ABSTRACT

Title of dissertation: Constraints and Priors for Inverse Rendering from Limited Observations

Soumyadip Sengupta, Doctor of Philosophy, 2019

Dissertation directed by: David Jacobs
Department of Electrical and Computer Engineering

Inverse Rendering deals with recovering the underlying intrinsic components of an image, i.e. geometry, reflectance, illumination and the camera with which the image was captured. Inferring these intrinsic components of an image is a fundamental problem in Computer Vision. Solving Inverse Rendering unlocks a host of real world applications in Augmented and Virtual Reality, Robotics, Computational Photography and gaming. Researchers have made significant progress in solving Inverse Rendering from a large number of images of an object or a scene under relatively constrained settings. However, most real life applications rely on a single or a small number of images captured in an unconstrained environment. Thus in this thesis we explore Inverse Rendering under limited observations from unconstrained images.

We consider two different approaches for solving Inverse Rendering under limited observations. First, we consider learning data-driven priors that can be used for Inverse Rendering from a single image. Our goal is to jointly learn all intrinsic components of an image, such that we can recombine them and train on unlabeled real data using self-supervised reconstruction loss. A key component that enables self-supervision is a differentiable rendering module that can combine the intrinsic components to accurately regenerate the image. We show how such a self-supervised reconstruction

loss can be used for Inverse Rendering of faces. While this is relatively straightforward for faces, complex appearance effects (e.g. inter-reflections, cast-shadows and near-field lighting) present in a scene can't be captured with a differentiable rendering module. Thus we also propose a deep CNN based differentiable rendering module (Residual Appearance Renderer) that can capture these complex appearance effects and enable self-supervised learning. Another contribution is a novel Inverse Rendering architecture, SfSNet, that performs Inverse Rendering for faces and scenes.

Second, we consider enforcing low-rank multi-view constraints in an optimization framework to enable Inverse Rendering from a few images. To this end, we propose a novel multi-view rank constraint that connects all cameras capturing all the images in a scene and is enforced to ensure accurate camera recovery. We also jointly enforce a low-rank constraint and remove ambiguity to perform accurate Uncalibrated Photometric Stereo from a few images. In these problems, we formulate a constrained low-rank optimization problem in the presence of noisy estimates and missing data. Our proposed optimization framework can handle this non-convex optimization using Alternate Direction Method of Multipliers (ADMM). Given a few images, enforcing low-rank constraints significantly improves Inverse Rendering.

Constraints and Priors for Inverse Rendering from Limited Observations

by

Soumyadip Sengupta

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:

Professor David Jacobs, Chair/Advisor

Professor Rama Chellappa

Professor Behtash Babadi

Professor Thomas Goldstein

Professor Abhinav Shrivastava

Dr. Carlos Castillo

To Ma and Baba,
who always encouraged me to achieve my dreams.

To David, my advisor,
who steered me towards my dreams.

Acknowledgments

I would like to start by thanking all of my family members, mentors, peers and friends whose constant encouragement and support helped me achieve my dream that resulted in this thesis. I am grateful to all of these people that I encountered during this journey.

First and foremost, I would like to thank my advisor Prof. David Jacobs, who nurtured and helped me mature as a researcher. When I first joined UMD back in Fall 2013, fresh out of undergrad, I was enthusiastic and yet lost in the big sea of research. I was looking for an advisor who can guide my enthusiasm to mold me as a researcher. In the last five and half years, David played this critical role in my academic life by steering me towards my goal. In the beginning, he was patient and extended his helping hand to let me figure out what problem excites me most. During my PhD, he was very accommodating to let me change my research direction. His emphasis on quality over quantity, on the importance of finding the right problem and his attitude towards failures in research impacted me significantly during my PhD. David always encouraged me to spend some time to investigate the reasons of failures before moving on, which I find extremely valuable. I will miss David's extensive criticism of my ideas, which has always helped me to come up with better ones, even though I hated it in the beginning.

Through out my PhD, I was fortunate to have amazing mentors who inspired and improved me as a researcher. I am indebted to Prof. Ronen Basri for mentoring me during an year spent at Weizmann Institute, Israel and the rest of my PhD. I learned a great deal of mathematical insights and rigor under the guidance of Prof. Basri. I am thankful to Dr. Jinwei Gu, my mentor at NVIDIA Research, who provided me with an opportunity to work on a challenging problem and constantly motivated me. Beyond this internship, Dr. Gu inspired me in doing fundamental research which can create large impact in real life. I am also thankful to my former lab mates Dr. Angjoo Kanazawa, Dr. Arijit Biswas and Dr. Carlos Castillo who always encouraged and helped me. I am especially thankful to Angjoo who motivated me during the tough times and is still helping me in the next steps of my career. I am also grateful to Prof. Tom Goldstein, from whom I learned about optimization and machine learning and who inspired me to lift weights.

I thank my committee members: Prof. Rama Chellappa, Prof. Behtash Babadi, Prof. Thomas Goldstein, Prof. Abhinav Shrivastava, and Dr. Carlos Castillo for insightful comments, suggestions and questions. Special thanks to Prof. Chellappa and Prof. Goldstein for their amazing courses on Pattern Recognition and Optimization. I am grateful to the support from the wonderful staff of UMIACS and ECE, especially Janice Perrone, Arlene E Schenk and Melanie Prange.

I am thankful for all the support and friendship I received from my amazing lab mates: Arijit Biswas, Angjoo Kanazawa, Abhishek Sharma, Jin Sun, Hao Zhou, Abhay Yadav, Ryen Krusinga, Koutilya PNVR, Daniel Lichy as well as other members of UMD: Rajiv Ranjan, Joshua Gleeson, Amit Kumar, Pallabi Ghosh.

My mentors from my undergraduate program: Prof. Ananda Shankar Chowdhury, Prof. Swagatam Das, and my mentors from my undergraduate internship sponsored by DAAD: Prof. Günter Rudolph, Dr. Mike Preuss, deserve a special mention for encouraging me in research and to pursue a PhD. I would like to thank Linjie Luo and Chen Cao for a great internship at Snapchat and Jinwei Gu, Kihwan Kim, Guilin Liu, Jan Kautz for an amazing internship at NVIDIA.

This thesis would not have been possible without the support and affection I received from my family. Special thanks to my parents, who constantly encouraged me to follow my dreams and excel in what I love doing. Without their inspiration, love and support this journey would have been extremely difficult. Thanks to my aunts and uncles, my cousins: Agniv, Riya, Souvik, Suhita, Suman. Special thanks to Agniv, who was also my roommate for the past four years and tolerated me for not doing the chores.

A great source of encouragement, motivation and sometimes pure fun during the tough times in PhD are my friends. Abhishek, Agniv, Ahana, Biswadip, Dipankar, Manaswi, Rishov, Sankha,

Shawon, Siddarth, Soham, Sohini, Sunandita, Udit, Upamanyu and many others, thanks a lot for making these five and half years stay at College Park memorable. Special thanks to the great friendship with Biswadip, someone I can always count on. Thanks to my childhood buddies Saurav, Tridiv and Deblin who helped me to stay positive with countless road-trips across America. Also to Nasir, Mridul, Sayani, Deepan and Atreyee for your friendship and support during the difficult times of PhD.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Objective

Figure 1-1: Real objects have complex reflectance properties. In the two left images the shiny stove and counter reflect strong highlights whose appearance depends on the viewpoint. In the third image, the chair's and table's shiny material strongly affects its appearance. On the right we see the wall's appearance affected by varying distance to the light source.

Images taken of our everyday scenes with any camera consist of multiple different objects with their unique shapes and materials, and the illumination conditions. Computer Graphics deals with replicating these real world effects by modeling shape, material properties, illumination conditions and the rendering process leading to the image generation. In Computer Vision, we are interested

in solving the inverse problem, i.e. given an image, we want to obtain the underlying components that produce this image. This is an ambitious goal and an ill-posed problem because we aim to recover all underlying physical properties from an image. In this thesis we make a significant step in achieving this ambitious goal.

In Inverse Rendering, given an image, we decompose it into its underlying components related to geometry, reflectance, illumination and the camera parameters. This is a fundamental problem in Computer Vision as understanding and recovering the intrinsic components of an image unlocks the potential to solve a wide range of applications in AR/VR, Robotics, Computational Photography etc. Inverse Rendering, refers to a broader class of problems, which have been classically studied in Computer Vision over past decades. For example, in Structure from Motion, given multiple images of a scene, the goal is to recover the scene geometry and the camera parameters, focusing only on the structural components of Inverse Rendering. In Photometric Stereo, given multiple images of an object captured under varying illumination conditions, the aim is to recover the shape and reflectance of the object. In presence of a single image, the above problem is also known as Shape from Shading. All these sub-problems are solved with a final goal of solving the Inverse Rendering problem from unconstrained collections of images.

Given a large number of images of a scene, researchers have shown great promise in solving many different sub-problems related to Inverse Rendering. A significant improvement came in Structure from Motion to reconstruct a real world 3D scene from unconstrained image collections [143, 2, 59]. Another success story is the facial 3D reconstruction from a video [149] or from unconstrained internet photos [130]. However obtaining large number of images of an object or a scene during inference significantly limits applications in solving many real life problems where the users can only capture limited images. Thus, it is important to study Inverse Rendering given a single or a few images, which is more realistic and challenging.

In this thesis, we study the problem of Inverse Rendering under limited observations. In presence of a large number of images, different sub-problems of Inverse Rendering have been solved in an optimization framework by enforcing the geometric and photometric constraints or by designing the hand-crafted priors. In presence of a few images or a single image, this optimization framework will be under-constrained. Thus we aim to seek and enforce additional constraints and learn data-driven priors over intrinsic components that can allow us to perform inverse rendering from limited observations.

1.2 Impact

Inverse Rendering deals with inferring the underlying components of an image, which unlocks a host of real world applications that rely on one or more of them. Inverse Rendering has huge potential applications in Robotics, AR/VR, Gaming, Computational Photography and many other related areas.

Recovering only the geometric components of a scene, popularly known as Structure from Motion, has large potential applications in Robotics. A central problem in Robotics is navigating in an indoor or outdoor environment using camera(s). In fact this problem is popularly known as Simultaneous Localization and Mapping (SLAM) [80], where the aim is to reconstruct the world around the agent while also localizing the agent in the world. This problem is also important in context of self-driving cars [58]. Due to the need for real-time performance in SLAM and limited processing capabilities of the agent, many approaches often rely on using limited observations. These approaches often rely on learning based methods [185, 164] to recover the geometry and camera parameters.

Virtual and Augmented Reality is a major application where Inverse Rendering plays an important role. AR and VR have numerous applications in education, workforce training, tele-presence,

entertainment, gaming, and manufacturing. One of the chief challenges in improved AR/VR is the accurate construction of the 3D world in terms of the geometry, material reflectance and illumination condition. Understanding the geometry of the scene will allow us to insert virtual objects appropriately in the scene, which leads to many interesting Augmented Reality applications like furniture shopping [145] or clothes try-on [175]. Similarly, inferring the reflectance property of an object in the scene will allow us to perform different editing applications like changing the texture, material shininess and other attributes [98] which can be useful in Computational Photography, Computer Graphics and Virtual Shopping. Understanding the illumination in the scene allows us to insert an object in the scene with realistic lighting [75], or change the illumination in the scene to create interesting visual effects. Recovering illumination also impacts Digital Forensics [72] by checking whether the lighting in the scene matches that of the object of interest.

Most of the applications described above in AR, VR and computational photography are not possible if the users are required to collect a large number of images of a scene. Applications like Virtual Try-on, Virtual Shopping, Digital Forensics, Editing Photographs demand a single image or a few images. Thus in this thesis we aim to study Inverse Rendering given limited observations. Along with the practical impacts, studying Inverse Rendering from a single or a few images is an extremely ambitious goal and poses major fundamental challenges.

1.3 Challenges

Given a large number of images and a relatively constrained settings, researchers have solved for certain sub-problems related to Inverse Rendering in an optimization framework. A large number of images gives rise to a highly over-constrained problem and constrained settings allows us to form simplistic assumptions about the real world. However in presence of a single or a few images, Inverse Rendering becomes severely ill-posed and under-constrained. Also simplistic assumptions do

not hold true for unconstrained or in-the-wild images, which require the need of more complicated models.

One direction explored in this thesis to solve Inverse Rendering is to learn data-driven priors over intrinsic components. This means training a Convolutional Neural Network (CNN) to predict geometry, reflectance and lighting from a single image. Training a CNN is effective in presence of a large amount of labeled training data for predicting geometry, reflectance and lighting. It is possible to create a highly realistic labeled synthetic dataset by using the most advanced Computer Graphics techniques. However, the networks trained on synthetic data fails to generalize well when tested on real data. This is because, even with the best graphics techniques, it is tremendously difficult and expensive to capture the variations of shapes, material properties and illumination conditions existing in real world. Thus, it is very important to train on real data. But it is also extremely difficult to collect a large scale labeled real world dataset, specially with ground-truth reflectance and illumination. Thus in this thesis we aim to address the lack of labeled large scale real world datasets.

In Computer Graphics, researchers have improved the representation of geometry, reflectance of real world objects and the rendering process for image generation to produce as realistic images as possible. When solving for Inverse Rendering, we face an important question: *What is a good representation?*. A good representation of intrinsic components and the rendering or image generation process should ideally be mathematically simple to be employed in an inverse framework and yet representative enough to capture the realistic effects. In our thesis we have explored different choices of representations depending on the object of interests.

In presence of a large number of images, researchers have used geometric and photometric multi-view constraints that can relate multiple images of an object or a scene. In presence of a small number of images, enforcing these constraints become crucial in inferring accurate and con-

sistent intrinsics as the problem is less over-constrained. Thus a central challenge in presence of a small number of images is to consider and strongly enforce all important multi-view constraints to ensure consistent recovery. Many of these constraints manifest as low-rank optimization problem. Low-rank constraints are non-convex, and along with missing data and noisy estimations pose a significant challenge to be solved in an optimization framework. In our thesis, we show how to seek and enforce such constraints and solve them efficiently using a constrained low rank optimization.

1.4 Our Proposed Solution

In this thesis, we aim to address these above challenges in a principled way. We consider inverse rendering given a single image and a few images. In presence of a few images, we explore the role of low-rank constraints in an optimization framework. For single image based inverse rendering, we focus on the challenges of learning data-driven priors. In both cases, we explore the choices for representing the intrinsic components and image rendering model.

1.4.1 Self-supervised Learning from Real Images

The main challenge in Inverse Rendering from a single image is the lack of labeled real world datasets and the inability to generalize when trained on synthetic data. The goal is to reduce the domain gap between synthetic and real data.

One way to reduce the domain gap is to use advanced Computer Graphics techniques to generate synthetic data. In our work on Inverse Rendering of a Scene in Chapter 4, we use physically based ray-tracing with Mitsuba [68] to generate synthetic images. However even the most advanced graphics engines are limited in the variability of objects and lighting conditions they can capture under realistic processing capabilities. Thus an obvious answer to reduce the domain gap is to train on real data.

In absence of ground-truth labels, our goal is to train on real images using self-supervised photometric reconstruction loss. In Inverse Rendering, we decompose an image into its intrinsic components, geometry, reflectance and illumination, using a deep CNN. Then we can combine them using a previously defined differentiable image rendering module to reconstruct the original image. This reconstructed image along with the original image will create a photometric reconstruction loss that will be used for self-supervision.

However there are three key issues in using self-supervised reconstruction loss. The first issue is the ambiguity in decomposition. In absence of any supervision over the intrinsic components, the network can simply learn to produce a degenerate solution as intrinsic components such that the reconstructed image perfectly matches the original image. For example, the network can learn to simply reproduce the image as reflectance with flat geometry and uniform illumination, such that the reconstructed image perfectly matches the original image. To avoid this space of ambiguity, we take advantage of synthetic data. We propose a novel teacher-student learning based framework 'SfS-supervision' in Chapter 3 for resolving ambiguities in decomposition. This is used for both Inverse Rendering of faces and scenes.

Our proposed 'SfS-supervision' consists of a teacher network that is first trained on synthetic data with ground-truth labels and learns a strong data-driven prior over the intrinsic components. Then for real training images, we use the inferred intrinsic components from this teacher network as 'pseudo-supervision' for the student network to be trained on real data. We train the student network on real data using 'pseudo-supervision' over intrinsic components and self-supervised reconstruction loss.

