assume that the light source shares the same disparity with non-light neighbors. The confidence c^l is computed using $1 - \mu^l_{light}$. Then Equation 5.11 (smoothness term) becomes:

$$L^l_{s,light} = L_{cs}(d^l, 1 - \mu^l_{light}) \tag{5.13}$$

Glass: Glass surfaces reflect and transmit light. We define the alignment loss $L^l_{a,glass} = 0$ considering its unreliable matching. But the dominated alignment term of common materials

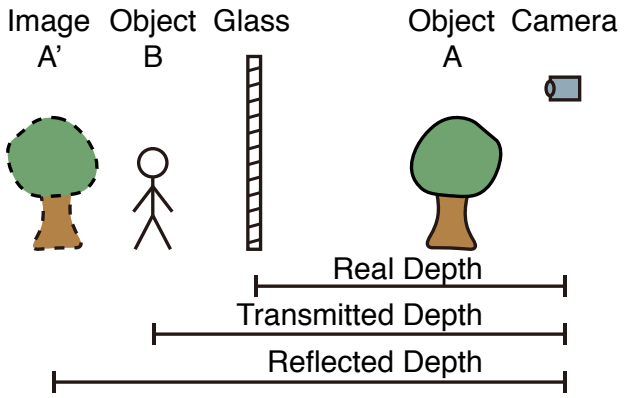


Figure 5.7: Transmitted and reflected scenes look farther than the real glass position.

still forces DPN to match the appearance on glass. As illustrated in Figure 5.7, both the reflected and transmitted scenes appear farther than the real position of glass. Therefore, we assign higher confidence to closer scenes with larger disparities. Assuming that glass can only be physically supported by 'common', 'glass', and 'glossy' materials, we define the confidence $c^l = (\mu^l_{common} + \mu^l_{glass} + \mu^l_{glossy})e^{\frac{d^l}{\sigma}}$. Thus the smoothness loss $L^l_{s,glass}$ is:

$$L^l_{s,glass} = L_{cs}(d^l, (\mu^l_{common} + \mu^l_{glass} + \mu^l_{glossy})e^{\frac{d^l}{\sigma}}) \tag{5.14}$$

where, σ is a constant parameter (details in Section 5.7).

Glossy Surfaces: Glossy surfaces exhibit complex specular reflection. We adopt the alignment term of common materials (Equation 5.3), considering that it still contains some reliable matching, and the smoothness term of glass (Equation 5.14), because the reflected scene has smaller disparity.

5.5 RGB-NIR Stereo Dataset

The dataset was captured by an RGB camera and a NIR camera mounted with $56mm$ baseline on a vehicle, alternating among short, middle and long exposures adapted by an auto-exposure algorithm at 20Hz. Close to 1 million 1164×858 rectified stereo frames equally distributed amongst the three exposure levels were collected. They were split into 12 videos, with a total length of 13.7 hours. The dataset covers campus roads, highways, downtown, parks and residential areas captured under sunny, overcast and dark conditions and includes materials such as lights, glass, glossy surfaces, vegetation, skin, clothing and bags. Reliable GPS and vehicle states (speed, vehicle pose, steering radius and traveled distance) are available for 70% of the data. Images are resized into 582×429 in all experiments.

Material and disparity labels are added to a subset of the middle-exposure images. The videos are split into two sets for training (8 videos) and testing (4 videos). 3600 frames are labeled with material segments in 8 classes (common, light, glass, glossy, vegetation, skin, clothing, bag). 5030 sparse points on 2000 testing images across the 8 materials are annotated with disparity. Depth sensors are not used because they often fail on glass and light sources. We calculated

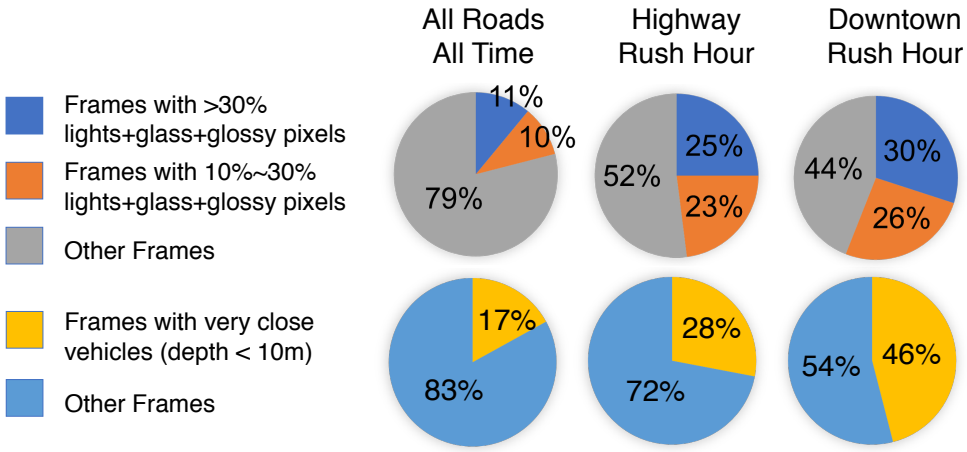


Figure 5.8: Material and vehicle statistics. Non-lambertian materials, including glass, glossy surfaces, and light sources, are common in road scenes. They can occupy a large portion of the image pixels, especially when vehicles are close to each other.

material and vehicle statistics in Fig. 5.8. It shows that non-lambertian materials, including glass, glossy surfaces, and light sources, are common in road scenes. They can occupy a large portion of the image pixels, especially when vehicles are close to each other.