The second issue is the choice of the network architecture for inverse rendering. Previous works, that mostly focus on obtaining only one of the components of inverse rendering, use a CNN architecture based on U-Net [129]. However we aim to decompose an image into all its intrinsic com-

ponents. Since geometry and reflectance is in the same domain as the original image, it is easier to learn a CNN that can transform an image into a geometric representation like depth or normal map, or a reflectance representation like albedo map. However the illumination models are independent of image domain. For example, representing illumination as spherical harmonics means regressing a 27 dimensional vector from an image. This is a more difficult task to learn than regressing geometry or reflectance which is more aligned to the original image domain. Thus we propose an inverse rendering network, SfSNet in Section 3, that conditions lighting estimation on geometry and reflectance inference. This greatly facilitates learning as evident from its application in faces and scenes.

The last issue, is the choice of the differentiable image rendering module that can be used for reconstructing the original image from its intrinsic components. This rendering module is ideally a non-learnable physics based renderer that can recreate an image accurately from its intrinsic components. In case of faces, simplistic assumptions allow us to use a differentiable lambertian renderer. However for scenes, the actual rendering process is more complicated and can only be captured by a non-differentiable recursive process called ray-tracing. As ray-tracing can not be employed in a learning based framework we aim to represent the differentiable rendering module using a non-learnable physics based model that only captures direct illumination effects and a learnable renderer, Residual Appearance Renderer (RAR), that captures complex appearance effects. In Chapter 4 we discuss the role of RAR in facilitating self-supervised learning for real world images of a scene.

1.4.2 Low-rank Constraints

Multi-view geometric and photometric constraints often manifest themselves as a constrained low-rank optimization problem. Constrained low-rank optimization is non-convex and is particularly difficult to solve in presence of noisy estimates with outliers and missing data. In Part II we focus

on enforcing these low rank constraints in an optimization framework for different sub-problems of Inverse Rendering.

In case of Structure from Motion in Chapter 7, given a few images of a scene, we aim to solve for the camera parameters and subsequently reconstruct the 3D scene. We propose a new multi-view rank constraint that relates all the camera parameters of the scene. Enforcing this rank constraint ensures accurate recovery of the camera matrices given noisy estimates with outliers. We formulate a constrained low-rank optimization problem that is solved using Augmented Direction Method of Multipliers (ADMM). We show that our rank constraint improves camera parameter estimation specially in presence of a few images.

In case of Uncalibrated Photometric Stereo in Chapter 8, given multiple images of a lambertian object captured under distant, direct, point-source illumination, we aim to reconstruct the 3D object. The mathematical formulation of photometric stereo consists of a low-rank constraint that connects the images in multiple views. In presence of large number of images, traditional approaches first find a low-rank approximation and then resolve a space of convex ambiguity. However, this sequential approach of enforcing constraints produces more errors, specially in presence of a few images as the problem is less over-constrained. Thus our goal is to jointly enforce the low-rank constraint and convex ambiguity constraint in an optimization framework. We also use ADMM framework to solve this non-convex constrained low-rank optimization problem.

1.4.3 Representations for Inverse Rendering

An important question while solving any Inverse Rendering problem is the choice of representing geometry, reflectance and lighting, and the image rendering model. While some models are simple and easy to use in an inverse framework, they fail to work well for an unconstrained scenario. In our thesis we represent geometry as surface normals or depth map as it suffices to capture the geometry

and can be easily employed in the image rendering process.

In our work on Uncalibrated Photometric Stereo, we consider the most constrained setup with an object with lambertian reflectance being illuminated with a distant point source light assuming only direct illumination. We also assume the camera projection to be orthographic as the camera is assumed to be far-away from the object of interest. Although it makes mathematical treatments of the problem relatively easier, it severely limits its application in unconstrained images. For Structure from Motion, we only care about the geometric camera model and ignore the photometric image rendering model. Geometry is represented by a point-cloud and perspective projection is assumed for the camera model. This setup is more general and is applied for 3D reconstruction of unconstrained images in real life.

In our work on learning based Inverse Rendering for faces, we consider lambertian reflectance and represent it using albedo. Illumination is considered to be distant and direct and represented by 27 dimensional spherical harmonics, 9 for each of the RGB channels. This is a standard photometric model for faces, and previous works have shown success in using these representations and model. The main advantage stems from the fact that the image generation model is differentiable and can be employed in a deep CNNs to regenerate back the image and train with self-supervision on real data.

In case of Inverse Rendering for scenes, the representation and choices used for faces are not enough. General objects are not lambertian and hence we need to also represent the shininess or specularity of the material. An obvious choice is representing specularity using the Phong or Ward model. However, we found that it is difficult to recover such detailed parametrization of shininess just from a single image of a scene. Thus we represent material shininess as ‘matte’, ‘glossy’ and ‘semi-glossy’ and treat it as a segmentation problem. Illumination is represented as an environment map. The major difference between face and scene stems from the image rendering

model. In a scene we observe complex appearance effects like, inter-reflections, cast shadows, near-field illumination, and realistic shading which needs to be modeled in order to regenerate the image from its intrinsic components and train with self-supervision. In Computer Graphics this can be modeled using a recursive non-differentiable procedure termed ray-tracing, which can not be employed in a CNN. Thus we propose a module Residual Appearance Renderer (RAR) that is trained to learn these complex appearance effects (inter-reflections, cast shadows, near-field illumination, and realistic shading).

1.5 Thesis Outline

The structure of the thesis is as follows: In Part I we present our approach of learning data driven priors for Inverse Rendering of a face (in Chapter 3) and a scene (in Chapter 4). In Chapter **??** we introduce learning data-driven priors for inverse rendering and discuss related works in Chapter 2. In Part II we discuss our approach of enforcing low-rank constraints for different sub-problems of inverse rendering, like Structure from Motion (in Chapter 7) and Uncalibrated Photometric Stereo (in Chapter 8). We introduce constrained low-rank optimization for Inverse Rendering in Chapter 5 and provide relevant background in Chapter 6

Part I

Learning Data-driven Priors

Chapter 2

Background

Classical approaches for inverse rendering: For inverse rendering from a few images, most traditional optimization-based approaches make strong assumptions about statistical priors on illumination and/or reflectance. A variety of sub-problems have been studied, such as intrinsic image decomposition [155], shape from shading [123, 119], and BRDF estimation [103]. Recently, SIRFS [6] showed it is possible to factorize an image of an object or a scene into surface normals, albedo, and spherical harmonics lighting. In [136] the authors use CNNs to predict the initial depth and then solve inverse rendering with an optimization. From an RGBD video, Zhang *et. al.* [178] proposed an optimization method to obtain reflectance and illumination of an indoor room. The problem of inverse rendering in the form of SfS gained particular attention in the domain of human facial modeling. This research was precipitated by the advent of the 3D Morphable Model (3DMM) [14] as a potential prior for shape and reflectance. Recent works used facial priors to reconstruct shape from a single image [79, 78, 27, 131] or from multiple images [130]. These optimization-based methods, although physically grounded, often do not generalize well to real images where those statistical priors are no longer valid.

Learning based approaches for inverse rendering of faces: In recent years, researchers have

focused on data driven approaches for learning priors rather than hand-designing them for the purpose of inverse rendering. Attempts at learning such priors were presented in [153] using Deep Belief Nets and in [88] using a convolutional encoder-decoder based network. However these early works were limited in their performance on real world unconstrained faces. Recent work from Shu *et. al.* [141] aims to find a meaningful latent space for normals, albedo and lighting to facilitate various editing of faces. Tewari *et. al.* [156] solves this facial disentanglement problem by fitting a 3DMM for shape and reflectance and regressing illumination coefficients. Both [141, 156] learn directly from real world faces by using convolutional encoder-decoder based architectures. Decompositions produced by [141] are often not realistic; and [156] only captures low frequency variations. In contrast, our method learns from a mixture of labeled synthetic and unlabeled real world faces using a novel decomposition architecture.

Another direction of research is to estimate shape or illumination of a face independently. Recently many research works aim to reconstruct the shape of real world faces by learning from synthetic data; by fitting a 3DMM [159, 91, 157], by predicting a depth map and subsequent non-rigid deformation to obtain a mesh [133] and by regressing a normal map [160]. Similarly [184] proposed a method to estimate lighting directly from a face. These learning based independent component estimation methods can not be trained with unlabeled real world data and thus suffer from the ability to handle unseen face modalities. In contrast our joint estimation approach performs the complete decomposition while allowing us to train on unlabeled real world images using self-supervised reconstruction loss.

Learning based approaches for inverse rendering of scenes: With recent advances in deep learning, researchers have proposed to learn data-driven priors to solve some of these inverse problems with CNNs, many of which have achieved promising results. For example, it is shown that depth and normals may be estimated from a single image [39, 44, 188] or multiple images [154].

Parametric BRDF may be estimated either from an RGBD sequence of an object [108, 83] or for planar surfaces [97]. Lighting may also be estimated from images, either as an environment map [45, 60], or spherical harmonics [184] or point lights [179]. Some recent works also jointly learn some of the intrinsic components of an object, like reflectance and illumination [48, 165], reflectance and shape [94], and normal, BRDF, and distant lighting [138, 99]. Nevertheless, these efforts are mainly limited to objects rather than scenes, and do not model the aforementioned residual appearance effects such as inter-reflection, near-field lighting, and cast shadows present in real images.

Architectures for learning based inverse rendering: In [141], a convolutional auto-encoder was used for disentanglement and generating normal and albedo images. However recent advances in skip-connection based convolutional encoder-decoder architectures for image to image translations [129, 65, 187] have also motivated their use in [139]. Even though skip connection based architectures are successful in transferring high frequency information from input to output, they fail to produce meaningful disentanglement of both low and high frequencies. Our proposed decomposition architecture uses residual block based connections that allow the flow of high frequency information from input to output while each layer learns both high and low frequency features. A residual block based architecture was used for image to image translation in [71] for style transfer and in a completely different domain to learn a latent subspace with Generative Adversarial Networks [101].

Differentiable Renderer. A few recent works from the graphics community proposed differentiable Monte Carlo renderers [93, 33] for optimizing rendering parameters (e.g., camera poses, scattering parameters) for synthetic 3D scenes. Neural mesh renderer [77] addressed the problem of differentiable visibility and rasterization. Our proposed RAR is in the same spirit, but its goal is to synthesize the complex appearance effects for inverse rendering on *real images*, which is signifi-

cantly more challenging.

Datasets for inverse rendering. High-quality synthetic data is essential for learning-based inverse rendering. SUNCG [144] created a large-scale 3D indoor scene dataset. The images of the SUNCG dataset are not photo-realistic as they are rendered with OpenGL using diffuse materials and point source lighting. An extension of this dataset, PBRS [180], uses physically based rendering to generate photo-realistic images. However, due to the computational bottleneck in ray-tracing, the rendered images are quite noisy and limited to one lighting condition. There also exist a few real-world datasets with partial labels on geometry, reflectance, or lighting. NYUv2 [111] provides surface normals from indoor scenes. OpenSurface [12] provides partial segmentation of objects with their glossiness properties labeled by humans. Relative reflectance judgments from humans are provided in the IIW dataset [11] which are used in many intrinsic image decomposition methods. In contrast to these works, we created a large-scale synthetic dataset with significant image quality improvement. For faces, we create synthetic data using 3DMM [14] in various viewpoints, reflectance and illumination. We render these models using 27 dimensional spherical harmonics coefficients (9 for each RGB channel), which comes from a distribution estimated by fitting 3DMM over real images from the CelebA dataset using classical methods. We use CelebA [102] as real data.

Intrinsic image decomposition. Intrinsic image decomposition is a sub-problem of inverse rendering, where a single image is decomposed into albedo and shading. Recent methods learn intrinsic image decomposition from labeled synthetic data [92, 110, 138] and from unlabeled [96] or partially labeled real data [186, 95, 113, 11]. Intrinsic image decomposition methods do not explicitly recover geometry, illumination or glossiness of the material, but rather combine them together as shading. In contrast, our goal is to perform a complete inverse rendering which has a wider range of applications in AR/VR.

Chapter 3

Inverse Rendering of a Face

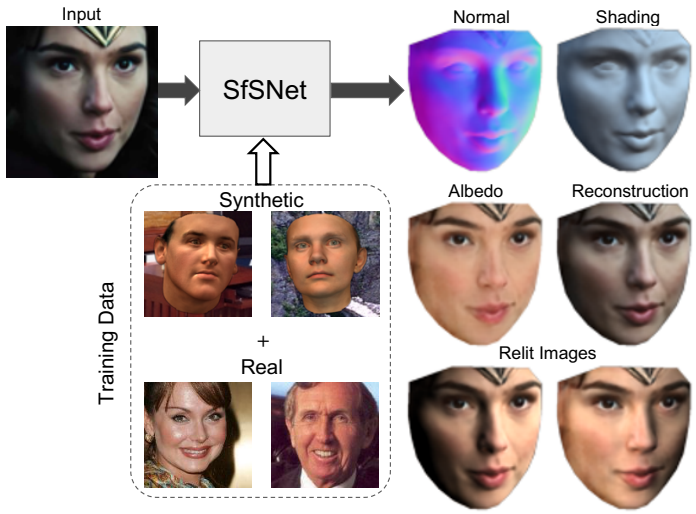


Figure 3-1: **Decomposing real world faces into shape, reflectance and illuminance.** We present SfSNet that learns from a combination of labeled synthetic and unlabeled real data to produce an accurate decomposition of an image into surface normals, albedo and lighting. Relit images are shown to highlight the accuracy of the decomposition. (Best viewed in color)

3.1 Introduction

In this work, we propose a method to decompose unconstrained real world faces into shape, reflectance and illuminance assuming lambertian reflectance. This decomposition or inverse rendering is a classical and fundamental problem in computer vision [155, 123, 119, 7]. It allows one to

edit an image, for example with re-lighting and light transfer [166]. Inverse rendering also has potential applications in Augmented Reality, where it is important to understand the illumination and reflectance of a human face. A major obstacle in solving this decomposition or any of its individual components for real images is the limited availability of ground-truth training data. Even though it is possible to collect real world facial shapes, it is extremely difficult to build a dataset of reflectance and illuminance of images in the wild at a large scale. Previous works have attempted to learn surface normal from synthetic data [159, 133], which often performs imperfectly in the presence of real world variations like illumination and expression. Supervised learning can generalize poorly if real test data comes from a different distribution than the synthetic training data.

We propose a solution to this challenge by jointly learning all intrinsic components of the decomposition from real data. In the absence of ground-truth supervision for real data, photometric reconstruction loss can be used to validate the decomposition. This photometric consistency between the original image and inferred normal, albedo and illuminance provide strong cues for inverse rendering. However it is not possible to learn from real images only with reconstruction loss, as this may cause the individual components to collapse on each other and produce trivial solutions. Thus, a natural step forward is to get the best of both worlds by simultaneously using supervised data when available and real world data with reconstruction loss in their absence. To this end we propose a training paradigm 'SfS-supervision'.

To achieve this goal we propose a novel deep architecture called SfSNet, which attempts to mimic the physical model of lambertian image generation while learning from a mixture of labeled synthetic and unlabeled real world images. Training from this mixed data allows the network to learn low frequency variations in facial geometry, reflectance and lighting from synthetic data while simultaneously understanding the high frequency details in real data using shading cues through reconstruction loss. This idea is motivated by the classical works in the Shape from Shading (SfS)

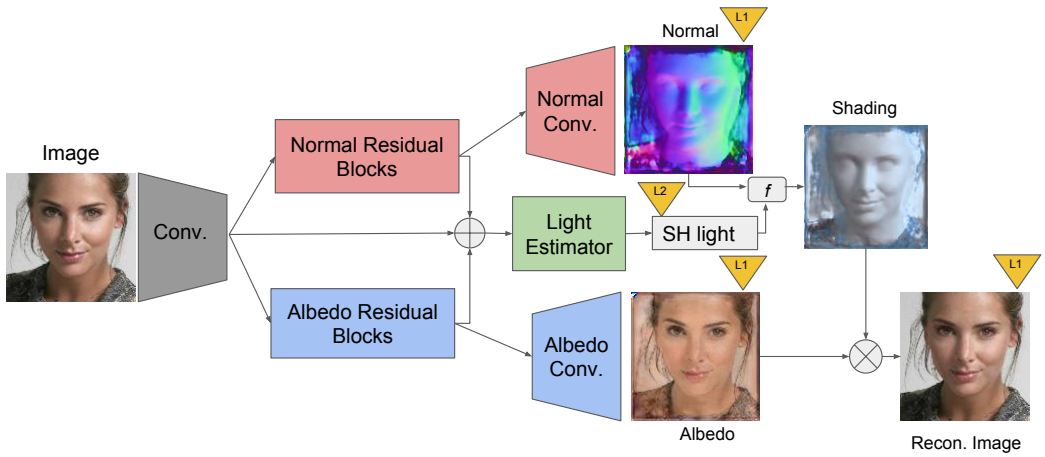


Figure 3-2: : **Network Architecture.** Our SfSNet consists of a novel decomposition architecture that uses residual blocks to produce normal and albedo features. They are further utilized along with image features to estimate lighting, inspired by a physical rendering model. f combines normal and lighting to produce shading. (Best viewed in color)

literature where often a reference model is used to compensate for the low frequency variations and then shading cues are utilized for obtaining high frequency details [78]. To meet this goal we develop a decomposition architecture with residual blocks that learns a complete separation of image features into normals and albedo. Then we use normal, albedo and image features to regress the illumination parameters. This is based on the observation that in classical illumination modeling, lighting is estimated from image, normal and albedo by solving an over-constrained system of equations. Our network architecture is illustrated in Figure 3-2. Our model and code is available for research purposes at https://senguptaumd.github.io/SfSNet/.