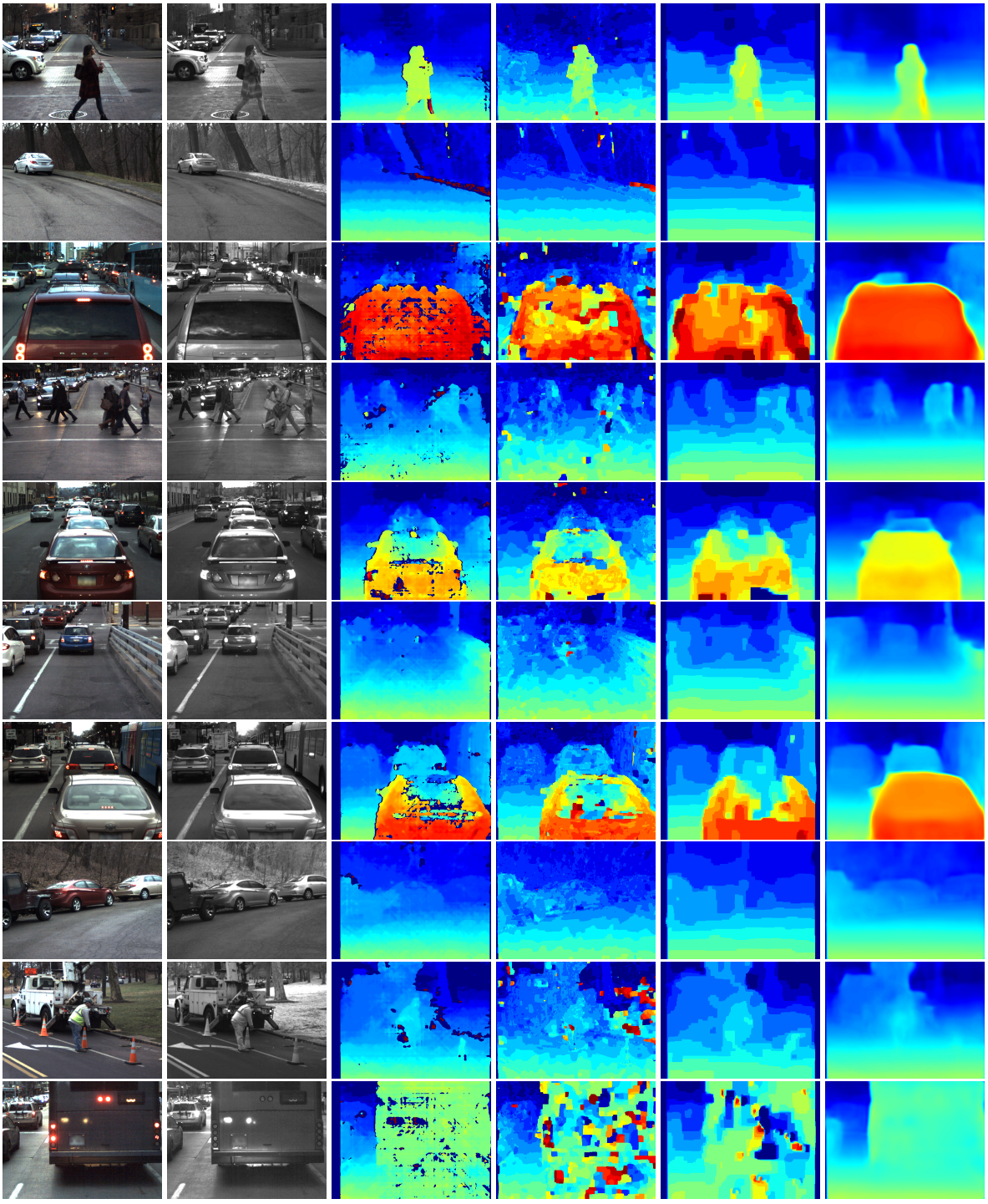
5.6 Implementation Details

DPN predicts the ratio between disparity and image width. The scaling factor η is 0.008 for the DPN and $1/3$ for the STN. The view consistency and alignment weights are $\lambda_v = 2$ and $\lambda_a = 1$ for all materials. The smoothness weights λ_s are 3000, 1000, and 80 for lights, glass and glossy surfaces, and 25 for other materials. The parameter in glass and glossy smoothness loss is $\sigma = 0.005$.

The DeepLab [33] net is fine-tuned from a model pre-trained on ImageNet [46], COCO [134] and Pascal VOC [50]. DPN and STN are trained on $40,000$ sampled middle-exposure images with Adam optimizer [109] (batch size=16, learning rate=0.00005). They are trained with material awareness for at least 12 epochs after 4 warmup epochs without it, taking about 18 hours on two TITAN X GPUs with PyTorch [163] code. Only the DPN is required for testing. Negative disparities are clamped to 0.

5.7 Experimental Analysis

Comparison: We have compared with Cross-Modal Adaptation (CMA) [39], ANCC [72] and DASC [105]. SIFT flow [135] search is constrained by epipolar geometry to obtain the whole image disparity in DASC. Disparity RMSE (Table 5.1), execution times (Table 5.1) and qualitative results (Figure 5.9) are presented. Our method outperforms the others, especially on lights, glass and glossy surfaces. Our method also provides cleaner disparity maps and clearer object

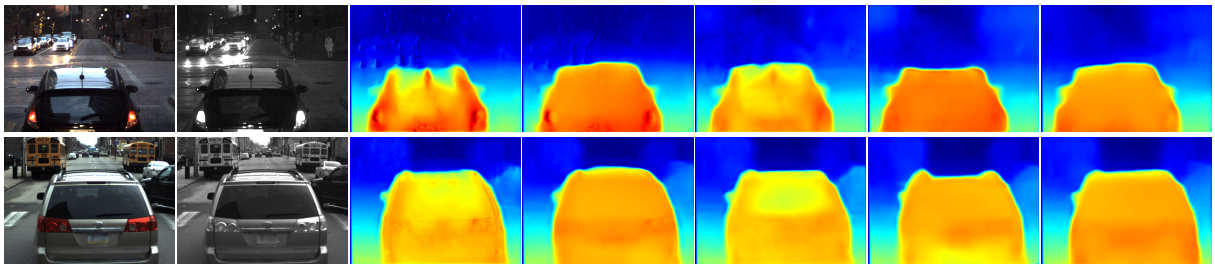


(a) Left RGB (b) Right NIR (c) CMA (d) ANCC (e) DASC (f) Proposed

Figure 5.9: Qualitative results on our dataset. Image contrast is adjusted for visualization. Compared with the baseline methods CMA [39], ANCC [72] and DASC [105], the proposed method provides less noisy disparity maps and performs better on lights (row 3, 5, 6, 7, 10), glass (row 3, 5, 7) and glossy surfaces (row 5, 7, 10) .

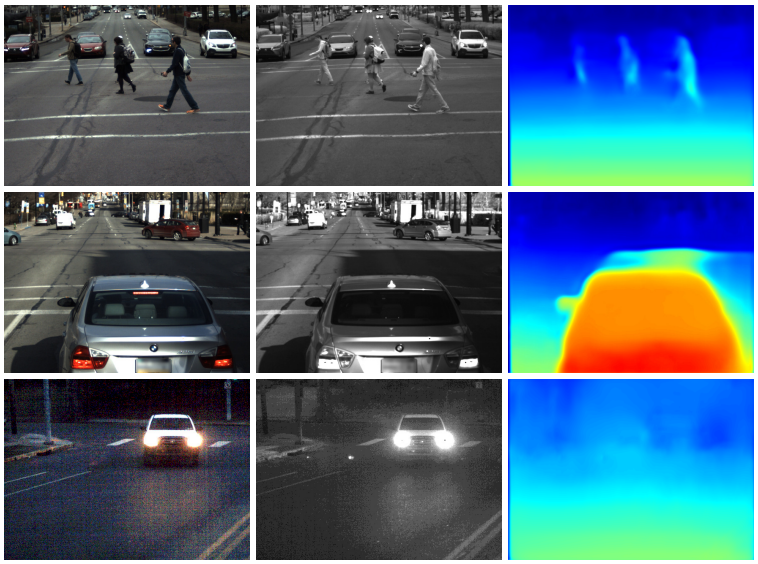
Method	Common	Light	Glass	Glossy	Vegetation	Skin	Clothing	Bag	Mean
Only RGB as DPN input	0.66	1.12	0.89	1.10	0.92	1.61	1.24	0.95	1.06
Averaging RGB as STN	0.52	0.80	0.74	0.78	0.76	1.30	1.21	1.04	0.89
No material awareness	**0.51**	1.08	1.05	1.57	**0.69**	**1.01**	1.22	0.90	1.00
Ignore light sources	0.54	0.81	0.74	0.71	0.76	1.37	1.17	1.10	0.90
Ignore glass	0.56	0.74	0.97	1.08	0.75	1.06	**1.02**	0.86	0.88
Ignore glossy surfaces	0.63	0.71	0.71	1.23	0.79	1.12	1.09	0.94	0.90
Smoothing w/o confidence	0.53	**0.69**	0.71	1.20	0.85	1.06	1.12	0.81	0.87
Full proposed method	0.53	**0.69**	**0.65**	**0.70**	0.72	1.15	1.15	**0.80**	**0.80**

Table 5.2: Ablation study. Network structure changes (row 1-2) result in the increase of error generally. Removing material awareness (row 3-7) leads to failure on corresponding materials. Smoothing without confidence (row 8) results in performance drop. There are small fluctuations but the full method performs better in general.



(a) Left RGB (b) Right NIR (c) No material (d) Ignore lights (e) Ignore glass (f) Ignore glossy (g) Full method

Figure 5.10: Qualitative material ablation study. Ignoring lights results in artifacts at light sources. Ignoring glass leads to wrong disparity predictions at windshields. Ignoring glossy surfaces causes failure at the specular top surfaces of cars.



(a) Left RGB (b) Right NIR (c) Predicted disparity

Figure 5.11: Failure cases. Row 1-3: failing to handle large spectral difference of clothing, treating shadow edge as object edge, and mismatching noise.

contours. DASC performs better on clothing, possibly due to the weak relationship between its RGB and NIR appearances. Additionally, our real-time method is much faster than the others.

Ablation Study: We have tested two network structure choices: "Only RGB as DPN input" and "Averaging RGB as STN" averaging R, G and B channels as pseudo-NIR. Table 5.2 shows that overall the full method outperforms the other choices. We have also studied fully or partially removing material awareness. Table 5.2 and Figure 5.10 show that ignoring lights, glass or glossy surfaces fails on corresponding materials with small fluctuations on other materials. It means that the proposed material-specific loss functions as designed. Table 5.2 also shows that smoothing with confidence is useful.