We evaluate our approach on the real world CelebA dataset [102] and present extensive comparison with recent state-of-the-art methods [141, 156], which also perform inverse rendering of faces. SfSNet produces significantly better reconstruction than [141, 156] on the same images that are showcased in their papers. We further compare SfSNet with state-of-the-art methods that aim to solve for only one component of the inverse rendering such as normals or lighting. SfSNet outperforms a recent approach that estimates normal independently [160], by improving normal estimation accuracy by 47% (37% to 84%) on the Photoface dataset [177], which contains faces captured under

harsh lighting. We also compare against 'Pix2Vertex' [133], which only estimate high resolution meshes. We demonstrate that SfSNet reconstructions are significantly more robust to expression and illumination variation compared to 'Pix2Vertex'. This results from the fact that we are jointly solving for all components, which allows us to train on real images through reconstruction loss. SfSNet outperforms 'Pix2Vertex' (before meshing) by 19% (25% to 44%) without training on the Photoface dataset. We also outperform a recent approach on lighting estimation 'LDAN' [184] by 12.5% (65.9% to 78.4%).

In summary our main contributions are as follows:

(1) We propose a network, SfSNet, inspired by a lambertian rendering model. This uses a decomposition architecture with residual blocks to separate image features into normal and albedo, further used to estimate lighting.

(2) We present a training paradigm 'SfS-supervision', which allows learning from a mixture of labeled synthetic and unlabeled real world images. This allows us to jointly learn normal, albedo and lighting from real images via reconstruction loss, outperforming approaches that only learn an individual component.

(3) SfSNet produces remarkably better visual results compared to state-of-the-art methods for inverse rendering [141, 156]. In comparison with methods that obtain one component of the inverse rendering [160, 133, 184], SfSNet is significantly better, especially for images with expression and non-ambient illumination.

3.2 Approach

Our goal is to use synthetic data with ground-truth supervision over normal, albedo and lighting along with real images with no ground-truth. We assume image formation under lambertian reflectance. Let $N(p)$, $A(p)$ and $I(p)$ denote the normal, albedo and image intensity at each pixel p.

We represent lighting L as nine dimensional second order spherical harmonics coefficients for each of the RGB channels. The image formation process under lambertian reflectance, following [9] is represented in equation (3.1), where $f_{render}(.)$ is a differentiable function.

$$I(p) = f_{render}(N(p), A(p), L) \tag{3.1}$$

3.2.1 'SfS-supervision' Training

Our 'SfS-supervision' consists of a multi-stage training as follows: (a) We train a simple skip-connection based encoder-decoder network on labeled synthetic data. (b) We apply this network on real data to obtain normal, albedo and lighting estimates. These elements will be used in the next stage as 'pseudo-supervision'. (c) We train our SfSNet with a mini-batch of synthetic data with ground-truth labels and real data with 'pseudo-supervision' labels. Along with supervision loss over normal, albedo and lighting we use a photometric reconstruction loss that aims to minimize the error between the original image and the reconstructed image following equation (3.1).

This reconstruction loss plays a key role in learning from real data using shading cues while 'pseudo-supervision' prevents the collapse of individual components of the decomposition that produce trivial solutions. In Section 3.5 we show that 'SfS-supervision' significantly improves inverse rendering over training on synthetic data only. Our idea of 'SfS-supervision' is motivated by the classical methods in SfS, where a 3DMM or a reference shape is first fitted and then used as a prior to recover the details [78, 79]. Similarly in 'SfS-supervision', low frequency variations are obtained by learning from synthetic data. Then they are used as priors or 'pseudo-supervision' along with photometric reconstruction loss to add high frequency details.

Our loss function is described in equation (3.2). For E_N, E_A and E_{recon} we use L_1 loss over all pixels of the face for normal, albedo and reconstruction respectively; E_L is defined as the L_2 loss

over 27 dimensional spherical harmonic coefficients. We train with a mixture of synthetic and real data in every mini-batch. We use λ_{recon}, λ_N and λ_A = 0.5 and λ_L = 0.1. Details of reconstruction loss under lambertian reflectance are presented in the Appendix 10.1.4.

$$E = \lambda_{recon}E_{recon} + \lambda_N E_N + \lambda_A E_A + \lambda_L E_L \tag{3.2}$$

3.2.2 Proposed Architecture

A common architecture in image to image translation is skip-connection based encoder-decoder networks [129, 65]. In the context of inverse rendering, [139] used a similar skip-connection based network to perform decomposition for synthetic images consisting of ShapeNet [31] objects. We observe that in these networks most of the high frequency variations are passed from encoder to decoders through the skip connections. Thus the networks do not have to necessarily reason about whether high frequency variations like wrinkles and beards come from normal or albedo. Also in these networks the illumination is estimated only from the image features directly and is connected to normal and albedo through reconstruction loss only. However since illumination can be estimated from image, normal and albedo by solving an over-constrained system of equations, it makes more sense to predict lighting from image, normal and albedo features.

The above observations motivate us to develop an architecture that learns to separate both low and high frequency variations into normal and albedo to obtain a meaningful subspace that can be further used along with image features to predict lighting. Thus we use a residual block based architecture as shown in Figure 3-2. The decomposition with 'Normal Residual Blocks' and 'Albedo Residual Blocks' allows complete separation of image features into albedo and normal features as shown in Figure 3-3b. The skip connections (shown in red) allow the high frequency information to flow directly from input feature to output feature while the individual layers can also learn from

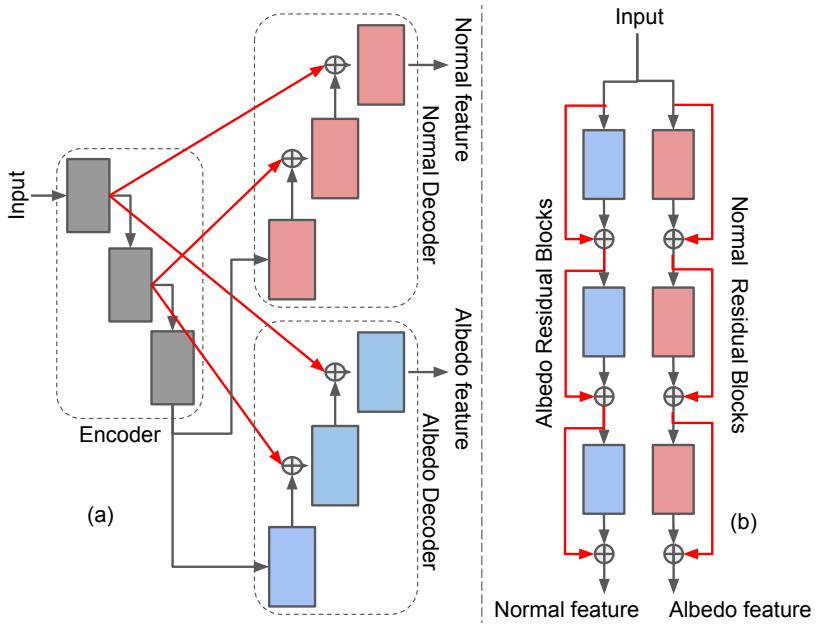


Figure 3-3: **Decomposition architectures.** We experiment with two architectures: (a) skip connection based encoder-decoder; (b) proposed residual block based network. Skip connections are shown in red.

the high frequency information present in the skip connections. This lets the network learn from both high and low frequency information and produce a meaningful separation of features at the output. In contrast a skip connection based convolutional encoder-decoder network as shown in Figure 3-3a consists of skip connections (shown in red) that bypass all the intermediate layers and flow directly to the output. This architecture allows us to estimate lighting from a combination of image, normal and albedo features. In Section 3.5 we show that using a residual block based decomposition improves lighting estimation by 11% (67.7% to 78.4%) compared to a skip connection based encoder-decoder.

The network uses few layers of convolution to obtain image features, denoted by I_f which is the output of the 'Conv' block in Figure 3-2. I_f is the input to two different residual blocks denoted as 'Normal Residual Blocks' and 'Albedo Residual Blocks', which take the image features and learns to separate them into normal and albedo features. Let the output of 'Normal Residual Blocks' and 'Albedo Residual Blocks' be N_f and A_f respectively. N_f and A_f are further processed

through 'Normal Conv' and 'Albedo Conv' respectively to obtain normal and albedo aligned with the original face. To estimate lighting we use image (I_f), normal (N_f) and albedo (A_f) features in the 'Light Estimator' block of Figure 3-2 to obtain 27 dimensional spherical harmonic coefficients of lighting. The 'Light Estimator' block simply concatenates image, normal and albedo features followed by 1x1 convolutions, average pooling and a fully connected layer to produce lighting coefficients. The details of the network are provided in the Appendix 10.1.1.

3.2.3 Implementation Details

To generate synthetic data we use 3DMM [14] in various viewpoints, reflectance and illumination. We render these models using 27 dimensional spherical harmonics coefficients (9 for each RGB channel), which comes from a distribution estimated by fitting 3DMM over real images from the CelebA dataset using classical methods. We use CelebA [102] as real data for both training, validation and testing, following the provided protocol. We detect keypoints using [124] and create a mask based on these keypoints. Each of the 'Residual Blocks' consists of 5 residual blocks based on the structure proposed by [57]. Our network is trained with input images of size 128×128 and the residual blocks all operate at 64×64 resolution. The 'pseudo-supervision' for real world images are generated by training a simple skip-connection based encoder-decoder network, similar to [141], on synthetic data. The details of this network, referred to as 'SkipNet' in Section 3.5, are provided in the Appendix 10.1.2.

3.3 Comparison with State-of-the-art Methods

We compare our SfSNet with [141, 156] qualitatively on unconstrained real world faces. As an application of inverse rendering we perform light transfer between a pair of images, which also illustrates the correctness of the decomposition. We quantitatively evaluate the estimated normals

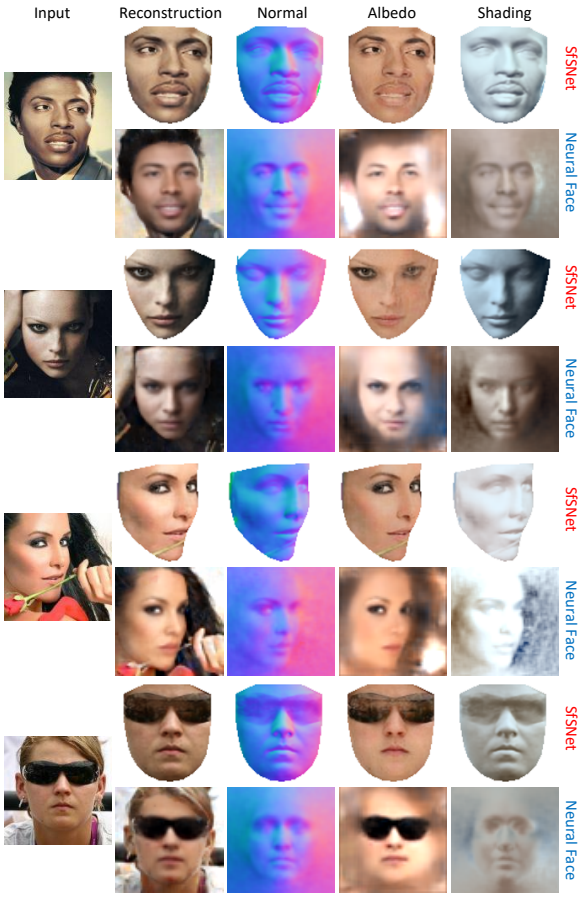


Figure 3-4: **Inverse Rendering. SfSNet vs 'Neural Face'** [141] on the data showcased by the authors.

on the Photoface dataset [177] and compare with the state-of-the-art [160, 133]. Similarly we also evaluate the accuracy of estimated lighting on the MultiPIE dataset [51] and compare with [184]. We outperform state-of-the-art methods by a large margin both qualitatively and quantitatively.

3.3.1 Evaluation of Inverse Rendering

In Figures 3-4 and 3-5 we compare performance of our SfSNet with 'Neural Face' [141] on inverse rendering and light transfer respectively. The results are shown on the same images used in their paper. The results clearly show that SfSNet performs more realistic decomposition than 'Neural Face'. Note that in light transfer 'Neural Face' does not use their decomposition, but rather recomputes the albedo of the target image numerically. Light transfer results in Figure 3-5, show that SfSNet recovers and transfers the correct ambient light compared to 'Neural Face', which fails to

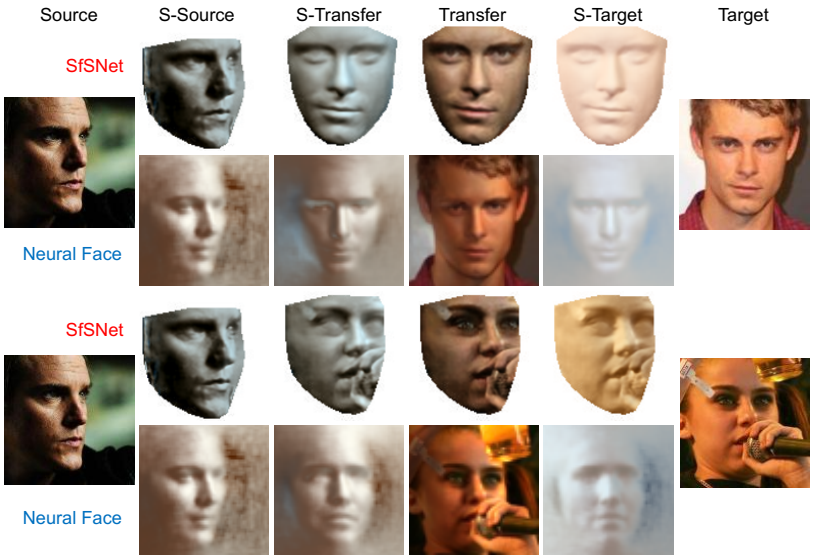


Figure 3-5: **Light Transfer. SfSNet vs 'Neural Face'** [141] on the image showcased by the authors. We transfer the lighting of the 'Source' image to the 'Target' image to produce 'Transfer' image. *S* denotes shading. Both 'Target' images contain an orangey glow, which is not present in the 'Source' image. Ideally in the 'Transfer' image, the orangey glow should be removed. 'Neural Face' fails to get rid of the orangey lighting effect of the 'Target' image in the 'Transfer' image.

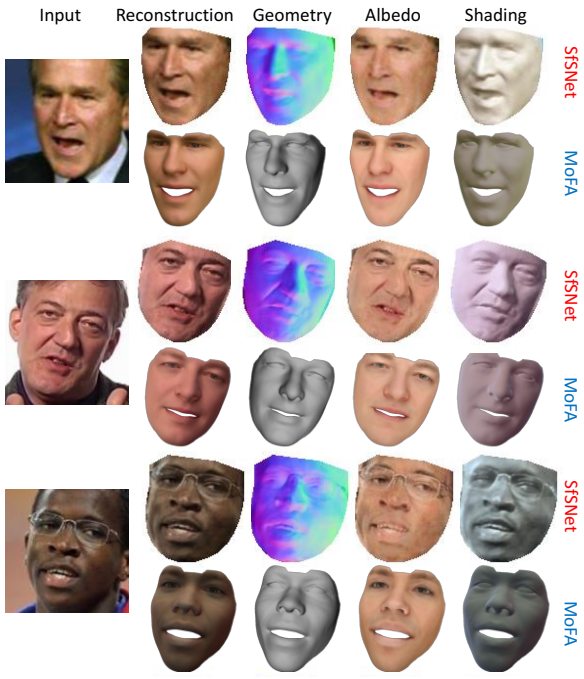


Figure 3-6: **Inverse Rendering. SfSNet vs 'MoFA'** [156] on the data provided by the authors of the paper.

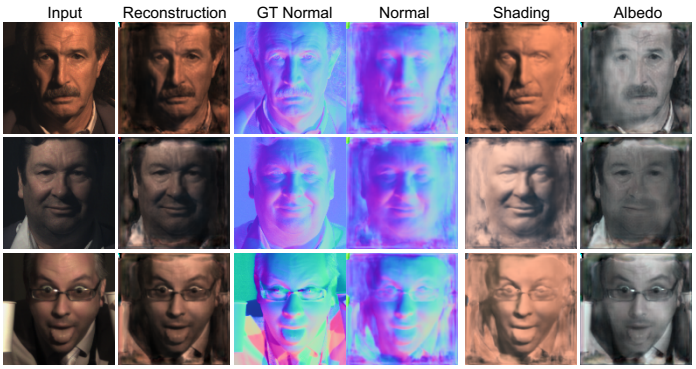


Figure 3-7: **Inverse Rendering on the Photoface dataset [177] with 'SfSNet-finetuned'.** The ground-truth albedo is in gray-scale and it encourages our network to also output gray-scale albedo.

get rid of the orangey lighting from the target images. We also compare inverse rendering results of SfSNet on the images provided to us by the authors of [156] in Figure 3-6. Since [156] aims to fit a 3DMM that can only capture low frequency variations, we obtain more realistic normals, albedo and lighting than them.

3.3.2 Evaluation of Facial Shape Recovery

In this section we compare the quality of our reconstructed normals with that of current state-of-the-art methods that only recover shape from a single image. We use the Photoface dataset [177], which provides ground-truth normals for images taken under harsh lighting. First we compare with algorithms that also train on the Photoface dataset. We finetune our SfSNet on this dataset using ground truth normals and albedo as supervision since they are available. We compare our 'SfSNet-ft' with 'NiW' [160] and other baseline algorithms, 'Marr Rev.' [5] and 'UberNet' [84], reported in [160] in Table 3.1. The metric used for this task is mean angular error of the normals and the percentage of pixels at various angular error thresholds as in [160]. Since the exact training split of the dataset is not provided by the authors, we create a random split based on identity with 100 individuals in test data as mentioned in their paper. Our 'SfSNet-ft' improves normal estimation accuracy by more than a factor of two for the most challenging threshold of 20 degrees accuracy. In Figure 3-7 we show visual results of decomposition on test data of the Photoface dataset.

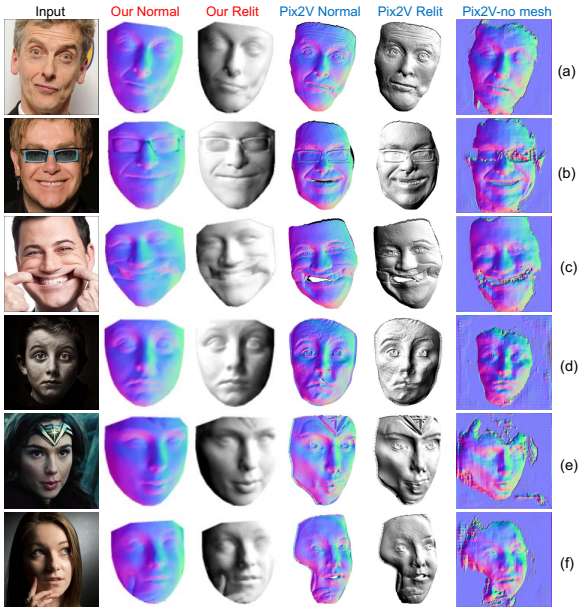


Figure 3-8: **SfSNet vs Pix2Vertex** [133]. Normals produced by SfSNet are significantly better than Pix2Vertex, especially for non-ambient illumination and expression. 'Relit' images are generated by directional lighting and uniform albedo to highlight the quality of the reconstructed normals. Note that (a), (b) and (c) are the images showcased by the authors. (Best viewed in color)

Algorithm	Mean ±std	< 20°	< 25°	< 30°
3DMM	26.3 ±10.2	4.3%	56.1%	**89.4%**
Pix2Vertex[133]	33.9 ±5.6	24.8%	36.1%	47.6%
SfSNet	**25.5 ±9.3**	**43.6%**	**57.5%**	68.7%
Marr Rev.[5]	28.3 ±10.1	31.8%	36.5%	44.4%
UberNet[84]	29.1 ±11.5	30.8%	36.5%	55.2%
NiW[160]	22.0 ±6.3	36.6%	59.8%	79.6%
SfSNet-ft	**12.8 ±5.4**	**83.7%**	**90.8%**	**94.5%**

Table 3.1: **Normal reconstruction error on the Photoface dataset**. 3DMM, Pix2Vertex and SfSNet are not trained on this dataset. Marr Rev., UberNet, NiW and SfSNet-finetuned (SfSNet-ft) are trained on the training split of this dataset. Lower is better for mean error (column 1), and higher is better for the percentage of correct pixels at various thresholds (columns 3-5).

Next we compare our algorithm with 'Pix2Vertex' [133], which is trained on higher resolution 512×512 images. 'Pix2Vertex' learns to produce a depth map and a deformation map that are post-processed to produce a mesh. In contrast our goal is to perform inverse rendering. Since we are able to train on real data, unlike 'Pix2Vertex', which is trained on synthetic data, we can better capture real world variations. Figure 3-8 compares normals produced by SfSNet with that of 'Pix2Vertex' both before and after meshing on the images showcased by the authors. Since 'Pix2vertex' handles larger resolution and produces meshes, their normals can capture more details than ours. But

Figure 3-9: Selected results from **top 5% (a,b,c,d)** and **worst 5% (e,f,g,h)** reconstructed images. (Best viewed in color)

with more expression and non-ambient illumination like (c), (d), (e) and (f) in Figure 3-8, we produce fewer artifacts and more realistic normals and shading. SfSNet is around 2000× faster than 'Pix2Vertex' due to the expensive mesh generation post-processing. These results show that learning all components of inverse rendering jointly allows us to train on real images to capture better variations than 'Pix2Vertex'. We further compare SfSNet with the normals produced by 'Pix2Vertex' quantitatively before meshing on the Photoface dataset. SfSNet, 'Pix2Vertex' and 3DMM are not trained on this dataset. The results shown in Table 3.1 shows that SfSNet outperforms 'Pix2Vertex' and 3DMM by a significant margin.

Figure 3-10: **Light transfer.** Our SfSNet allows us to transfer lighting of the 'Source' image to the 'Target' image to produce 'Transfer' image. 'S' refers to shading. (Best viewed in color)

Algorithm	top-1%	top-2%	top-3%
SIRFS log [7]	60.72	79.65	87.27
LDAN [184]	65.87	85.17	92.46
SfSNet	**78.44**	**89.44**	**92.64**

Table 3.2: **Light Classification Accuracy on MultiPIE dataset.** SfSNet significantly outperforms 'LDAN'.

3.3.3 Evaluation of Light Estimation

We evaluate the quality of the estimated lighting using MultiPIE dataset [51] where each of the 250 individuals is photographed under 19 different lighting conditions. We perform 19-way classification, to check the consistency of the estimated lighting as described in [184] and compare with their proposed algorithm 'LDAN'. 'LDAN' estimates lighting independently from a single face image using adversarial learning. Results in Table 3.2 shows that we improve top-1% classification accuracy by 12.6% over 'LDAN'.

3.4 Results on CelebA

In Figure 3-9 we provide sample results on CelebA test data from the best 5% and worst 5% reconstructed images respectively. For every test face, we also relight the face using a directional light source that highlights the flaws in the decomposition. As expected the best results are for frontal faces with little or no expression and easy ambient lighting as shown in Figure 3-9 (a-d). The worst reconstructed images have large amounts of cast shadows, specularity and occlusions as shown in Figure 3-9 (e-h). However, the recovered normal and lighting are still reasonable. We also show interesting results on light transfer in Figure 3-10, which also highlights the quality of the decomposition. Note that the examples shown in (c) and (d) are particularly hard as source and target images have opposite lighting directions. More qualitative results on CelebA and comparison with [141, 156, 133] are provided in the Appendix 10.1.5.

3.5 Ablation Studies

We analyze the relative importance of mixed data training with 'SfS-supervision' compared to learning from synthetic data alone. We also contrast the SfSNet architecture with skip-connection based networks. For ablation studies, we consider photometric reconstruction loss (Recon. Error) and lighting classification accuracy (Lighting Acc.) as performance measures.

Role of 'SfS-supervision' training: To analyze the importance of our mixed data training we consider the SfSNet architecture and compare its performance using different training paradigms. We consider the following:

SfSNet-syn: We train SfSNet on synthetic data only.

SkipNet-syn: We observe that our residual block based network can not generalize well on unseen

real world data when trained on synthetic data, as there is no direct skip connections that can transfer high frequencies from input to output. However a skip connection based encoder-decoder network can generalize on unseen real world data. Thus we consider a skip connection based network, 'SkipNet', which is similar in structure with the network presented in [141], but with increased capacity and skip connections. We train 'SkipNet' on synthetic data only and this training paradigm is similar to [139], which also uses a skip-connection based network for decomposition in ShapeNet objects.

SfSNet: We use our 'SfS-supervision' to train our SfSNet, where 'pseudo-supervision' is generated by 'SkipNet'.

Training Paradigm	Recon. Error		Lighting Acc.		
	MAE	RMSE	Rank 1	Rank 2	Rank 3
SkipNet-syn	42.83	48.22	54.86%	76.78%	85.76%
SfSNet-syn	48.54	58.13	63.88%	80.52%	87.24%
SfSNet	**10.99**	**13.55**	**78.44%**	**89.52%**	**92.64%**

Table 3.3: **Role of 'SfS-supervision' training.** 'SfS-supervision' outperforms training on synthetic data only.

Note that another alternative is training on synthetic data and fine-tuning on real data. It has been shown in [141] that it is not possible to train the network on real data alone by using only reconstruction loss, as the ambiguities in the decomposition can not be constrained, leading to a trivial solution. We also find that the same argument is true in our experiments. Thus we compare our 'SfS-supervision' training paradigm with only synthetic data training in Table 3.3. The results show that our 'SfS-supervision' improves significantly over the 'pseudo-supervision' used from SkipNet, indicating that we are successfully using shading information to add details in the reconstruction.

Role of SfSNet architecture: We evaluate the effectiveness of our proposed architecture against a skip connection based architecture. Our proposed architecture estimates lighting from image, nor-

mal and albedo, as opposed to a skip connection based network which estimates lighting directly from the image only. SkipNet described in the Appendix based on [141] does not produce a good decomposition because of the fully connected bottleneck. Thus we compare with a fully convolutional architecture with skip connection, similar to Pix2Pix [65], which we refer to as Skipnet+. This network has one encoder, two decoders for normal and albedo and a fully connected layer from the output of the encoder to predict light (see Appendix 10.1.3 for details).

In Table 3.4 we show that our SfSNet outperforms SkipNet+, also trained using the 'SfS-supervision' paradigm. Although reconstruction error is similar for both networks, SfSNet predicts better lighting than 'SkipNet+'. This improved performance can be attributed to the fact that SfSNet learns an informative latent subspace for albedo and normal, which is further utilized along with image features to estimate lighting. Whereas in the case of the skip connection based network, the latent space is not informative as high frequency information is directly propagated from input to output bypassing the latent space. Thus lighting parameters estimated only from the latent space of the image encoder fail to capture the illumination variations.

Training Paradigm	Recon. Error		Lighting Acc.		
	MAE	RMSE	top-1%	top-2%	top-3%
SkipNet+	11.33	14.42	67.70%	85.08%	90.34%
SfSNet	**10.99**	**13.55**	**78.44%**	**89.52%**	**92.64%**

Table 3.4: **SfSNet vs SkipNet+**. Proposed SfSNet outperforms a skip connection based SkipNet+ which estimates lighting directly from the image.

Chapter 4

Inverse Rendering of an Indoor Scene

Figure 4-1: We propose a holistic data-driven approach for inverse rendering, *i.e.* decompose an indoor scene image into surface normals, albedo, glossiness segmentation (matte–blue, glossy–red and semi-glossy–green) and environment map.

4.1 Introduction

Inverse Rendering of a scene from a single image is more challenging than a face. In case of faces, researchers have traditionally modeled reflectance as lambertian. In contrast, a scene consists of multiple different objects with varying reflectance properties. Similarly, illumination in a face can be assumed to be distant and direct, thus can be modeled using spherical harmonics. Whereas, in a scene we observe complex appearance effects like inter-reflections, cast-shadows, near-field lighting and highlights. In this work we aim to learn from unlabeled real world images using self-supervised reconstruction loss while tackling these challenges related to representing scene reflectance and illumination.

In this chapter, we take the first step and propose a holistic data-driven approach for inverse

rendering of an indoor scene from a single image with CNNs. Unlike recent works which either estimate only one of the scene attributes [45, 39, 97] or are limited to single objects [83, 99, 108], our approach jointly learns and estimates reflectance (albedo and gloss), surface normals, and illumination of an indoor scene from a single image, as shown in Fig. 4-1. Inverse rendering of a *scene* is particularly challenging due to complex appearance effects (e.g., inter-reflection, cast shadows, near-field illumination, and realistic shading).

Two key contributions make our approach possible. 1) We introduce a new synthetic dataset using physically based rendering. 2) Our aim is to learn from unlabeled real data using self-supervised reconstruction loss. To this end, we propose the Residual Appearance Renderer (RAR) which learns to predict complex appearance effects on real images.

Rendering dataset. It is especially challenging for inverse rendering tasks to obtain accurate ground truth labels for real images. Hence, we create a large-scale synthetic dataset with physically-based rendering, named *SUNCG-PBR*, of all the 3D indoor scenes from SUNCG [144]. Compared to prior work PBRS [180], SUNCG-PBR significantly improves data quality in the following ways: (1) The rendering of a scene is performed under multiple natural illuminations. (2) We render the same scene twice. Once with all materials set to Lambertian and once with the default material settings to produce image pairs (diffuse, specular). (3) We utilize deep denoising [28], which allows us to render high-quality images from limited samples per pixel. Our dataset consists of 235,893 images with labels for normal, depth, albedo, Phong [90] model parameters and semantic segmentation. Examples are shown in Fig. 4-4. We plan to release the SUNCG-PBR dataset upon acceptance.

Residual Appearance Renderer. Our key idea is to learn from unlabeled real data using self-supervised reconstruction loss, which is enabled by our proposed Residual Appearance Renderer (RAR) module. While using a reconstruction loss for domain transfer from synthetic to real has been explored previously [134, 142, 99], their renderer is limited to direct illumination under distant

lighting with a single material. For real images of a scene, however, this simple direct illumination renderer cannot synthesize important, complex appearance effects, such as inter-reflections, cast shadows, near-field lighting, and realistic shading. These effects termed *residual appearance* in this paper, can only be simulated with the rendering equation via physically-based ray-tracing, which is non-differentiable and cannot be employed in a learning-based framework. To this end, we propose the Residual Appearance Renderer (RAR) module, which along with a direct illumination renderer, can reconstruct the original image from the estimated scene attributes, for self-supervised learning on real images.

Moreover, similar to prior works, we also incorporate sparse labels on real data (i.e., pair-wise reflectance comparison [11, 95, 186], sparse material segmentation [12]) as a form of weakly supervised learning, to further improve the performance on real images.

To our knowledge, our approach is the first data-driven solution to single-image based inverse rendering of a scene. SIRFS [6], which is a classical optimization based method, seems to be the only prior work with similar goals. Compared with SIRFS, as shown in Sec. 4.4, our method is more robust and accurate. In addition, we also compare with recent DL-based methods that estimate only one of the scene attributes, such as intrinsic images [95, 96, 186, 113], lighting [45], and normals [180]. Experimental results show that our approach outperforms most of these single-attribute methods (especially on real images), which seems to indicate that the joint learning of all these scene attributes is helpful for the ability to generalize.