5.8 Limitations

Although we use STN to translate RGB images to NIR images and use structural dissimilarity as loss function, the spectral difference problem cannot be completely handled. For example, in the first row of Fig. 5.11, the black clothing looks bright in NIR, leading to incorrect predictions in the disparity map. This problem could potentially be solved by converting RGB and NIR into an intermediate representation and comparing in this intermediate domain.

In addition, the results are generally blurry at object boundaries, due to the usage of smoothness losses. This problem could potentially be solved by explicitly consider occlusion in the disparity maps. Using both forward warping and backward warping [216] could be a possible approach to implement it.

5.9 Conclusion

To sum up, to show that confidence supervision from appearance location information helps fixing failures, we present a deep learning based cross-spectral stereo matching method without depth supervision. The proposed method simultaneously predicts disparity and translates an RGB image to a NIR image. A warping-based image synthesis method is adopted for obtaining supervision signal. To handle its failure on non-Lambertian regions, we identify uncertain predictions based on material awareness provided by a segmentation model. A confidence-based disparity propagation loss is introduced to fix incorrect predictions using contextual information.

Our method outperforms compared methods, especially on challenging materials, although it fails on some clothing with large spectral difference, shadow edges, and dark noisy regions (Figure 5.11). Redesigning the loss function might help address those problems. This work could possibly be extended to other spectra (SWIR, MWIR, thermal) in the future.

Chapter 6

Conclusion

In this thesis, we demonstrate that the awareness of materials provides easy-to-obtain signals for training deep networks. We propose a framework (Fig. 6.1) that can be used for different tasks to exploit material-aware supervisions. To demonstrate the effectiveness of the proposed framework, we present four applications, including translucent powder recognition, human geometry and texture reconstruction, floor appearance decomposition for object insertion, and cross-spectral stereo matching while fixing non-Lambertian regions.

In Chapter 2, we introduce an approach for fine-grained recognition of powders on complex backgrounds, to provide an example of synthetic ground truth supervision from translucent material awareness.

In Chapter 3, we demonstrate a method for recovering human texture and geometry from an RGB-D video, as an example of photometric supervision from Lambertian material assumption.

In Chapter 4, we propose a floor appearance decomposition approach for realistic object insertion, as an example of adversarial supervision from specular/sunlight appearance locations for appearance separation.

In Chapter 5, we present a cross-spectral stereo matching method for road scenes, to show that the confidence supervision from non-Lambertian appearance locations helps fix regions of failure.

The framework can be applied to but is not limited to these tasks. In Sec. 6.1, we discuss the factors that should be taken into consideration when selecting suitable supervision signals for different tasks and present several applications that can potentially be solved with the proposed framework in the future. Besides, the framework itself is not perfect and could be extended or improved. Sec. 6.2 and Sec. 6.3 discuss its limitations and possible improvement directions.

6.1 Material-Aware Supervision for Various Tasks

The proposed framework (Fig. 6.1) for material-aware supervision can potentially be applied to various tasks. We discuss the way to select suitable supervision signals and present several example potential tasks.

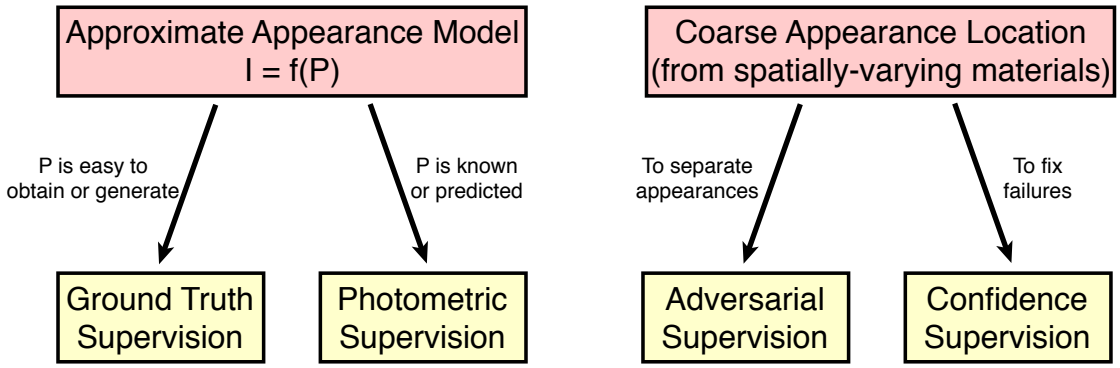


Figure 6.1: Proposed framework for exploiting material-aware supervisions. Given a task, an approximate appearance model could be build to describe the whole or part of the scene. Depending on the availability of scene properties, ground truth supervision and/or photometric supervision could be applied. If additional appearance location information is provided by spatially-varying materials, adversarial supervision and/or confidence supervision could be used for separating appearances or fixing failures.

6.1.1 Selecting Suitable Supervision Signals

To select the suitable supervision signals for a task, we should consider two factors: the target property and the information from material-awareness.

Target Property: We need to understand what the real target is. For example, the goal of depth estimation is to estimate the distance between the object and the camera in most cases, where depth is the final target. In some other scenarios, depth is just an intermediate result. For example, depth estimation might be an intermediate step for novel view synthesis, where the real target is the synthesized image rather than depth.

Information from Material-Awareness: As we discussed in previous chapters, the information from material awareness can present in the form of appearance model or appearance location. For appearance models, we need to figure out how easy it is to obtain or generate scene properties for rendering, how simple the rendering equation is and whether it is differentiable. For appearance locations, we need to check whether the appearance location information provides a coarse version of the target property or it only tells where the prediction may fail.

After checking the target property and the information we have, we can choose the suitable supervision signals based on what they can or cannot be used for.

Ground Truth Supervision: To use appearance model to generate synthesis data for ground truth supervision, the scene properties as the input to the renderer have to be easy to obtain or generate. Besides, when final target is to predict the scene property, ground truth supervision is usually a good choice because it directly provides the "correct answer" of the target. When the scene property is an intermediate result rather than the final target, ground truth supervision is also useful. But we need to be careful whether a small error in the scene property could translate to a large error in the final target.

Photometric Supervision: To use an appearance model as an differentiable renderer to provide

Task	Ground Truth	Photometric	Adversarial	Confidence
Fine-grained recognition of liquids	✓			
Thickness estimation of translucent materials		✓		
Indoor-outdoor illumination separation			✓	
RGB-NIR intrinsic image decomposition				✓
Human reconstruction with non-Lambertian accessories		✓		✓
Layout estimation for rooms with mirrors				✓
Dehazing and transmittance estimation	✓	✓		
3D reconstruction of transparent objects	✓	✓		
Powder removal		✓	✓	
Raindrop removal		✓	✓	
Shadow detection and separation		✓	✓	
Depth estimation for rainy scenes	✓			✓

Table 6.1: Possible applications of the proposed framework using different forms of supervision signals. the checked supervisions is/are the main possible one(s). Other supervisions could also be incorporated in real tasks.

photometric supervision, the scene properties for rendering have to be known or to be predicted. The rendering equation should be differentiable. Besides, photometric supervision is more suitable for tasks when the goal is to generate images using the predicted scene properties.

Adversarial Supervision: Adversarial supervision is suitable for tasks targeting at separating appearances. Usually the provided appearance location information is directly related to the target or is a coarse version of the target property.

Confidence Supervision: The target of using confidence supervision is usually for fixing failures. It is usually not used solely, because it is more like a regularization term rather than a data term. Usually the provided appearance location information is not directly related to the target, but tells us where the prediction is good and where it is not.

6.1.2 Future Applications

Below we describe several possible applications. These tasks can also be future works. Tab. 6.1 provides a list of possible applications using different forms of material-aware supervision signals. Below we describe a subset of these applications in details.

Fine-Grained Recognition of Liquids:

Liquids are a type of important materials while the fine-grained recognition of them is not well studied. Common liquids are participating media dissolved, diluted, or suspended in water. Examples include wine, saline water, milk, blood, juice, etc. Similar to powders, liquid appearance is subject to multiple factors (concentration, container, background, lighting), making it hard to capture a large real dataset covering different cases. For example, as shown in Fig. 6.2, the concentration significantly changes liquid appearance. Narasimhan *et al*. [155] presented an image formation model for participating media and a simple method for model parameter calibration. It is possible to use this model (or its approximate version) to generate a large synthetic dataset for training deep networks with ground truth supervision. This approach follows the first

Low Concentration High Concentration

Figure 6.2: Liquid appearances rendered with different concentration levels [155]. Concentration significantly affects liquid appearance, making it hard to capture a large real dataset for fine-grained recognition.

branch of the proposed framework.