4.2 Our Approach

We present a deep learning based approach for inverse rendering from a single 8-bit image, which is shown in Fig. 4-2. Specifically, given an input image I, we estimate all its intrinsic components, i.e., reflectance, geometry, and lighting. The reflectance is represented as a diffuse albedo map A

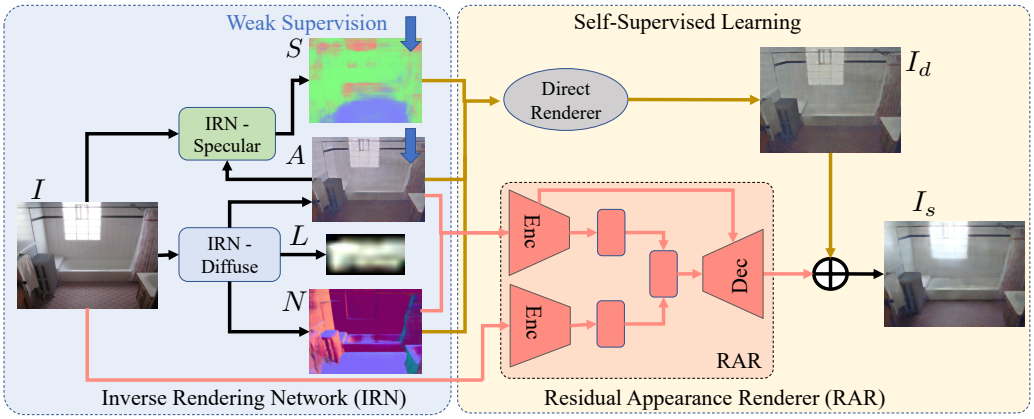


Figure 4-2: **Overview of our approach.** Our Inverse Rendering Network (IRN) consists of two modules IRN-Diffuse and IRN-Specular to predict albedo, normals, illumination map and glossiness segmentation respectively. We train on unlabeled real images using self-supervised reconstruction loss. Reconstruction loss consists of a closed-form Direct Renderer with no learnable parameters and proposed Residual Appearance Renderer (RAR), which learns to predict complex appearance effects.

plus Phong [90] model parameters for the specularity (i.e., K_s and N_s). We represent the geometry as a normal map N and the lighting as an environment map L. To make per-pixel Phong parameter estimation feasible, we simplify the task as a glossiness segmentation problem and define three glossiness categories (*et. al.*, matte, semi-glossy and glossy) based on the statistical distributions of BRDFs in our dataset. The output probability maps from the glossiness-based segmentation network are used as weights to compute the per-pixel Phong parameters. Thus, as shown in Fig. 4-2, our proposed neural Inverse Rendering Network (IRN) consists of two sub-networks: IRN-Diffuse, denoted as $h_d(\cdot;\Theta_d)$ which estimates N, A, and L, and IRN-Specular, denoted as $h_s(\cdot;\Theta_s)$, which estimates the glossiness segmentation S:

$$\text{IRN-Diffuse}: \quad h_d(I;\Theta_d) \rightarrow \left\{\hat{A},\hat{N},\hat{L}\right\} \tag{4.1}$$

$$\text{IRN-Specular}: \quad h_s(I,\hat{A};\Theta_s) \rightarrow \hat{S}, \tag{4.2}$$

where Θ_d and Θ_s are the network parameters. The estimated $\hat{K}_s$ and $\hat{N}_s$ can then be computed from the soft segmentation mask $\hat{S}$.

Using our synthetic data SUNCG-PBR, we can simply train these two networks ($h_d(\cdot;\Theta_d)$

and $h_s(\cdot;\Theta_s)$) with supervised learning – with only one caveat, i.e. we need to approximate the "ground truth" environment maps (using a separate network $h_e(\cdot;\Theta_e) \rightarrow \hat{L}^*$. See Sec. 4.2.1 for details). To generalize on real images, we use a self-supervised reconstruction loss. Specifically, as shown in Fig. 4-2, we use two renderers to re-synthesize the input image from the estimations. The direct renderer $f_d(\cdot)$ is a simple closed-form shading function with no learnable parameters, which synthesizes the direct illumination part $\hat{I}_d$ of the the raytraced image. The Residual Appearance Renderer (RAR), denoted by $f_r(\cdot;\Theta_r)$, is a trainable network module, which learns to synthesize the complex appearance effects $\hat{I}_r$

$$\text{Direct Renderer}: \quad f_d(\hat{A},\hat{N},\hat{L}) \rightarrow \hat{I}_d \tag{4.3}$$

$$\text{RAR}: \quad f_r(I,\hat{A},\hat{N};\Theta_r) \rightarrow \hat{I}_r. \tag{4.4}$$

The self-supervised reconstruction loss is thus defined as $||I - (\hat{I}_d + \hat{I}_r)||_1$. We explain the details of the direct renderer and the RAR in Sec. 4.2.2.

In summary, four sets of weights, Θ_d, Θ_s, Θ_e, and Θ_r, need to be learned from training. At inference time, given an input image I, we run IRN-Diffuse and IRN-Specular to estimate all the intrinsic components for the scene.

4.2.1 Training on Synthetic Data

We first train IRN-Diffuse and IRN-Specular on our synthetic dataset SUNCG-PBR with supervised learning. As shown in Fig. 4-2, IRN-Diffuse has a structure similar to [134], which consists of a convolutional encoder, followed by nine residual blocks and a convolutional decoder for estimating albedo and normals. We condition the lighting estimation block on the image, normals and albedo features. IRN-Specular is trained on the image and the albedo predicted by IRN-Diffuse to produce

a three-class glossiness segmentation map for matte, glossy and semi-glossy. IRN-Specular is based on U-Net [129], which has been shown to be effective for simple segmentation problems. Details of the IRN architecture are provided in the Appendix.

We use ground truth albedos A^* and normals N^* from SUNCG-PBR for supervised learning. The ground truth glossiness segmentation mask $S^*(x)$ is obtained by performing per-pixel classification based on the ground truth $K_s^*(x)$, $N_s^*(x)$, and $A^*(x)$ as follows:

$$\begin{aligned} \text{matte:} \quad & \frac{K_s^*(x)}{K_s^*(x) + A^*(x)} \leq T_0, \quad N_s^*(x) \leq N_0 \\ \text{glossy:} \quad & \frac{K_s^*(x)}{K_s^*(x) + A^*(x)} \geq T_1, \quad N_s^*(x) > N_1 \\ \text{semi-glossy:} \quad & \text{otherwise,} \end{aligned} \tag{4.5}$$

where $T_0 = 0.1$, $T_1 = 0.2$, $N_0 = 30$, $N_1 = 50$ are based on statistical distributions of the Phong parameters K_s and N_s in the SUNCG-PBR dataset.

The ground truth environmental lighting L^* is challenging to obtain, as it is the first-order approximation of the actual surface light field. We use environment maps as the exterior lighting for rendering SUNCG-PBR, but these environment maps cannot be directly set as L^*, because the virtual cameras are placed *inside* each of the indoor scenes. Due to occlusions, only a small fraction of the exterior lighting (e.g., through windows and open doors) is directly visible. The surface light field of each scene is mainly attributed to global illumination (i.e., inter-reflection) and some interior lighting. One could approximate L^* by minimizing the difference between the raytraced image I and the output I_d of the direct renderer $f_d(\cdot)$ with ground truth albedo A^* and normal N^*. However, we found this approximation to be inaccurate, since $f_d(\cdot)$ cannot model the residual appearance present in the raytraced image I.

We thus resort to a learning-based method to approximate the ground truth lighting L^*. Specif-

ically, we train a residual block based network, $h_e(\cdot;\Theta_e)$, to predict $\hat{L}^*$ from the input image I, normalw N^* and albedo A^*. We first train $h_e(\cdot;\Theta'_e)$ with the images synthesized by the direct renderer $f_d(\cdot)$ with ground truth normals, albedo and indoor lighting, $I_d = f_d(A^*, N^*, L)$, where L is randomly sampled from a set of real *indoor* environment maps.

Here the network learns a prior over the distribution of indoor lighting, i.e., $h(I_d;\Theta'_e) \rightarrow L$. Next, we fine-tune this network $h_e(\cdot;\Theta'_e)$ on the raytraced images I, by minimizing the reconstruction loss: $\|I - f_d(A^*, N^*, \hat{L}^*)\|$. Thus we obtain the approximated ground truth of the environmental lighting $\hat{L}^* = h_e(I;\Theta_e)$ which can best reconstruct the raytraced image I modelled by the direct render.

Finally, with all the ground truth components ready, the supervised loss for training IRN-Diffuse is

$$\begin{aligned} L_s = \lambda_1 ||\hat{N} - N^*||_1 + \lambda_2 ||\hat{A} - A^*||_1 \\ + \lambda_3 ||f_d(A^*, N^*, \hat{L}) - f_d(A^*, N^*, \hat{L}^*)||_1. \end{aligned} \tag{4.6}$$

where $\lambda_1 = 1$, $\lambda_2 = 1$, and $\lambda_3 = 0.5$. We use cross-entropy loss over $\hat{S}$ and S^* for training IRN-Specular.

4.2.2 Training on Real Images with Self-supervision

Learning from synthetic data alone is not sufficient to perform well on real images. Although SUNCG-PBR was created with physically-based rendering, the variation of objects, materials, and illumination is still limited compared to those in real images. Since obtaining ground truth labels for inverse rendering is almost impossible for real images (especially for reflectance and illumination), we use two key ideas for domain transfer from synthetic to real: (1) self-supervised reconstruction loss and (2) weak supervision from sparse labels.

Previous works on faces [134, 142] and objects [99] have shown success in using a self-supervised reconstruction loss for learning from unlabeled real images. As mentioned earlier, the reconstruction in these prior works is limited to the direct renderer $f_d(\cdot)$, which is a simple closed-form shading function (under distant lighting) with no learnable parameters. In this paper, we implement $f_d(\cdot)$ simply as

$$\hat{I}_d = f_d(\hat{A}, \hat{N}, \hat{L}) = \hat{A} \sum_i \max(0, \hat{N} \cdot \hat{L}_i), \tag{4.7}$$

where $\hat{L}_i$ corresponds to the pixels on the environment map $\hat{L}$. While using $f_d(\cdot)$ to compute the reconstruction loss may work well for faces [134] or small objects with homogeneous material [99], we found that it fails for inverse rendering of a scene. In order to synthesize the aforementioned residual appearances (e.g., inter-reflection, cast shadows, near-field lighting), we propose to use the differentiable Residual Appearance Renderer (RAR), $f_r(\cdot; \Theta_r)$, which learns to predict a residual image $\hat{I}_r$. The self-supervised reconstruction loss is thus defined as $L_u = ||I - (\hat{I}_d + \hat{I}_r)||_1$.

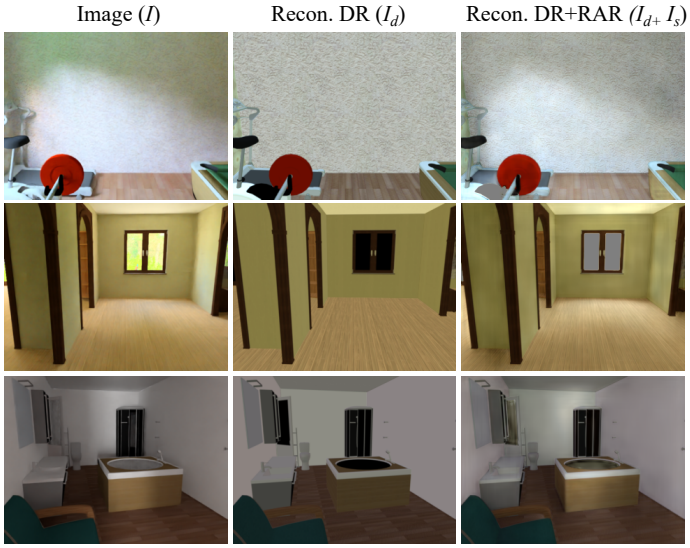


Figure 4-3: RAR $f_r(\cdot)$ learns to predict complex appearance effects (e.g. near-field lighting, cast shadows, inter-reflections) which cannot be modeled by a direct renderer (DR) $f_d(\cdot)$.

Our goal is to train RAR to capture only the residual appearances but *not* to correct the artifacts of the direct rendered image due to faulty normals, albedo, and lighting estimation of the IRN. To achieve this goal, we train RAR *only* on synthetic data with ground-truth normals and albedo, and

fix it for training on real data, so that it only learns to correctly predict the residual appearances when the direct renderer reconstruction is accurate.

As shown in Fig. 4-2, RAR consists of a U-Net [129], with normals and albedo as its input, and latent image features ($D = 300$ dimension) learned by a convolutional encoder ('Enc'). We combine the image features at the end of the U-Net encoder and process them with the U-Net decoder to produce the residual image. As shown in Fig. 4-3, RAR indeed learns to synthesize complex residual appearance effects present in the original input image.

Similar to prior work [186, 95], we use sparse labels over reflectance as weak supervision during training on real images. Specifically, we use pair-wise relative reflectance judgments from the Intrinsic Image in the Wild (IIW) dataset [11] as a form of supervision over albedo. For glossiness segmentation we use sparse human annotations from the OpenSurfaces dataset [12] as weak labels. More details are provided in the supplementary material. As shown later in Sec. 4.5, using such weak supervision can substantially improve performance on real images.

4.2.3 Training Procedure

We summarize the different stages of training from synthetic to real data. More details are in the supplementary.

Estimate GT indoor lighting: (a) First train $h_e(\cdot; \Theta'_e)$ on images rendered by the direct renderer $f_d(\cdot)$. (b) Fine-tune $h_e(\cdot; \Theta_e)$ on raytraced synthetic images to estimate GT indoor environment map $\hat{L}^*$.

Train on synthetic images: (a) Train IRN-Diffuse with supervised L1 loss on albedo, normal and lighting. (b) Train RAR. (c) Train IRN-Specular with the input image and the albedo predicted by IRN-Diffuse.

Train on real images: Fine-tune IRN-Diffuse and IRN-Specular on real data with (1) the

pseudo-supervision over albedo, normal and lighting (to handle ambiguity of decomposition as proposed in [134]), (2) the self-supervised reconstruction loss L_u with RAR, and (3) the weak supervision over the albedo (i.e., pair-wise relative reflectance judgment) and the sparse glossiness segmentation.

4.3 The SUNCG-PBR Dataset

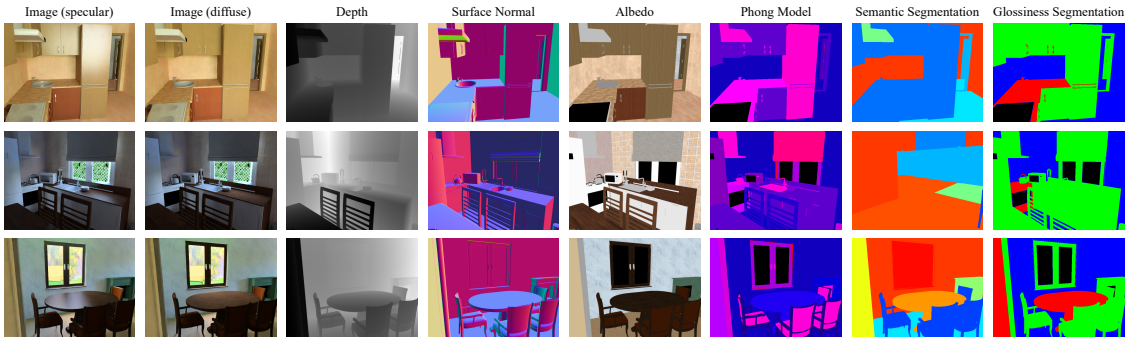


Figure 4-4: **Our SUNCG-PBR Dataset.** We provide 235,893 images of a scene assuming specular and diffuse reflectance along with ground truth depth, surface normals, albedo, Phong model parameters, semantic segmentation and glossiness segmentation.

High-quality synthetic datasets are essential for learning-based inverse rendering. The SUNCG dataset [144] contains 45,622 indoor scenes with 2644 unique objects, but their images are rendered with OpenGL under fixed point light sources. The PBRS dataset [180] extends the SUNCG dataset by using physically based rendering with Mitsuba [68]. Yet, due to a limited computational budget, many rendered images in PBRS are quite noisy. Moreover, the images in PBRS are rendered with only diffuse materials and a single outdoor environment map, which also significantly limits the photo-realism of the rendered images. High-quality photo-realistic images are necessary for training RAR to capture residual appearances.

In this paper, we introduce a new dataset named SUNCG-PBR, which improves data quality in the following ways: (1) The rendering is performed under multiple outdoor environment maps. (2) We render the same scene twice, once with all materials set to lambertian and once with the

default material settings. This offers (diffuse, specular) image pairs which can be useful to the community for learning to remove highlights and many other potential applications. (3) We utilize deep denoising [28], which allows us to raytrace high-quality images from limited samples per pixel. Our dataset consists of 235,893 images with labels related to normal, depth, albedo, Phong [90] model parameters, semantic and glossiness segmentation. Examples are shown in Fig. 4-4. A comparison with the SUNCG and PBRS datasets is shown in Fig. 4-5.

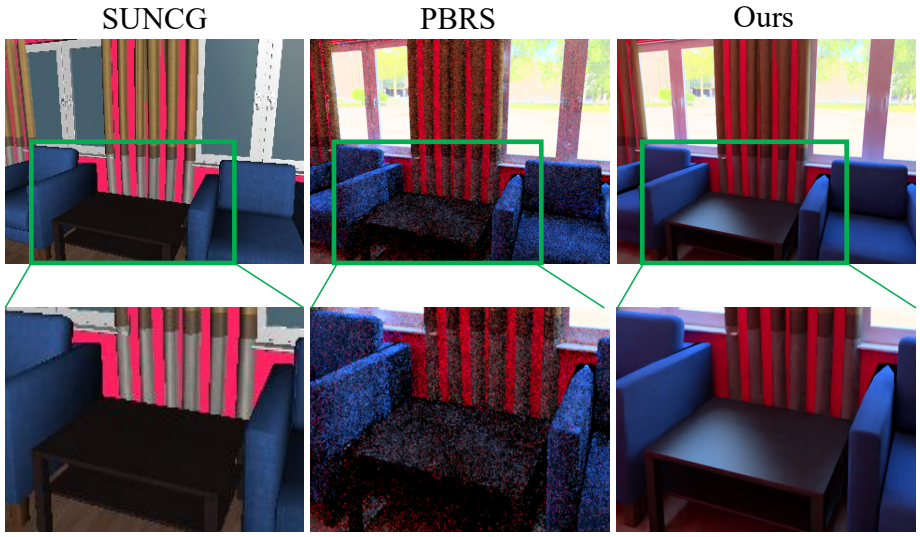


Figure 4-5: **Comparison with PBRS [180] and SUNCG [144].** Our dataset provides more photo-realistic and less noisy images with specular highlights under multiple lighting conditions.

4.4 Experimental Results

Comparison with SIRFS.

SIRFS [6] is an optimization-based method for inverse rendering, which estimates surface normals, albedo and spherical harmonics lighting from a single image. It is an inspiring work, as it shows the power of using statistical priors (over lighting, reflectance, and geometry) for inverse rendering from a single image. We compare with SIRFS on the test data from the IIW dataset [11]. As shown in Fig. 4-6, our method produces more accurate normals and better disambiguation of reflectance from shading. This is expected, as we are using deep CNNs, which are known to better

learn and utilize statistical priors present in the data than traditional optimization techniques.

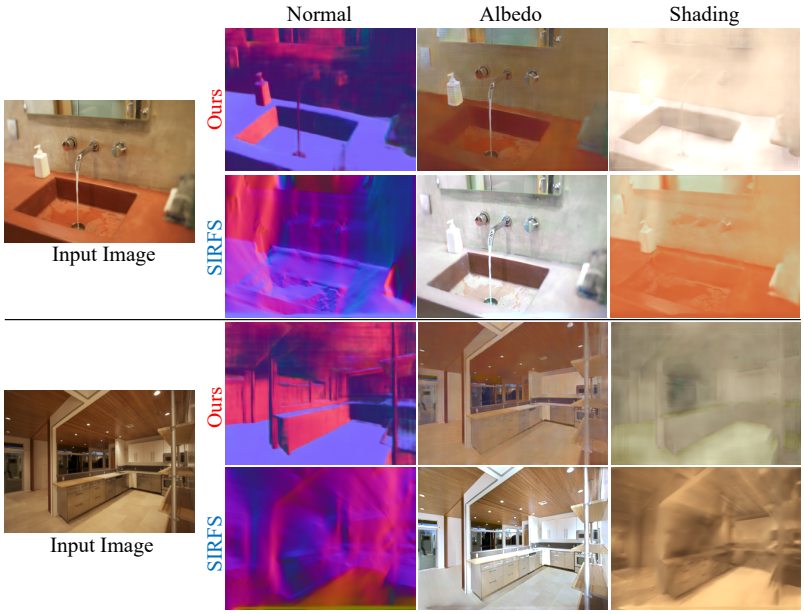


Figure 4-6: **Comparison with SIRFS [6].** Using deep CNNs our method performs better disambiguation of reflectance from shading and predicts better surface normals.

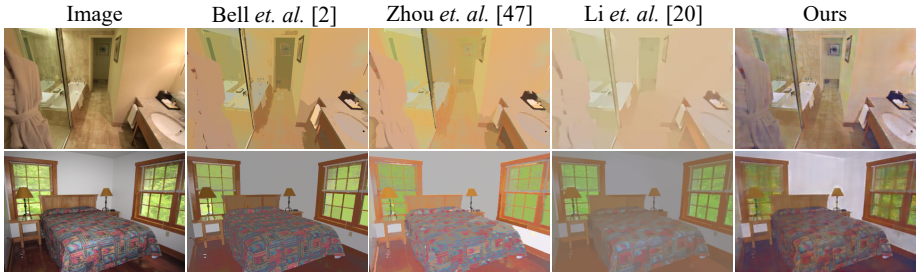


Figure 4-7: **Comparison with intrinsic image algorithms.** Our method seems to preserve more detailed texture and has fewer artifacts in the predicted albedo, compared to the prior works.

Comparison with intrinsic image decomposition algorithms.

Intrinsic image decomposition aims to decompose an image into albedo and shading, which is a sub-problem in inverse rendering. Several recent works [11, 186, 113, 95] showed promising results with deep learning. While our goal is to solve the complete inverse rendering problem, we still compare albedo prediction with these latest intrinsic image decomposition methods. We evaluate the WHDR (Weighted Human Disagreement Rate) metric [11] on the test set of the IIW dataset [11] and report the result in Table 4.1. As shown, we outperform these algorithms that

Table 4.1: **Intrinsic image decomposition on the IIW test set [11]**

Algorithm	Training set	WHDR
Bell *et. al.* [11]	-	20.6%
Li *et. al.* [96]	-	20.3%
Zhou *et. al.* [186]	IIW	19.9%
Nestmeyer *et. al.* [113]	IIW	19.5%
Li *et. al.* [95]	IIW	17.5%
Ours	IIW	**16.7%**

train on the original IIW dataset [11]. Since our goal is not intrinsic image decomposition, we do not train on additional intrinsic image specific datasets and avoid any post-processing as done in CGIntrinsics [95]. We also present a qualitative comparison of the inferred albedo with different existing algorithms in Figure 4-7 and with Li *et. al.* [95] in Figure 4-8. As shown, our method seems to preserve more detailed texture and has fewer artifacts in the predicted albedo, compared to the prior work.

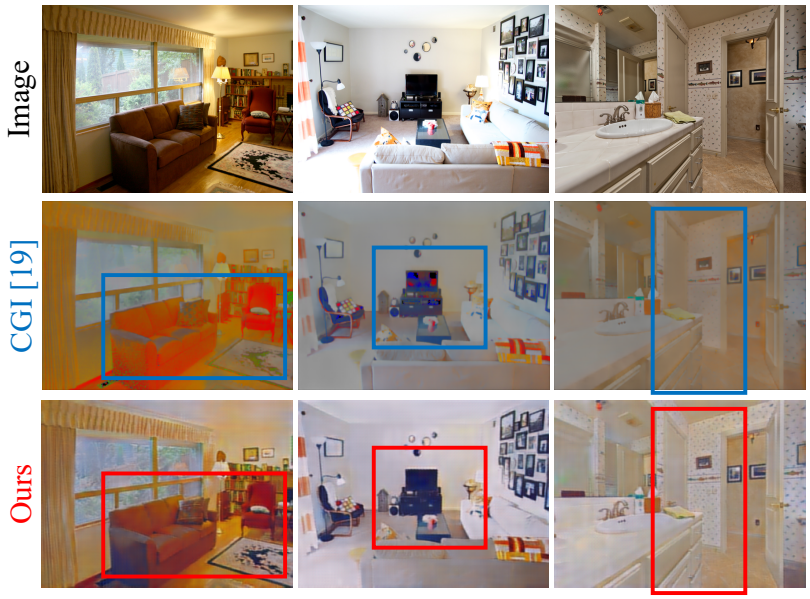


Figure 4-8: **Comparison with CGI (Li *et. al.* [95]).** In comparison with CGI [95], our method performs better disambiguation of reflectance from shading and preserves the texture in the albedo.

Evaluation of lighting estimation.

We estimate an environment map of low spatial resolution from an image. Although this is not the best representation of illumination, it can still capture the significant effects of illumination and

can be inferred jointly with other components. We present a qualitative evaluation of lighting estimation by inserting a diffuse hemisphere into the scene and rendering it with the inferred light from the image in Figure 4-9. We compare this with our implementation of the method proposed by Gardner *et. al.* [45], which also estimates an environment map from a single indoor image. $h_e(\cdot, \Theta_e)$ is a deep network that predicts the environment map given the image, normals, and albedo. 'GT+$h_e(\cdot)$' estimates the environment map given the image, ground-truth normals and albedo, and thus serves as an achievable upper-bound in the quality of the estimated lighting. 'Ours' estimates environment map from an image with IRN. 'Ours+$h_e(\cdot)$' predicts environment map by combining the inferred albedo and normals from IRN to predict lighting with $h_e(\cdot)$. Both 'Ours' and 'Ours+$h_e(\cdot)$' seem to produce more realistic environment maps and outperform Gardner *et. al.* [45]. 'Ours+$h_e(\cdot)$' improves lighting estimation over 'Ours' by utilizing the predicted albedo and normals to a greater degree.

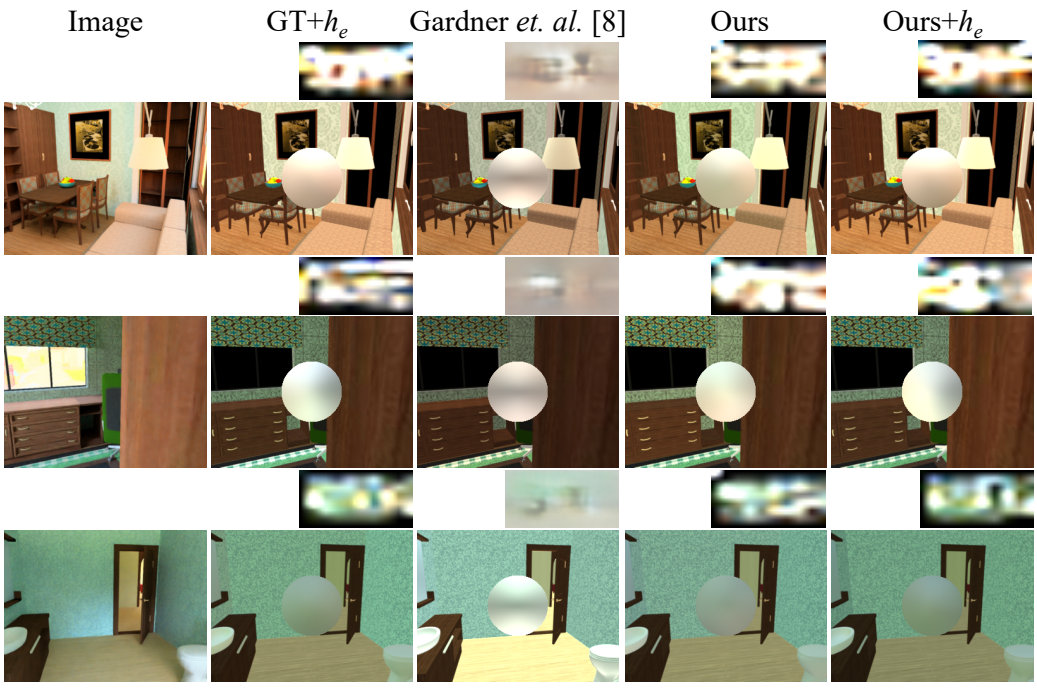


Figure 4-9: **Evaluation of lighting estimation.** We compare with our implementation of Gardner *et. al.* [45]. 'GT+$h_e(\cdot)$' predicts lighting conditioned on the ground-truth normals and albedo. 'Ours+$h_e(\cdot)$' predicts the environment map by conditioning it on the albedo and normals inferred by IRN.

Evaluation of normal estimation.

We also compare with PRBS [180] which predicts only surface normals from an image. Both PRBS and our model are trained on NYUv2 [111], and are tested on both NYUv2 and 7-scenes datasets [140]. As shown in Table 4.2, PBRS outperforms our method by about 2 degrees on NYUv2 dataset, and it is comparable to ours in the 7-scenes dataset. This shows that our joint decomposition network IRN-Diffuse generalizes well across datasets and performs comparably to the state-of-the-art normal prediction method PBRS.

Table 4.2: **Mean and median angular errors for surface normals**

Algorithm	NYUv2	7-scenes
PBRS [180]	**21.85°; 15.33°**	38.34°; 25.65°
Ours	23.89°; 16.92°	**37.75°; 24.54°**

Our results.

Figure 4-10 shows two examples of our results, with the albedo, glossiness segmentation, normal and lighting predicted by the network, as well as the reconstructed image with the direct renderer and the proposed Residual Appearance Renderer (RAR).

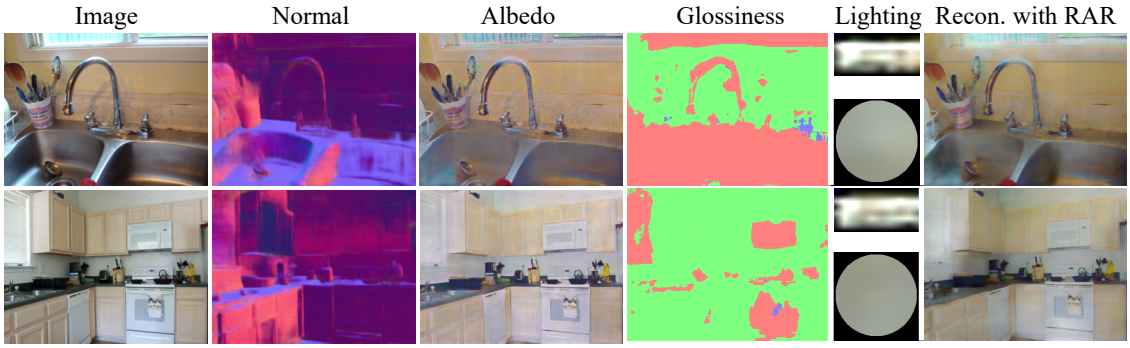


Figure 4-10: **Our Result.** We show the estimated intrinsic components; normals, albedo, glossiness segmentation (matte-blue, glossy-red and semi-glossy-green) and lighting predicted by the network, along with the reconstructed image with our direct renderer and the RAR.

4.5 Ablation Study

Role of the RAR in self-supervised training.

We have argued before that the RAR plays an important role in self-supervised training on un-

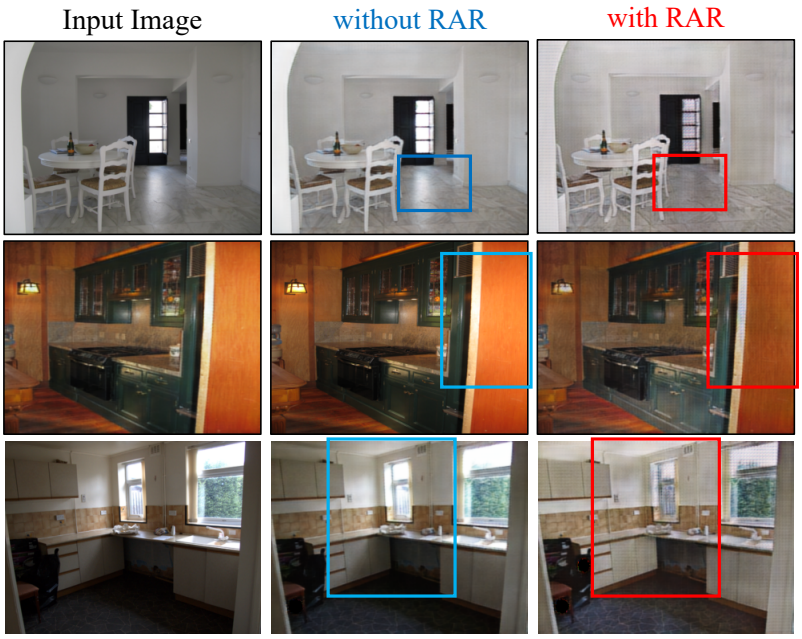


Figure 4-11: **Role of RAR in self-supervised training.** We train IRN on real data with and without RAR with self-supervision, and show the predicted albedo in column 2 and 3. The albedo predicted by training 'without RAR' fails to remove complex appearance effects like highlights, cast shadows and near-field lighting.

labeled real images, because the RAR captures complex appearance effects that cannot be modeled by a direct renderer. As shown in Figure 4-11, when being trained without the RAR, the network fails to separate such appearance effects from the estimated albedo, as it attributes these factors to the albedo in order to minimize the reconstruction loss. When the RAR is used during training, the network correctly removes such appearance effects from the albedo. We trained our network with and without RAR on the IIW dataset [11] (without any weak supervision), and we observed that using the RAR improves the quality of the albedo by reducing the WHDR metric from 37.4% to 32.8%. Thus, we conclude qualitatively and quantitatively that the RAR improves inverse rendering with self-supervised training on real data.

Role of weak supervision.

Our inverse rendering framework allows us to use weak supervision over intrinsic components whenever available. We train our method on the IIW dataset [11], which contains sparse pairwise relative reflectance judgments from humans. Training with this weak supervision significantly

Figure 4-12: **Role of weak supervision.** We predict more consistent albedo across large objects like walls, floors and ceilings using pair-wise relative reflectance judgments from the IIW dataset [11].

improves albedo estimation, making it more consistent across large objects like walls, floors, and ceilings as shown in Figure 4-12.