Thickness Estimation of Translucent Materials:

Chapter 2 introduces a powder detection and recognition method. In some cases, people may be also interested in estimating the thickness of the detected powder sample. For example, police may need to estimate the quantity of drug samples based on captured images. As shown in Fig. 6.3, the appearance of translucent materials depends on the background appearance and the thickness value. The proposed appearance model for translucent materials can be used for photometric supervision. However, without knowing the background color, this photometric supervision is not enough for estimating thickness. Fortunately, the background color can be estimated using adversarial supervision from appearance location, where the approximate location of translucent materials could be obtained via translucent material recognition. This approach uses both photometric supervision from appearance model and adversarial supervision from appearance location.

Indoor-Outdoor Illumination Separation:

Chapter 4 introduces a method for diffuse-specular and sunlight-ambient separation. In some scenarios, people would like to separate indoor and outdoor illumination. For example, when visiting a house rental website, the customer may want to see the visual effects at night, or when all lamps are turned off. Fig. 6.4 shows that a bedroom can look significantly different during the day and at night. This task could potentially be solved by the adversarial supervision method presented in Chapter 4. We could estimate the approximate regions that are strongly affected by indoor lamps or outdoor lighting based on lamp (emissive material) and window (transparent) positions. Given such appearance location information, the GAN-based approach

Figure 6.3: Thickness affects the appearance of translucent materials.

Day Night

Figure 6.4: Appearances of a synthetic bedroom at day and night [3] are significantly different.

(a) RGB (b) NIR

Figure 6.5: Failure cases of "NIR represents shading" assumption. The top row [38] shows that the black pigment strongly absorbs NIR light. The bottom example shows that some LED lights do not emit NIR. In these two cases, NIR intensity is not consistent with RGB shading.

could be applied to predict the indoor/outdoor illumination removal effects. In this way the indoor and outdoor illumination can be separated.

RGB-NIR Intrinsic Image Decomposition:

Near infrared (NIR) images can be used for assisting intrinsic image decomposition [38], based on the assumption that NIR represents shading. However, this assumption could on certain materials or lighting conditions. For example, in Fig. 6.5, the black pigment significantly absorbs NIR light, creating dark intensities which are not consistent with shading. Besides, some artificial light sources do not emit NIR light, leading to the inconsistency between RGB and NIR shading. Fortunately, these failure cases can be identified by detecting certain materials. For example, the visible light LED could be detected by finding regions where NIR is significantly darker than RGB. The regions of failure could be regions around the LED light source. Then, similar to Chapter 5, a lower confidence value can be applied to those regions for intrinsic image decomposition. This approach uses confidence supervision from material-aware appearance location information.

Human Reconstruction with Non-Lambertian Accessories:

Chapter 3 reconstructs human by assuming human is Lambertian. However, human may wear non-Lambertian accessories like watches, jewelry, hair clips, glasses, etc. Photometric supervision based on Lambertian assumption may fail in those regions. Similar to Chapter 5, one possible method is to detect those materials and assign low confidence values to them. The proposed confidence-weighted smoothing technique could potentially be applied. This approach uses confidence supervision from material-aware appearance location information.

6.2 Framework Limitations

Material Recognition Requires Annotated Data:

When material information is not directly available as a prior, a common approach is to run semantic segmentation algorithms on the images to localize and recognize materials. However, state-of-the-art semantic segmentation methods require training data with ground truth. Although pre-trained models cover most semantic classes, some special materials are not included. Thus, additional human annotations are often required. This costs time and resources. Fortunately, the requirement of the annotation quality is usually not high, since our framework targets at coarse knowledge of materials.

Information Hard to be Represented as Appearance Model / Location:

Some information from material awareness does not present in the form of appearance models or appearance locations. For example, some shiny clothing is specular. However, without knowing the material BRDF, it is hard to build an appearance model for such clothing. The clothing wrinkle deformation information revealed by the spatially-varying intensity is also hard to be represented in the form of appearance locations. Our current framework does not incorporate such information from material awareness.

6.3 Future Improvements

Spatially-Varying Appearance Model:

In our current framework, we mainly consider a single dominated appearance model for the whole or most part of the scene. However, multiple appearance models can be built for spatially-varying materials. For example, a specular car surface appearance consists of two layers: the surface itself and the reflected scene; but the glass windshield appearance consists of three layers: the glass texture, the reflected scene, and the transmitted scene. Using different appearance models for different materials may help tasks like specular reflection removal. This is actually a combination of appearance model and appearance location information.

Adversarial Supervision for Non-Sparse Signals:

The adversarial supervision presented in Chapter 4 can only be applied to sparse signals using binary masks representing appearance locations. However, many appearances are not sparse (*e.g.* haze). One potential approach to handle non-sparse appearances is to set a threshold to binarize the signal. By using multiple thresholds, one could quantize the signal into several bins (*e.g.* according to the thickness of the haze). The local discriminator is asked to predict the correct bin for each pixel. In this way, the network might be able to learn to change the haze level.

Bibliography

[1] http://secure.axyz-design.com/. 3.5.1

[2] https://www.choosechicago.com/neighborhoods/loop/. 1.2b

[3] https://s3.amazonaws.com/cgcookie-rails/wp-uploads/2016/02/Exercise_example.jpg. 6.4

[4] http://hdrihaven.com/. 3.5.1, 4.4.1

[5] Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. Two-shot svbrdf capture for stationary materials. *ACM Transactions on Graphics (TOG)*, 34(4):1–13, 2015. 4.2

[6] Yasuhiro Akashi and Takayuki Okatani. Separation of reflection components by sparse non-negative matrix factorization. In *Asian Conference on Computer Vision (ACCV)*, pages 611–625. Springer, 2014. 4.2

[7] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Detailed human avatars from monocular video. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 98–109. IEEE, 2018. 3.2, 3.3.1

[8] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8387–8397, 2018. 3.2, 3.5.2

[9] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single rgb camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1175–1186, 2019. 3.2

[10] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 2293–2303, 2019. 1.3, 3.1, 3.2b, 3.2, 3.2, 3.3.3, 3.14, 3.5.2

[11] Nikolaos Arvanitopoulos, Radhakrishna Achanta, and Sabine Susstrunk. Single image reflection suppression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4498–4506, 2017. 4.2

[12] Amir Atapour-Abarghouei and Toby P Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2800–

2810, 2018. 1.3

[13] Dejan Azinovic, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. Inverse path tracing for joint material and lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2447–2456, 2019. 1.3

[14] Daulet Baimukashev, Alikhan Zhilisbayev, Askat Kuzdeuov, Artemiy Oleinikov, Denis Fadeyev, Zhanat Makhataeva, and Huseyin Atakan Varol. Deep learning based object recognition using physically-realistic synthetic depth scenes. *Machine Learning and Knowledge Extraction*, 1(3):883–903, 2019. 1.3

[15] Jonathan Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37:1670–1687, 08 2015. doi: 10.1109/TPAMI.2014.2377712. 4.2

[16] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019. 3.3.2, 3.3.3

[17] Jonathan T Barron and Ben Poole. The fast bilateral solver. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 617–632. Springer, 2016. 4.5.3

[18] Anil S Baslamisli, Hoang-An Le, and Theo Gevers. Cnn based learning using reflection and retinex models for intrinsic image decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6674–6683, 2018. 1.3

[19] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014. 4.1

[20] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 5420–5430, 2019. 3.2

[21] James F Blinn and Martin E Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19(10):542–547, 1976. 3.3.2

[22] Zalán Bodó. Some optical properties of luminescent powders. *Acta Physica Academiae Scientiarum Hungaricae*, 1(2):135–150, 1951. 2.2

[23] Federica Bogo, Michael J Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular rgb-d sequences. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2300–2308, 2015. 3.2

[24] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 561–578. Springer, 2016. 3.2, 3.5.1