Role of albedo in glossiness segmentation.

We also show that conditioning IRN-Specular on the albedo predicted by IRN-Diffuse significantly improves glossiness segmentation. We train IRN-Specular with and without albedo as its input on our synthetic SUNCG-PBR dataset. Conditioning on albedo predicted by IRN-Diffuse as its input improves glossiness segmentation on real data by reducing cross-entropy loss from 0.76 to 0.62, which shows the benefits of joint multi-task learning in inverse rendering.

Part II

Enforcing Low-Rank Constraints

Chapter 5

Overview

In Part I we propose a data-driven technique for inverse rendering of faces and scenes from a single image. We leverage the huge amount of labeled synthetic and unlabeled real data to learn priors over intrinsic components (surface normals, reflectance and lighting) using a self-supervised reconstruction loss. In Part II, our goal is to explore the geometric and photometric constraints that connect multiple images of a scene. Specifically, we aim to perform inverse rendering from a few images by formulating a low-rank optimization problem that enforces the constraints that connect multiple images.

We study two specific forms of inverse rendering. First, we consider the problem of Structure from Motion, where given multiple images of a scene we aim to reconstruct the 3D scene and the cameras. Here we only consider the geometric aspect of inverse rendering. Next, we consider the problem of Uncalibrated Photometric Stereo, where given multiple images of an object captured under different illumination conditions, we aim to reconstruct the surface normals, albedo and illumination, assuming lambertian reflectance with point source lighting. In both of the above mentioned problems, we derive and enforce a low-rank constraint in an optimization framework.

Given a large number of images, Structure from Motion (SfM) and Uncalibrated Photometric

Stereo (UPS) can be solved by enforcing simple multi-view constraints. Since in the presence of a large number of images, the problem is highly over-constrained, we do not need to enforce any additional constraints to obtain an accurate solution. However, in presence of a small number of images, simple multi-view constraints are not enough to guarantee an accurate solution. Thus in this part, we derive stricter constraints and enforce them in a constrained low-rank optimization framework.

Low-rank optimization problems appear in many practical applications, when the data is low dimensional and only certain entries of the data have been observed. They have a diverse range of applications in machine learning, computer vision, bio-informatics [74], seismology [89] etc. In the case of machine learning, they have been applied for solving collaborative filtering [128, 87], the netflix problem [150, 146] , multi-label classification [22, 21] and label disambiguation [34]. In the case of computer vision, matrix completion problems have been used in Photometric Stereo (PS) [170, 3], Structure from Motion (SfM) [66, 62], image in-painting and completion [173, 125, 86, 100] and video denoising [69]. However in many such applications there are additional constraints along with the rank constraint, which makes the problem harder to solve. In this thesis we consider the low-rank matrix completion in the presence of additional constraints for solving Uncalibrated Photometric Stereo (UPS) and SfM.

Given a measurement matrix $\hat{M}$ of size $m \times n$ in which only certain entries of it are observed, defined by W (where $W_{ij} = 1$ indicates (i, j)-th entry of $\hat{M}$ is observed), in matrix completion problems we want to solve for a matrix M which is low rank, defined as :

$$\min_{M} \ \|W \odot (\hat{M} - M)\|_F \ \ s.t.\ M \text{ is low rank}, \tag{5.1}$$

In many computer vision applications, like Photometric Stereo and Structure from Motion, the rank

of the measurement matrix M is known beforehand. In this thesis we are particularly interested in this class of problems, where the rank of the matrix is known a priori, defined as :

$$\min_{M} \ \|W \odot (\hat{M} - M)\|_F \ \ s.t.\ rank(M) = r, \tag{5.2}$$

where $\odot$ represents element-wise dot product. In the presence of convex constraints we can define a low-rank constrained matrix completion problem as :

$$\min_{M} \ \|W \odot (\hat{M} - M)\|_F \ \ s.t.\ rank(M) = r,\ f(M) \leq 0,\ g(M) = 0, \tag{5.3}$$

where $f(.)$ and $g(.)$ are convex functions.

In this thesis we formulate and solve matrix completion problems in the presence of constraints in the context of UPS and SfM. In both applications we solve a general optimization problem similar to 5.3. we use Augmented Lagrangian Method of Multipliers (ADMM) [18, 50] to separately handle a constrained least square cost function and a rank constraint, by creating a copy of M as shown in 5.4.

$$\begin{aligned} \max_{\Gamma} \min_{M,N} \ & \|W \odot (\hat{M} - M)\|_F^2 + \frac{\tau}{2}\|N - M + \Gamma\|_F^2 \\ & s.t.\ M = N,\ rank(N) = r,\ f(M) \leq 0,\ g(M) = 0, \end{aligned} \tag{5.4}$$

where Γ is a matrix of Lagrange multipliers of the same size as M, and τ is a constant. The constraint $rank(N) = r$ can be handled in different ways using convex relaxation or singular value projection, which we will discuss later.

In Chapter 7 we introduce a novel rank constraint on collections of fundamental matrices in multi-view settings. We show that in general, with the selection of proper scale factors, a matrix

formed by stacking fundamental matrices between pairs of images has rank 6. Moreover, this matrix forms the symmetric part of a rank 3 matrix whose factors relate directly to the corresponding camera matrices. We use this new characterization to produce better estimations of fundamental matrices by optimizing an L1-cost function using Iterative Re-weighted Least Squares and Alternate Direction Method of Multiplier. The constraints of this problem relate to specific structure of multi-view geometry. We further show that this procedure can improve the recovery of camera locations, particularly in multi-view settings in which fewer images are available.

In Chapter 8, we introduce a new integrated approach for solving Uncalibrated Photometric Stereo (UPS). We perform 3D reconstruction of Lambertian objects using multiple images produced by unknown, directional point light sources. Under these assumptions, the collection of images produced by light sources is rank three. Traditionally, UPS is solved as a series of sub-problems, starting with SVD to produce a rank 3 decomposition into normals, albedos and lights. The surface normals provided by SVD are in general inconsistent with the partial derivatives of the surface (i.e. they are not *integrable*). Then additional steps are applied to recover an integrable set of surface normals and then to fit a surface to those normals. We show how to formulate a single optimization that includes rank and integrability constraints, allowing also for missing data. We then solve this optimization using the Alternate Direction Method of Multipliers (ADMM). We conduct extensive experimental evaluation on real and synthetic data sets. Our integrated approach is particularly valuable when performing photometric stereo using as few as 4-6 images, since the integrability constraint is capable of improving estimation of the linear subspace of possible solutions. We show good improvements over prior work in these cases.

Chapter 6

Background

6.1 Low-rank Matrix Completion

Researchers have worked on both theoretical and practical aspects of matrix completion problems in the past. An overview on matrix completion is given in [37]. There are several different techniques to perform matrix completion. We can broadly categorize them into methods that perform explicit factorization of the matrix and methods that introduce convex relaxation of the rank constraint.

Explicit factorization methods decompose a rank r matrix $\hat{M}$ of size $m \times n$ into LS, where L is $m \times r$ and S is $r \times n$. This leads to a non-convex optimization of the form :

$$\min_{L,S} \|W \odot (\hat{M} - LS)\|_F^2. \tag{6.1}$$

One common technique to solve a bi-convex problem of this nature is to perform alternate least squares [183]. In [20], the authors re-formulated the matrix completion problem as :

$$\min_{L,S} \|W \odot (\hat{M} - LS)\|_F^2 + \lambda_1 \|L\|_F^2 + \lambda_2 \|S\|_F^2. \tag{6.2}$$

The minimization is performed by standard Newton's algorithm [19] that includes a damping factor. Two drawbacks of this method are the extra memory required for Hessian computation and the choice of parameters λ_1 and λ_2. One commonly used technique to solve this problem, namely Wiberg's algorithm, as discussed in [116] uses an alternate optimization approach to solve (6.1). It first solves for L assuming S is fixed and then it updates S using Gauss-Newton update. In [117], the authors extended it to the damped-Wiberg's algorithm, which includes a damping factor or regularization in the Gauss-Newton update of the algorithm. In [17], the authors have re-cast the matrix completion problem as an unconstrained optimization on the Grassmann manifold and solve it using a gradient descent technique. Most of these methods are not guaranteed to find the global optimum. However in [81], the authors have shown an interesting performance guarantee on their explicit factorization based algorithm.

Another commonly used technique for solving matrix completion problems is to obtain a convex relaxation of the rank constraint and subsequently formulate a convex optimization. One such technique is to use the nuclear norm, i.e. the sum of singular values, of M as a relaxation. This translates to solving the following problem :

$$\min_{M} \|M\|_* \quad s.t.\ W \odot \hat{M} = W \odot M, \tag{6.3}$$

where $\|M\|_* = \sum_{i=1}^{min(m,n)} \sigma_i(M)$ is the nuclear norm. Fazel in [42] suggested that the optimization problem defined in (6.3) can be reformulated as semi-definite programming (SDP), which can be solved by an interior point method. However in many cases the matrix M is high-dimensional which makes the SDP problem intractable. Current SDP solvers can only efficiently solve a problem of size 100×100, which is too small for many computer vision applications.

Instead of solving a SDP problem, Cai *et. al.* in [24] proposed the Singular Value Thresholding

(SVT) algorithm which formulates a strongly convex relaxation of the matrix completion problem as follows :

$$\min_{M} \|M\|_* + \alpha\|M\|_F^2 \quad s.t.\ W \odot \hat{M} = W \odot M. \tag{6.4}$$

The SVT method applies a gradient descent method to the lagrangian dual of (6.4). To improve the convergence rate of SVT, [158] and [70] developed accelerated proximal gradient based techniques to solve for the regularized least square version of (6.4) :

$$\min_{M} \mu\|M\|_* + \frac{1}{2}\|W \odot (\hat{M} - M)\|_F^2. \tag{6.5}$$

To improve robustness in real world applications, the authors developed Robust Principle Component Analysis (Robust PCA) in [169], replacing least square regularization in (6.5) with an L1 regularization.

Authors in [26, 127, 126] have shown that in the absence of noise, the solution to nuclear norm relaxation based convex optimization produces global minima of the original non-convex problem. Some works [25, 41, 85, 112] have explored the inaccuracy of the solution of the convex relaxation formulation in the presence of noise. Authors of [13, 46] have shown that for positive semi-definite matrices matrix completion problems do not have any local minima in the absence of noise. In the presence of noise, all local minima are very close to the global minimum.

In more challenging practical situations, where the theoretical assumptions on the distributions of noise and missing data do not hold, it has been observed that nuclear norm relaxation for a rank r matrix leads also to reduction of the first r singular values, degrading the quality of the solution. The rank constraint treats all singular values of the matrix equally, whereas in nuclear norm relaxation singular values are treated differently by adding them. To counter this effect, the authors in [62] have introduced truncated nuclear norm relaxation, where all the singular values

except the top r are minimized. They define truncated nuclear norm regularization as a penalty of the form $f_{tnn}(M) = \sum_{i=r+1}^{min(m,n)} \sigma_i(M)$. However this leads to a non-convex optimization, which the authors have convexified using a series of majorizations. At each iteration, we replace $f_{tnn}(M)$ with a majorizer $f_{maj}(M)$. Specifically, at iteration k, let $M^{(k)} = U\Sigma V^T$ be the singular value decomposition of $M^{(k)}$, and let U_3 (and V_3) be the matrices containing the left (right) singular vectors corresponding to the three largest singular values of $M^{(k)}$. We then define

$$f_{maj}^{(k)}(M) = \|M\|_* - \text{trace}(U_3^T M V_3). \tag{6.6}$$

It was shown in [62] that $f_{maj}^{(k)}(M) \geq f_{tnn}(M)$ for all M and that $f_{maj}^{(k)}(M^{(k)}) = f_{tnn}(M^{(k)})$, and so decreasing f_{maj} leads to decreasing f_{tnn}. Thus at every iteration, (6.7) is solved using either ADMM or accelerated proximal gradient method.

$$\min_M \frac{1}{2}\|W \odot (\hat{M} - M)\|_F^2 + \mu f_{maj}^{(k)}(M) \tag{6.7}$$

In this thesis, I investigate a more general problem of matrix completion in the presence of convex constraints. This is similar to a class of problem termed Structured Low Rank Approximation (SLRA), where the goal is to find a low rank matrix that satisfies certain affine constraints and is closest to a given measurement. This can be expressed as :

$$\min_M \|\hat{M} - M\|_F \quad s.t.\ M \text{ is low rank},\ M \in \Omega, \tag{6.8}$$

where Ω is an affine subspace. Thus the solution M lies at the intersection of an affine subspace and a low-rank manifold. The matrix completion problem can be considered as a special case of SLRA, where the affine subspace is defined by the known entries of the matrix $\hat{M}$.

This problem is first formally introduced in [35]. Authors have discussed a wide range of applications of SLRA from signal enhancement, protein folding to computer algebra. Specifically in many applications the matrix of interest has a certain specific structure like the Henkel matrix (denoising procedure), Sylvester matrix (univariate approximation of GCDs) and Toeplitz matrix (signal processing). Based on Cadzow's method in [23] the authors devised an alternate projection based algorithm for solving SLRA problems. Alternating projection into the low rank space and the affine subspace leads to solution at their intersection and the algorithm converges to a local minima. Authors in [132] proposed a Newton-like iterative algorithm, which also performs alternate projection, and shows that it converges locally quadratically.

Another class of algorithm solves the problem by considering explicit factorization. Authors in [64] formulate this problem as :

$$\min_{L,S} \|\hat{M} - LS\|_F^2 + \lambda \, dist(LS, \text{closest structured matrix}). \tag{6.9}$$

They have devised two strategies where one of the constraints, i.e. either the rank or the affine subspace constraint, is always satisfied and the other constraint is satisfied upon convergence. Both of these optimization formulations are solved using alternate least squares and are shown below :

$$\min_{L,S} \|\hat{M} - LS\|_F^2 + \lambda \|LS - P_S(LS)\|_F^2, \tag{6.10}$$

$$\min_{L,S} \|\hat{M} - P_S(LS)\|_F^2 + \lambda \|LS - P_S(LS)\|_F^2, \tag{6.11}$$

where $P_S(LS)$ is the orthogonal projection of LS into an affine subspace S. In (6.10), the low rank constraint is always satisfied, whereas in (6.11) an affine subspace constraint is always satisfied.

Convex relaxation of the rank constraint, like nuclear norm, helps to formulate a convex optimization. In [52] the authors suggested a modified convex relaxation of the rank constraint and

formulate a semi-definite programming (SDP) technique to solve it. The authors in [135] have formulated a SDP problem using bi-convex relaxation for low-rank positive semi-definite matrices.

Another popular technique to tackle this problem is to use proximal methods to handle the affine/convex constraint and the rank constraint separately. In fact these methods are more general and can tackle any convex constraints. This idea is explained in (5.4) and is used in this thesis. In [181] authors used an ADMM algorithm to solve this problem and discussed its convergence. Similarly in [16] the author discussed an ADMM method for solving a structured low rank problem in chemistry. A proximal gradient based penalty approach is discussed in [174].

6.2 Structure from Motion (SfM)

Given a large number of images of a scene, several recent works [143, 2, 59] have proposed different incremental SfM techniques. These methods first recover camera matrices and structure from two images. Then, adding one image at a time, they apply bundle adjustment to estimate the camera matrix (and structure) of the new image. These methods aim to sequentially add images that can lead to stable expansion of the structure while dealing with noise and outliers. Recent works also attempt to further improve recovery by considering simultaneously subsets of multiple images and recovering camera matrices that are consistent over the entire subsets. Indeed a number of papers have focused on the consistent recovery of *either* camera orientation or location [4, 121, 120, 162, 167, 106].

Our work is related to a variety of approaches to structure from motion (SfM) that utilize rank constraints [67, 66, 109]. Tomasi and Kanade showed that under an orthographic projection, and after centering, projected points form a rank 3 matrix. Sturm and Triggs [147, 161] extended this to perspective projection by showing that projected points, when scaled properly, form a rank 4 matrix. Unlike their work, which uses a rank constraint on tracks of points in images, our work

only considers fundamental matrices and so in multiview settings it gives rise to systems with many fewer variables, relying on potentially less noisy estimates. Our approach, which seeks to recover a consistent set of fundamental matrices, is analogous to rotation or translation averaging and to loop closure [54, 32, 36]. In fact, obtaining consistent fundamental matrices can be regarded as simultaneous averaging of rotation, translation and camera calibration and as a way to close all loops. Our experiments indicate that such joint averaging performs better than a separate averaging of rotation and translation.