[25] Thirimachos Bourlai, Arun Ross, Cunjian Chen, and Lawrence Hornak. A study on using mid-wave infrared images for face recognition. In *Sensing Technologies for Global Health, Military Medicine, Disaster Response, and Environmental Monitoring II; and Biometric Technology for Human Identification IX*, volume 8371, page 83711K. International

Society for Optics and Photonics, 2012. 5.1

[26] M Brown and S Susstrunk. Multi-spectral sift for scene category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 177–184, 2011. 5.1

[27] Donald A Burns and Emil W Ciurczak. *Handbook of near-infrared analysis*. CRC press, 2007. 2.2

[28] Ethan Butler, Melissa Chin, and Anders Aneman. Peripheral near-infrared spectroscopy: Methodologic aspects and a systematic review in post-cardiac surgical patients. *Journal of cardiothoracic and vascular anesthesia*, 31(4):1407–1416, 2017. 2.2

[29] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(1):172–186, 2019. 3.5.1

[30] Chein-I Chang and Su Wang. Constrained band selection for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 44(6):1575–1585, 2006. 2.2

[31] Chein-I Chang, Qian Du, Tzu-Lung Sun, and Mark LG Althouse. A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 37(6):2631–2641, 1999. 2.2, 2.3, 2.3.3

[32] Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 40–48, 2018. 1.1

[33] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Representations (ICLR)*, 2015. 5.4, 5.4.2, 5.6

[34] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2017. 4.1

[35] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 3.3.1, 3.3.2, 3.3.3

[36] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2018. 1.2, 2.6

[37] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2018.

2.6, 2.7

[38] Ziang Cheng, Yinqiang Zheng, Shaodi You, and Imari Sato. Non-local intrinsic decomposition with near-infrared priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 2521–2530, 2019. 6.5, 6.1.2

[39] Walon Wei-Chen Chiu, Ulf Blanke, and Mario Fritz. Improving the kinect by cross-modal stereo. In *British Machine Vision Conference (BMVC)*, 2011. 5.2, 5.1, 5.7, 5.9

[40] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *International Conference on Learning Representations (ICLR)*, 2016. 5.3.2

[41] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG)*, 2015. 3.1

[42] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016. 2.7

[43] Steve Cruz, Will Hutchcroft, Yuguang Li, Naji Khosravan, Ivaylo Boyadzhiev, and Sing Bing Kang. Zillow indoor dataset: Annotated floor plans with 360° panoramas and 3D room layouts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1.2e, 4.4

[44] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, 2005. 5.2

[45] Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 667–675, 2016. 5.3.3

[46] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 5.6

[47] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001. 2.5

[48] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (TOG)*, 39(5):1–38, 2020. 1.3

[49] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K Mantiuk, and Jonas Unger. Hdr image reconstruction from a single exposure using deep cnns. *ACM Transactions on Graphics (TOG)*, 36(6):1–15, 2017. 4.2

[50] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew

Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010. 5.6

[51] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3238–3247, 2017. 4.1, 4.2

[52] Chen Feng, Shaojie Zhuo, Xiaopeng Zhang, Liang Shen, and Sabine Süsstrunk. Near-infrared guided color image dehazing. In *Proceedings of the IEEE International Conference on Image Processing (ICCV)*, pages 2363–2367, 2013. 5.1

[53] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007. 2.2

[54] Yossi Gandelsman, Assaf Shocher, and Michal Irani. double-dip": Unsupervised image decomposition via coupled deep-image-priors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6, page 2, 2019. 1.3, 4.5.3

[55] 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. *ACM Transactions on Graphics (TOG)*, 36(6):1–14, 2017. 3.5.1, 3.6, 4.2, 4.4.1

[56] Marc-André Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagné, and Jean-François Lalonde. Deep parametric indoor lighting estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 7175–7183, 2019. 4.1, 4.2

[57] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–756. Springer, 2016. 1.3, 5.2

[58] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-François Lalonde. Fast spatially-varying indoor lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6908–6917, 2019. 4.2

[59] Xiurui Geng, Kang Sun, Luyan Ji, and Yongchao Zhao. A fast volume-gradient-based band selection method for hyperspectral image. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 52(11):7111–7119, 2014. 2.2

[60] Stamatios Georgoulis, Konstantinos Rematas, Tobias Ritschel, Efstratios Gavves, Mario Fritz, Luc Van Gool, and Tinne Tuytelaars. Reflectance and natural illumination from single-material specular objects using deep learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(8):1932–1947, 2017. 4.2

[61] Vasileios Gkitsas, Nikolaos Zioulis, Federico Alvarez, Dimitrios Zarpalas, and Petros Daras. Deep lighting environment map estimation from spherical panoramas. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 640–641, 2020. 4.2

[62] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 270–279, 2017. 1.3, 5.2, 5.3.1, 5.3.2, 5.3.2

[63] Artur Grigorev, Artem Sevastopolsky, Alexander Vakhitov, and Victor Lempitsky. Coordinate-based texture inpainting for pose-guided human image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12135–12144, 2019. 3.2

[64] Roger Grosse, Micah K Johnson, Edward H Adelson, and William T Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2335–2342. IEEE, 2009. 4.1

[65] Jie Guo, Zuojian Zhou, and Limin Wang. Single image highlight removal with a sparse and low-rank reflection model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–283, 2018. 4.2

[66] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 345–360. Springer, 2014. 5.1

[67] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Livecap: Real-time human performance capture from monocular video. *ACM Transactions on Graphics (TOG)*, 38(2):1–17, 2019. 3.2

[68] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Understanding real world indoor scenes with synthetic data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4077–4085, 2016. 1.3

[69] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–14. Springer, 2010. 5.1

[70] Ran He, Xiang Wu, Zhenan Sun, and Tieniu Tan. Learning invariant deep representation for nir-vis face recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2017. 5.1

[71] Harry G Hecht. The interpretation of diffuse reflectance spectra. In *Standardization in Spectrophotometry and Luminescence Measurements: Proceedings of a Workshop Seminar Held at the National Bureau of Standards, Gaithersburg, Maryland, November, November 19-20, 1975*, volume 466, page 57. US Department of Commerce, National Bureau of Standards, 1976. 2.2

[72] Yong Seok Heo, Kyong Mu Lee, and Sang Uk Lee. Robust stereo matching using adaptive normalized cross-correlation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(4):807–822, 2010. 5.2, 5.1, 5.7, 5.9

[73] Yong Seok Heo, Kyoung Mu Lee, and Sang Uk Lee. Joint depth map and color consistency estimation for stereo images with different illuminations and cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(5):1094–1106, 2012. 5.2

[74] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7312–7321, 2017. 4.2

[75] Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde. Deep sky modeling for single image outdoor lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6927–6935, 2019. 4.2

[76] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 954–960, 2018. 1.3

[77] Yinghao Huang, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V Gehler, Javier Romero, Ijaz Akhter, and Michael J Black. Towards accurate marker-less human shape and pose estimation over time. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 421–430. IEEE, 2017. 3.2

[78] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3093–3102, 2020. 3.2

[79] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 353–369. Springer, 2016. 4.5.3

[80] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2024–2032, 2019. 4.5.3

[81] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1037–1045, 2015. 5.1

[82] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 448–456. PMLR, 2015. 2.6, 4.5, 5.3.2

[83] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 2017–2025, 2015. 5.3.1

[84] Arjun Jain, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt. Moviereshape: Tracking and reshaping of humans in videos. *ACM Transactions on Graphics (TOG)*, 29(6):1–10, 2010. 3.2

[85] Wenzel Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 4.8

[86] Michael Janner, Jiajun Wu, Tejas D Kulkarni, Ilker Yildirim, and Joshua B Tenenbaum. Self-supervised intrinsic image decomposition. In *Proceedings of the International Con-*

ference on Neural Information Processing Systems (NeurIPS), pages 5938–5948, 2017. 1.3

[87] Hae-Gon Jeon, Joon-Young Lee, Sunghoon Im, Hyowon Ha, and In So Kweon. Stereo matching with color and monochrome cameras in low-light conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4086–4094, 2016. 5.2

[88] Sen Jia, Guihua Tang, Jiasong Zhu, and Qingquan Li. A novel ranking-based clustering approach for hyperspectral band selection. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 54(1):88–102, 2016. 2.2

[89] Salma Jiddi, Philippe Robert, and Eric Marchand. Detecting specular reflections and cast shadows to estimate reflectance and illumination of dynamic indoor scenes. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2020. 4.2

[90] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016. 3.3.2, 3.3.2, 3.3.3, 4.5.2

[91] Peter D Johnson. Absolute optical absorption from diffuse reflectance. *Journal of the Optical Society of America (JOSA)*, 42(12):978–981, 1952. 2.2

[92] Nathan D Kalka, Thirimachos Bourlai, Bojan Cukic, and Lawrence Hornak. Cross-spectral face recognition in heterogeneous environments: A case study on matching visible to short-wave infrared imagery. In *Proceedings of the International Joint Conference on Biometrics (IJCB)*, pages 1–8, 2011. 5.1

[93] Takeo Kanade, Peter Rander, and PJ Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia*, 4(1):34–47, 1997. 3.1

[94] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7122–7131, 2018. 3.2

[95] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics (TOG)*, 30(6):1–12, 2011. 4.1, 4.2

[96] 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 Transactions on Graphics (TOG)*, 33(3):1–15, 2014. 4.2

[97] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3907–3916, 2018. 1.3

[98] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv preprint arXiv:2006.12057*, 2020. 1.3

[99] Parneet Kaur, Kristin J Dana, and Gabriela Oana Cula. From photography to microbiology: Eigenbiome models for skin appearance. In *IEEE Conference on Computer Vision*

and Pattern Recognition Workshops, 2015. 5.1

[100] Kodo Kawase. Terahertz imaging for drug detection and large-scale integrated circuit inspection. *Optics and photonics news*, 15(10):34–39, 2004. 2.2

[101] Kodo Kawase, Yuichi Ogawa, Yuuki Watanabe, and Hiroyuki Inoue. Non-destructive terahertz imaging of illicit drugs using spectral fingerprints. *Optics express*, 11(20):2549–2554, 2003. 2.2

[102] Nirmal Keshava. A survey of spectral unmixing algorithms. *Lincoln laboratory journal*, 14(1):55–78, 2003. 2.2

[103] Hyeongwoo Kim, Hailin Jin, Sunil Hadap, and Inso Kweon. Specular reflection separation using dark channel prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1460–1467, 2013. 4.2

[104] Kihwan Kim, Jinwei Gu, Stephen Tyree, Pavlo Molchanov, Matthias Nießner, and Jan Kautz. A lightweight approach for on-the-fly reflectance estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 20–28, 2017. 4.2

[105] Seungryong Kim, Dongbo Min, Bumsub Ham, Seungchul Ryu, Minh N Do, and Kwanghoon Sohn. Dasc: Dense adaptive self-correlation descriptor for multi-modal and multi-spectral correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2103–2112, 2015. 5.2, 5.1, 5.7, 5.9

[106] Seungryong Kim, Dongbo Min, Stephen Lin, and Kwanghoon Sohn. Deep self-correlation descriptor for dense cross-modal correspondence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 679–695. Springer, 2016. 5.2

[107] Suhong Kim, Hamed RahmaniKhezri, Seyed Mohammad Nourbakhsh, and Mohamed Hefeeda. Unsupervised single-image reflection separation using perceptual deep image priors. *arXiv preprint arXiv:2009.00702*, 2020. 1.3

[108] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 3.4, 4.7

[109] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 5.6

[110] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1077–1086, 2019. 1.3

[111] Balazs Kovacs, Sean Bell, Noah Snavely, and Kavita Bala. Shading annotations in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 850–859. IEEE, 2017. 2.5

[112] Paul Kubelka and Franz Munk. An article on optics of paint layers. *Z. Tech. Phys*, 12 (593-601), 1931. 2.1, 2.2, 2.4.2

[113] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6647–6655, 2017. 5.3.1

[114] Zorah Lahner, Daniel Cremers, and Tony Tung. Deepwrinkles: Accurate and realistic clothing modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 667–684, 2018. 3.2, 3.5.1

[115] Jean-François Lalonde, Srinivasa G Narasimhan, and Alexei A Efros. What do the sun and the sky tell us about the camera? *International Journal of Computer Vision (IJCV)*, 88(1):24–51, 2010. 4.2

[116] Jean-François Lalonde, Alexei A Efros, and Srinivasa G Narasimhan. Estimating the natural illumination conditions from a single outdoor image. *International Journal of Computer Vision (IJCV)*, 98(2):123–145, 2012. 4.2

[117] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6050–6059, 2017. 3.2

[118] Eric Lengyel. *Mathematics for 3D game programming and computer graphics*. Cengage Learning, 2012. 3.3.3

[119] Anat Levin and Yair Weiss. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(9):1647–1654, 2007. 4.2

[120] José Lezama, Qiang Qiu, and Guillermo Sapiro. Not afraid of the dark: Nir-vis face recognition via cross-spectral hallucination and low-rank embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6628–6637, 2017. 5.1

[121] Chao Li, Yixiao Yang, Kun He, Stephen Lin, and John E Hopcroft. Single image reflection removal through cascaded refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3565–3574, 2020. 4.2, 4.16, 4.17, 4.8, 4.6

[122] Hao Li, Robert W Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum*, volume 27, pages 1421–1430. Wiley Online Library, 2008. 3.5.1

[123] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent feature reasoning for image inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7760–7768, 2020. 4.6.2

[124] Yu Li and Michael S Brown. Single image layer separation using relative smoothness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2752–2759, 2014. 4.2

[125] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3867–3876, 2019. 4.5.3

[126] 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. 1.3

[127] Zhengqi Li and Noah Snavely. Learning intrinsic image decomposition from watching the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9039–9048, 2018. 1.3, 2.5

[128] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2041–2050, 2018. 1.2, 1.4a, 1.4

[129] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: SVBRDF acquisition with a single mobile phone image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 4.2

[130] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2475–2484, 2020. 1.3, 4.1, 4.2

[131] Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Sai Bi, Zexiang Xu, Hong-Xing Yu, Kalyan Sunkavalli, Miloš Hašan, Ravi Ramamoorthi, and Manmohan Chandraker. Openrooms: An end-to-end open framework for photorealistic indoor scene datasets, 2020. 4.1

[132] John Lin, Mohamed El Amine Seddik, Mohamed Tamaazousti, Youssef Tamaazousti, and Adrien Bartoli. Deep multi-class adversarial specularity removal. In *Scandinavian Conference on Image Analysis*, pages 3–15. Springer, 2019. 4.1

[133] Stephen Lin, Yuanzhen Li, Sing Bing Kang, Xin Tong, and Heung-Yeung Shum. Diffuse-specular separation and depth recovery from image sequences. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 210–224. Springer, 2002. 4.6.2

[134] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 5.6

[135] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(5):978–994, 2010. 5.7

[136] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa G Narasimhan, and Jan Kautz. Neural rgb (r) d sensing: Depth and uncertainty from a video camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10986–10995, 2019. 1.2, 1.6

[137] Jie Liu, Wenjie Zhang, Yuting Tang, Jie Tang, and Gangshan Wu. Residual feature aggregation network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2359–2368, 2020. 4.5.3

[138] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4571–4580, 2019. 1.3

[139] Pengpeng Liu, Irwin King, Michael R Lyu, and Jia Xu. Flow2stereo: Effective self-supervised learning of optical flow and stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6648–6657, 2020. 1.3

[140] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 7708–7717, 2019. 1.3, 3.3.2, 3.3.3

[141] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 1.2, 4.1, 4.7

[142] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):1–16, 2015. 1.3, 3.2, 3.4

[143] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 2.6

[144] Riccardo de Lutio, Stefano D'aronco, Jan Dirk Wegner, and Konrad Schindler. Guided super-resolution as pixel-to-pixel transformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 8829–8837, 2019. 4.5.3

[145] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 30, page 3. Citeseer, 2013. 4.5

[146] Adolfo MartÍnez-UsÓMartinez-Uso, Filiberto Pla, José Martínez Sotoca, and Pedro García-Sevilla. Clustering-based hyperspectral band selection using information measures. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 45(12):4158–4171, 2007. 2.2

[147] Takashi Matsuyama and Takeshi Takai. Generation, visualization, and editing of 3d video. In *3DPVT*, 2002. 3.1

[148] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision (IJCV)*, 126(9):942–960, 2018. 1.3

[149] Haiyang Mei, Xin Yang, Yang Wang, Yuanyuan Liu, Shengfeng He, Qiang Zhang, Xiaopeng Wei, and Rynson WH Lau. Don't hit me! glass detection in real-world scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3687–3696, 2020. 1.4

[150] Abhimitra Meka, Maxim Maximov, Michael Zollhoefer, Avishek Chatterjee, Hans-Peter Seidel, Christian Richardt, and Christian Theobalt. Lime: Live intrinsic material estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6315–6324, 2018. 1.3

[151] NT Melamed. Optical properties of powders. part i. optical absorption coefficients and

the absolute value of the diffuse reflectance. part ii. properties of luminescent powders. *Journal of Applied Physics*, 34(3):560–570, 1963. 2.2

[152] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–421. Springer, 2020. 1.3

[153] Christian Mostegel, Markus Rumpler, Friedrich Fraundorfer, and Horst Bischof. Using self-contradiction to learn confidence measures in stereo vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4067–4076, 2016. 5.4

[154] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 807–814, 2010. 3.1, 4.4

[155] Srinivasa G Narasimhan, Mohit Gupta, Craig Donner, Ravi Ramamoorthi, Shree K Nayar, and Henrik Wann Jensen. Acquiring scattering properties of participating media by dilution. *ACM Transactions on Graphics (TOG)*, 25(3):1003–1012, 2006. 6.1.2, 6.2

[156] Matthew P Nelson, Shawna K Tazik, Patrick J Treado, Tiancheng Zhi, Srinivasa Narasimhan, Bernardo Pires, and Martial Hebert. Real-time, short-wave, infrared hyperspectral conforming imaging sensor for the detection of threat materials. In *Next-Generation Spectroscopic Technologies XI*, volume 10657, page 106570U. International Society for Optics and Photonics, 2018. 2.2, 2.3.1

[157] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136. IEEE, 2011. 3.2

[158] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 343–352, 2015. 3.2, 3.3.3

[159] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 4531–4540, 2019. 3.2

[160] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 484–494. IEEE, 2018. 3.2

[161] Jeong Joon Park, Aleksander Holynski, and Steven M Seitz. Seeing the world in a bag of chips. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1417–1427, 2020. 4.2

[162] Celio Pasquini. Near infrared spectroscopy: fundamentals, practical aspects and analytical

applications. *Journal of the Brazilian Chemical Society*, 14:198–219, 2003. 2.2

[163] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Proceedings of the International Conference on Neural Information Processing Systems Workshops*, 2017. 5.4.1, 5.6

[164] Swarnajyoti Patra, Prahlad Modi, and Lorenzo Bruzzone. Hyperspectral band selection based on rough set. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 53 (10):5495–5503, 2015. 2.2, 2.3, 2.3.3

[165] Massimo Piccardi. Background subtraction techniques: a review. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 4, pages 3099–3104. IEEE, 2004. 3.3.1, 3.3.2, 3.3.3

[166] Peter Pinggera12, Toby Breckon, and Horst Bischof. On cross-spectral stereo matching using dense gradient features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2012. 5.2

[167] Giovanni Pintore, Marco Agus, and Enrico Gobbetti. Atlantanet: Inferring the 3d indoor layout from a single 360 image beyond the manhattan world assumption. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 432–448. Springer, 2020. 4.1

[168] Thomas Porter and Tom Duff. Compositing digital images. In *ACM Siggraph Computer Graphics*, volume 18, pages 253–259. ACM, 1984. 2.2

[169] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500, 2001. 3.1, 3.3, 3.3.1, 3.3.1, 3.3.3

[170] N Dinesh Reddy, Minh Vo, and Srinivasa G Narasimhan. Occlusion-net: 2d/3d occluded keypoint localization using graph networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7326–7335, 2019. 1.3

[171] Helge Rhodin, Nadia Robertini, Dan Casas, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt. General automatic human shape and motion capture using volumetric contour cues. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 509–526. Springer, 2016. 3.2

[172] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3.4, 4.7

[173] Dominic Rüfenacht, Clément Fredembach, and Sabine Süsstrunk. Automatic and accurate shadow detection using near-infrared information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(8):1672–1678, 2013. 5.1

[174] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 2304–2314, 2019. 1.4.2, 3.2, 3.6

[175] Matan Sela, Elad Richardson, and Ron Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1576–1585, 2017. 3.2

[176] S. Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, D. Jacobs, and J. Kautz. Neural inverse rendering of an indoor scene from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8597–8606, 2019. 4.2

[177] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. Sfsnet: Learning shape, reflectance and illuminance of facesin the wild'. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6296–6305, 2018. 4.2

[178] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. Sfsnet: Learning shape, reflectance and illuminance of facesin the wild'. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6296–6305, 2018. 3.2, 3.6

[179] Steven A. Shafer. Using color to separate reflection components. *Color Research & Application*, 10(4):210–218, 1985. doi: https://doi.org/10.1002/col.5080100409. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/col.5080100409`. 4.6.2

[180] Steven A Shafer. Using color to separate reflection components. *Color Research & Application*, 10(4):210–218, 1985. 4.2

[181] Lavanya Sharan, Ruth Rosenholtz, and Edward Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, 9(8):784–784, 2009. 1.1

[182] Hui-Liang Shen and Zhi-Huan Zheng. Real-time highlight removal using intensity ratio. *Applied optics*, 52(19):4483–4493, 2013. 4.1

[183] Xiaoyong Shen, Li Xu, Qi Zhang, and Jiaya Jia. Multi-modal and multi-spectral registration for natural images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 309–324. Springer, 2014. 5.2

[184] YC Shen, a T Lo, PF Taday, BE Cole, WR Tribe, and MC Kemp. Detection and identification of explosives using terahertz pulsed spectroscopic imaging. *Applied Physics Letters*, 86(24):241116, 2005. 2.2

[185] Jian Shi, Yue Dong, Hao Su, and Stella X Yu. Learning non-lambertian object intrinsics across shapenet categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1694, 2017. 4.2

[186] YiChang Shih, Dilip Krishnan, Fredo Durand, and William T Freeman. Reflection removal using ghosting cues. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3193–3201, 2015. 4.2

[187] Yael Shor and Dani Lischinski. The shadow meets the mask: Pyramid-based shadow removal. *Comput. Graph. Forum*, 27(2):577–586, 2008. doi: 10.1111/j.1467-8659.2008.01155.x. URL `https://doi.org/10.1111/j.1467-8659.2008.01155.x`. 4.6.2

[188] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5541–5550, 2017. 4.2

[189] Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, et al. Textured neural avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2387–2397, 2019. 1.3, 3.1, 3.2b, 3.2, 3.2, 3.3.2, 3.8c, 3.8, 3.3.2, 3.5.2

[190] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 746–760. Springer, 2012. 1.2, 1.5, 2.5, 2.11

[191] EL Simmons. An equation relating the diffuse reflectance of weakly absorbing powdered samples to the fundamental optical parameters. *Optica Acta: International Journal of Optics*, 18(1):59–68, 1971. 2.2

[192] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 3.3.2, 3.3.3, 3.4

[193] Gowri Somanath and Daniel Kurz. HDR environment map estimation for real-time augmented reality. *arXiv preprint arXiv:2011.10687*, 2020. 4.2

[194] Shuran Song and Jianxiong Xiao. Tracking revisited using rgbd camera: Unified benchmark and baselines. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 233–240, 2013. 5.1

[195] Olga Sorkine. Differential representations for mesh processing. In *CGF*, 2006. 3.3.2

[196] Jonathan Stets, Zhengqin Li, Jeppe Revall Frisvad, and Manmohan Chandraker. Single-shot analysis of refractive shape using convolutional neural networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 995–1003. IEEE, 2019. 1.3

[197] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1047–1056, 2019. 4.1

[198] Robby T Tan and Katsushi Ikeuchi. Separating reflection components of textured surfaces using a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(2):178–193, 2005. 4.2

[199] Marshall F Tappen, William T Freeman, and Edward H Adelson. Recovering intrinsic images from a single image. *IEEE transactions on pattern analysis and machine intelligence*, 27(9):1459–1472, 2005. 4.2

[200] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning

for monocular reconstruction at over 250 hz. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2549–2559, 2018. 3.2

[201] Ayush Tewari, Florian Bernard, Pablo Garrido, Gaurav Bharaj, Mohamed Elgharib, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Fml: Face model learning from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10812–10822, 2019. 3.2

[202] Alessio Tonioni, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Unsupervised adaptation for deep stereo. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1605–1613, 2017. 5.2, 5.4.1

[203] Maxime Tremblay, Shirsendu Sukanta Halder, Raoul de Charette, and Jean-François Lalonde. Rain rendering for evaluating and improving robustness to bad weather. *International Journal of Computer Vision (IJCV)*, pages 1–20, 2020. 1.3

[204] Tatsumi Uezato, Danfeng Hong, Naoto Yokoya, and Wei He. Guided deep decoder: Unsupervised image pair fusion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102. Springer, 2020. 4.9, 4.8

[205] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 3.1, 4.4

[206] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9446–9454, 2018. 1.3, 3.1, 3.3.2

[207] Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz, and Wojciech Matusik. Dynamic shape capture using multi-view photometric stereo. *ACM Transactions on Graphics (TOG)*, 28(5), 2009. 3.1

[208] Minh Vo, Srinivasa G. Narasimhan, and Yaser Sheikh. Spatiotemporal bundle adjustment for dynamic 3d reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3.3.3, 3.3.3

[209] Aaron Walsman, Weilin Wan, Tanner Schmidt, and Dieter Fox. Dynamic high resolution deformable articulated tracking. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 38–47. IEEE, 2017. 3.1, 3.5.1

[210] Renjie Wan, Boxin Shi, Tan Ah Hwee, and Alex C Kot. Depth of field guided reflection removal. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 21–25. IEEE, 2016. 4.2

[211] Renjie Wan, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, and Alex C Kot. Crrn: Multi-scale guided concurrent reflection removal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4777–4785, 2018. 4.2

[212] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 4.1, 4.4, 4.4.2, 4.5.1, 4.11c

[213] Lin Wang, Chein-I Chang, Li-Chien Lee, Yulei Wang, Bai Xue, Meiping Song, Chuanyan

Yu, and Sen Li. Band subset selection for anomaly detection in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 55(9):4887–4898, 2017. 2.2

[214] Rui Wang, Nan Yang, Jörg Stückler, and Daniel Cremers. Directshape: Direct photometric alignment of shape priors for visual vehicle pose and shape estimation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11067–11073. IEEE, 2020. 1.3

[215] Shaohong Wang, Bradley Ferguson, Carmen Mannella, Derek Abbott, and X-C Zhang. Powder detection using thz imaging. In *Lasers and Electro-Optics, 2002. CLEO'02. Technical Digest. Summaries of Papers Presented at the*, pages 131–vol. IEEE, 2002. 2.2

[216] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4884–4893, 2018. 1.4.2, 5.8

[217] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003. 4.8

[218] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 13(4):600–612, 2004. 3.3.3, 3.5.2, 3.5.2, 5.3.2

[219] Henrique Weber, Donald Prévost, and Jean-François Lalonde. Learning to estimate indoor lighting from 3D objects. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 199–207. IEEE, 2018. 4.2

[220] Kaixuan Wei, Jiaolong Yang, Ying Fu, David Wipf, and Hua Huang. Single image reflection removal exploiting misaligned training data and network enhancements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8178–8187, 2019. 4.2, 4.16, 4.17, 4.8, 4.6

[221] Tobias Wilken, Gaspare Lo Curto, Rafael A Probst, Tilo Steinmetz, Antonio Manescau, Luca Pasquini, Jonay I González Hernández, Rafael Rebolo, Theodor W Hänsch, Thomas Udem, et al. A spectrograph for exoplanet observations calibrated at the centimetre-per-second level. *Nature*, 485(7400):611–614, 2012. 2.2

[222] Pengxiang Wu, Siheng Chen, and Dimitris N Metaxas. Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11385–11395, 2020. 1.3

[223] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10, 2020. 1.3

[224] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 2.6

[225] Zhongqi Wu, Chuanqing Zhuang, Jian Shi, Jun Xiao, and Jianwei Guo. Deep specular

highlight removal for single real-world image. In *SIGGRAPH Asia 2020 Posters*, pages 1–2, 2020. 4.2

[226] Dan Xu, Wanli Ouyang, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Learning cross-modal deep representations for robust pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5363–5371, 2017. 5.1

[227] Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. Monoperfcap: Human performance capture from monocular video. *ACM Transactions on Graphics (TOG)*, 37(2):1–15, 2018. 3.2

[228] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 7760–7770, 2019. 3.2

[229] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T Freeman, and Ce Liu. Lasr: Learning articulated shape reconstruction from a monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1.3

[230] Jie Yang, Dong Gong, Lingqiao Liu, and Qinfeng Shi. Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 654–669, 2018. 1.2, 1.4b, 1.4

[231] Qingxiong Yang, Shengnan Wang, and Narendra Ahuja. Real-time specular highlight removal using bilateral filtering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–100. Springer, 2010. 4.2, 4.16, 4.17, 4.8, 4.6

[232] Xin Yang, Haiyang Mei, Ke Xu, Xiaopeng Wei, Baocai Yin, and Rynson WH Lau. Where is my mirror? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8809–8818, 2019. 1.2c, 1.2, 1.5

[233] Renjiao Yi, Chenyang Zhu, Ping Tan, and Stephen Lin. Faces as lighting probes via unsupervised deep highlight extraction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 317–333, 2018. 3.6

[234] Tao Yu, Kaiwen Guo, Feng Xu, Yuan Dong, Zhaoqi Su, Jianhui Zhao, Jianguo Li, Qionghai Dai, and Yebin Liu. Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 910–919, 2017. 3.2

[235] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7287–7296, 2018. 3.2

[236] Tao Yu, Zerong Zheng, Yuan Zhong, Jianhui Zhao, Qionghai Dai, Gerard Pons-Moll, and Yebin Liu. Simulcap: Single-view human performance capture with cloth simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5504–5514, 2019. 1.4.2, 3.2, 3.6

[237] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 215–224, 1999. 4.1, 4.2, 4.5.1, 4.6.2

[238] Edward Zhang, Michael F Cohen, and Brian Curless. Emptying, refurnishing, and relighting indoor spaces. *ACM Transactions on Graphics (TOG)*, 35(6):1–14, 2016. 4.2

[239] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 4.5.2

[240] Jinsong Zhang, Kalyan Sunkavalli, Yannick Hold-Geoffroy, Sunil Hadap, Jonathan Eisenman, and Jean-François Lalonde. All-weather deep outdoor lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10158–10166, 2019. 4.2

[241] Xiaopeng Zhang, Terence Sim, and Xiaoping Miao. Enhancing photographs with near infra-red images. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 5.1

[242] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4786–4794, 2018. 4.2, 4.5.2

[243] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 519–535. Springer, 2020. 1.3

[244] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. Deephuman: 3d human reconstruction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 7739–7749, 2019. 3.2, 3.14, 3.5.2

[245] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 633–641, 2017. 4.4.2

[246] Chao Zhou, Hong Zhang, Xiaoyong Shen, and Jiaya Jia. Unsupervised learning of stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1567–1575, 2017. 1.3, 5.2, 5.4, 5.4.1

[247] Shizhe Zhou, Hongbo Fu, Ligang Liu, Daniel Cohen-Or, and Xiaoguang Han. Parametric reshaping of human bodies in images. *ACM Transactions on Graphics (TOG)*, 29(4):1–10, 2010. 3.2

[248] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1851–1858, 2017. 1.3, 5.2, 5.3.1

[249] Hao Zhu, Xinxin Zuo, Sen Wang, Xun Cao, and Ruigang Yang. Detailed human shape estimation from a single image by hierarchical mesh deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4491–4500, 2019. 3.2, 3.14, 3.5.2

[250] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3D room layout from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2051–2059, 2018. 4.1

[251] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6365–6373, 2017. 1.3