A number of algorithms have recently been proposed for solving unconstrained, low rank systems with outliers and missing data (e.g., [26, 63, 116]) with remarkable success. Extending such techniques to incorporate SfM constraints is an important next step.

6.3 Uncalibrated Photometric Stereo (UPS)

In this section we review research works related to photometric stereo. Belheumer *et al.* [10] showed that in UPS the integrable set of surface normals can only be recovered up to a Generalized Bas-Relief transformation (GBR). A number of recent papers have concentrated on methods of solving the GBR ambiguity. Researchers have used priors on the albedo distribution [3], reflectance extrema [40], grouping based on image appearance and color [137], inter-reflections [30], isotropy and symmetries [152], and specularity [38] as constraints while solving for the GBR. All of these methods have first used the above mentioned baseline described in Algorithm 2 to obtain a solution up to the GBR.

Recent works have explored a variety of other research directions in photometric stereo. Mecca *et al.* [107] proposed an integrated, PDE based approach to calibrated photometric stereo that uses a mere two images under perspective projection. It is not clear how to extend this to create an integrated approach for uncalibrated photometric stereo. Basri *et al.* [8] extended the baseline to

handle multiple light sources in each image using a spherical harmonics formulation. Chandraker *et al.* [29] proposed a method to handle attached and cast shadows in the case of multiple light sources per image. In [148] the authors determine the visibility subspace for a set of images to remove the cast and attached shadows for performing UPS. Various works have addressed non-Lambertian materials (e.g., Georghiades *et al.* [47] and Okabe *et al.* [115]).

In the context of Lambertian UPS, Georghiades *et al.* [47] proposed to remove shadows and specularities and recover the missing pixel values using matrix completion algorithms, e.g., using the damped Wiberg [118] or Cabral's algorithm [22]. Wu *et al.* [170] proposed a Robust PCA formulation as preprocessing for calibrated photometric stereo. Their approach seeks a low-rank (not necessarily rank 3) approximation to M while removing outlier pixels (corresponding to shadows and specularities). Oh *et al.* [114] applied Robust PCA in the context of calibrated photometric Stereo, replacing the Nuclear Norm with a Truncated Nuclear Norm (TNN) regularizer [62]. In [40], Favaro *et al.* have used Robust PCA as preprocessing to the baseline algorithm for UPS.

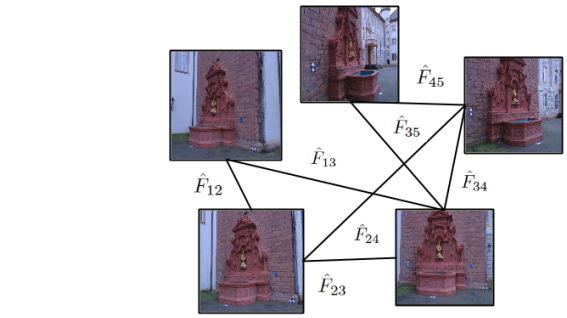
Chapter 7

Low-Rank Constraints in Structure from Motion

7.1 Introduction

Accurate reconstruction of 3D scenes from multiview stereo images is one of the primary goals of computer vision. Current techniques use point correspondences to estimate either the essential or fundamental matrices between pairs of images, and then use the estimated matrices to recover the camera matrices and structure. Notable success was achieved when sequential methods were introduced [2, 143]. These methods first recover camera matrices and structure from two images. Then, adding one image at a time, they apply bundle adjustment to estimate the camera matrix (and structure) of the new image. Recent work attempts to further improve recovery by considering simultaneously subsets of multiple images and recovering camera matrices that are consistent over the entire subsets. Indeed a number of papers have focused on the consistent recovery of *either* camera orientation or location [4, 121, 120, 162, 167, 106].

This paper introduces new constraints to enable the consistent recovery of fundamental and es-



$$\underbrace{\begin{bmatrix} \mathbf{0} & \hat{F}_{12} & \hat{F}_{13} & - & - \\ \hat{F}_{21} & \mathbf{0} & \hat{F}_{23} & \hat{F}_{24} & - \\ \hat{F}_{31} & \hat{F}_{32} & \mathbf{0} & \hat{F}_{34} & \hat{F}_{35} \\ - & \hat{F}_{42} & \hat{F}_{43} & \mathbf{0} & \hat{F}_{45} \\ - & - & \hat{F}_{53} & \hat{F}_{54} & \mathbf{0} \end{bmatrix}}_{\hat{F}} \approx \underbrace{\begin{bmatrix} \mathbf{0} & \lambda_{12} & \lambda_{13} & \lambda_{14} & \lambda_{15} \\ \lambda_{21} & \mathbf{0} & \lambda_{23} & \lambda_{24} & \lambda_{25} \\ \lambda_{31} & \lambda_{32} & \mathbf{0} & \lambda_{34} & \lambda_{35} \\ \lambda_{41} & \lambda_{42} & \lambda_{43} & \mathbf{0} & \lambda_{45} \\ \lambda_{51} & \lambda_{52} & \lambda_{53} & \lambda_{54} & \mathbf{0} \end{bmatrix}}_{\Lambda} \odot \underbrace{\begin{bmatrix} \mathbf{0} & F_{12} & F_{13} & F_{14} & F_{15} \\ F_{21} & \mathbf{0} & F_{23} & F_{24} & F_{25} \\ F_{31} & F_{32} & \mathbf{0} & F_{34} & F_{35} \\ F_{41} & F_{42} & F_{43} & \mathbf{0} & F_{45} \\ F_{51} & F_{52} & F_{53} & F_{54} & \mathbf{0} \end{bmatrix}}_{F}$$

$$\text{with } F = A + A^T \text{ and } \operatorname{rank}(A) = 3.$$

Figure 7-1: Illustration of our rank constraint. Collections of fundamental matrices $\{\hat{F}_{ij}\}$ estimated for pairs of images (top) are arranged in a matrix $\hat{F}$ (bottom). This matrix should be equal (up to noise) to a matrix F or properly scaled fundamental matrix, which in turn forms the symmetric part of a rank 3 matrix A.

sential matrices. This is potentially advantageous since those matrices capture simultaneously the location and orientation of the cameras, along (in the case of fundamental matrices) with their internal calibration parameters. For configurations of cameras that are not all collinear, our main result establishes that, when scaled properly, the matrix formed by appending all pairwise fundamental matrices in a multiview setting is of rank 6. More tightly, this matrix forms the symmetric part of a rank 3 matrix whose factors relate directly to the entries of the corresponding camera matrices. We further show that multiview settings of collinear cameras yield a rank 4 matrix.

We use this characterization to develop an optimization formulation for estimating consistent sets of fundamental matrices. Our formulation can accept sets of estimated fundamental matrices in which some are noisy, some are outliers, and some cannot be estimated at all from image pairs (i.e., missing data). In solving this optimization we seek a set of scaled fundamental matrices that satisfy our constraints and fit the estimated fundamental matrices. Our formulation uses an L1 cost function, which is optimized with Iterative Re-weighted Least Squares (IRLS) [61], to remove outliers, and uses Alternating Direction Method of Multipliers (ADMM) [18] to incorporate rank

constraints.

When thousands of images are available, existing methods that use pairwise epipolar constraints or tri-focal tensors can exploit highly over-determined systems to handle noise and outliers quite accurately. However, when fewer images are available the importance of rank constraints grows, and their introduction can potentially yield more accurate estimation of camera parameters. Indeed, we provide experiments that show that using our characterization, essential matrices can be estimated more accurately than with current state-of-the-art methods, and these in turn can be translated to better estimates of camera locations.

7.2 Low-Rank Characterization of Fundamental Matrices in Multi-view Settings

7.2.1 Background

We first introduce notations and give a short summary of the relevant concepts in multi-view geometry. An extensive discussion of this topic can be found in [55]. Let $I_1, ..., I_n$ denote a collection of n images of a scene and let $\mathbf{t}_i \in \mathbb{R}^3$ and $R_i \in SO(3)$ denote the location and orientation of the i'th camera in a global coordinate system. Let the 3×3 K_i denote the intrinsic camera calibration matrix for I_i. K_i is nonsingular and is typically specified in the form :

$$K_i = \begin{bmatrix} f_x & \alpha & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{7.1}$$

where, f_x and f_y respectively are the focal lengths in the x and y direction, (u_0, v_0) form the principal point and α represents the skew coefficient. Let $P = (X, Y, Z)^T$ be a scene point in the

global coordinate system. Its projection onto I_i (expressed in homogeneous coordinates) is given by $\mathbf{p}_i = P_i/Z_i$, where $P_i = (X_i, Y_i, Z_i)^T = K_i R_i^T (P - \mathbf{t}_i)$. We therefore associate with I_i the 3×4 camera matrix $C_i = K_i R_i^T \left[I, -\mathbf{t}_i \right]$, where I is a 3×3 identity matrix and note that scaling C_i does not affect projection.

Next, we consider the relations between pairs of images, I_i and I_j. We can express the camera rotation and translation relating two images by $R_{ij} = R_i^T R_j$ and $\mathbf{t}_{ij} = R_i^T (\mathbf{t}_i - \mathbf{t}_j)$. Clearly, $R_{ji} = R_{ij}^T$ and $\mathbf{t}_{ji} = -R_{ij}^T \mathbf{t}_{ij}$. Two images are further related by epipolar line constraints, which are expressed by $\mathbf{p}_i^T F_{ij} \mathbf{p}_j = 0$, where F_{ij} denotes the fundamental matrix relating I_i to I_j. F_{ij} can be estimated up to scale from point correspondences. F_{ij} is related to the rotation and translation between I_i and I_j and to their respective calibration matrices by $F_{ij} = K_i^{-T} [\mathbf{t}_{ij}]_\times R_{ij} K_j^{-1}$, where $[\mathbf{t}_{ij}]_\times$ denotes the skew-symmetric matrix corresponding to cross-product with $\mathbf{t}_{ij}$. In cases in which the cameras are calibrated we set $K_i = K_j = I$ and replace the fundamental matrix with the essential matrix $E_{ij} = [\mathbf{t}_{ij}]_\times R_{ij}$. Therefore, $F_{ij} = K_i^{-T} E_{ij} K_j^{-1}$.

To derive our rank constraint we will need to express the essential and fundamental matrices relative to a global coordinate system. [182] derived an expression in terms of the camera matrices C_i and C_j. Here we will use the more recent derivation of [4] that, as we shall see below, is amenable to factorization:

$$E_{ij} = R_i^T (T_i - T_j) R_j, \tag{7.2}$$

$$F_{ij} = K_i^{-T} R_i^T (T_i - T_j) R_j K_j^{-1}, \tag{7.3}$$

where $T_i = [\mathbf{t}_i]_\times$.

7.2.2 Low-rank Construction

We next introduce our main result, which includes a low rank characterization of the collection of fundamental matrices in multiview settings. For our result we will construct a matrix of size $3n \times 3n$, denoted F, in which each of the 3×3 blocks includes a fundamental matrix F_{ij} (see Figure 7-1), where we assume that each of the pairwise fundamental matrices in F is scaled properly. We further define $F_{ii} = 0$ for all $1 \leq i \leq n$, and note that this is consistent with (7.3). Likewise we define the $3n \times 3n$ matrix E from the essential matrices E_{ij}. We refer to F (resp. E) as the *multiview matrix of fundamentals (essentials)*.

Claim 1: F (and likewise E) is symmetric and $\mathrm{rank}(F) \leq 6$. Moreover,

1. If F is produced by n cameras whose centers are not all collinear then $\mathrm{rank}(F) = 6$ and there exists a $3n \times 3n$ matrix A with $\mathrm{rank}(A) = 3$ such that $F = A + A^T$.

2. If F is produced by n cameras whose centers are all collinear then $\mathrm{rank}(F) \leq 4$ and there exists a matrix A with $\mathrm{rank}(A) \leq 2$ such that $F = A + A^T$.

Proof: To prove the claim we begin by defining the matrix A as follows. Let $U_i = K_i^{-T} R_i^T T_i$, $V_i = K_i^{-T} R_i^T$, and $A_{ij} = U_i V_j^T$. U_i, V_i, and A_{ij} are 3×3 matrices. Observing (7.3) and recalling that T_i is skew-symmetric we see that $F_{ij} = A_{ij} + A_{ji}^T$.

Next we construct the $3n \times 3$ matrices U and V as

$$U = \begin{bmatrix} U_1 \\ \vdots \\ U_n \end{bmatrix} \text{ and } V = \begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix}$$

and set $A = UV^T$. Clearly, by construction, $\mathrm{rank}(A) \leq 3$. Moreover, $F = A + A^T$, and so F is symmetric and $\mathrm{rank}(F) \leq 6$.

Case 1: We show that unless the cameras are all collinear $\mathrm{rank}(A) = 3$. Clearly $\mathrm{rank}(V) = 3$. Therefore we need to show that also $\mathrm{rank}(U) = 3$. We prove this by contradiction. Assume $\mathrm{rank}(U) < 3$. Then $\exists\, \mathbf{t} \in \mathbb{R}^3$, $\mathbf{t} \neq \mathbf{0}$, s.t. $U\mathbf{t} = \mathbf{0}$. This implies that $\mathbf{t}_i \times \mathbf{t} = \mathbf{0}$ for all $1 \leq i \leq n$. Thus, all the $\mathbf{t}_i$'s are parallel to $\mathbf{t}$, violating our assumption that not all camera locations are collinear. Consequently $\mathrm{rank}(U) = 3$ and therefore also $\mathrm{rank}(A) = 3$.

Next we show that when the cameras are not all collinear $\mathrm{rank}(F) = 6$. We recall that $F_{ij} = K_i^{-T} E_{ij} K_j^{-1}$ where K_i and K_j are non-singular. We can therefore write $F = K^T E K$ where the $3n \times 3n$ matrix K is block diagonal with blocks formed by $\{K_i^{-1}\}_{i=1}^n$. This implies that $\mathrm{rank}(F) = \mathrm{rank}(E)$, and so we are left to show that $\mathrm{rank}(E) = 6$.

We assume WLOG that the camera locations are centered at the origin, i.e., $\sum_{i=1}^n \mathbf{t}_i = 0$ (since E is invariant to global translation of the cameras). We further argue that each column of U is orthogonal to each column of V. This is evident from the following identities

$$V^T U = \sum_{i=1}^{n} V_i^T U_i = \sum_{i=1}^{n} T_i = \left[\sum_{i=1}^{n} \mathbf{t}_i\right]_\times = 0_{3\times 3}. \tag{7.4}$$

Let $\tilde{A}$ denote the matrix A where we substitute $K_i = I, \forall i$ (so that $E = \tilde{A} + \tilde{A}^T$.) Denote by $\tilde{A} = \hat{U}\Sigma\hat{V}^T$ the SVD of $\tilde{A}$ ($\hat{U}$ and $\hat{V}$ are $3n \times 3$ and Σ is 3×3). Since $\tilde{A} = UV^T$ we have that $\mathrm{span}(U) = \mathrm{span}(\hat{U})$ and $\mathrm{span}(V) = \mathrm{span}(\hat{V})$. Now we can decompose E as :

$$\begin{aligned} E = \tilde{A} + \tilde{A}^T &= \hat{U}\Sigma\hat{V}^T + \hat{V}\Sigma\hat{U}^T \\ &= \begin{bmatrix} \hat{U} & \hat{V} \end{bmatrix} \begin{bmatrix} \Sigma & \\ & \Sigma \end{bmatrix} \begin{bmatrix} \hat{V}^T \\ \hat{U}^T \end{bmatrix} \end{aligned} \tag{7.5}$$

Since the columns of U are orthogonal to those of V, the matrix $\begin{bmatrix} \hat{U} & \hat{V} \end{bmatrix}$ is column orthogonal. Thus, (7.5) is the SVD of E. And since $\tilde{A}$ is rank 3, Σ is full rank. Consequently, $\mathrm{rank}(F) = \mathrm{rank}(E) = 6$.

Case 2: Suppose all camera centers are collinear. WLOG assume that the origin of the global coordinate system is also collinear with the n cameras (since F is unaffected by global translation), and so we can write $\mathbf{t}_i = \alpha_i \mathbf{t}$ for $1 \leq i \leq n$ where $\alpha_i \in \mathbb{R}$ and $\mathbf{t} \in \mathbb{R}^3$. Let $T = [\mathbf{t}]_\times$, then clearly $U_i = \alpha_i K_i^{-T} R_i^T T$. Define $\tilde{U}_i = \alpha_i K_i^{-T} R_i^T$ (so $U_i = \tilde{U}_i T$) and let the $3n \times 3$ matrix $\tilde{U}$ be formed by stacking $U_1, U_2, ...$ on top of each other then

$$A = UV^T = \tilde{U}TV^T.$$

Since T is skew-symmetric its rank is at most 2 and so is $\mathrm{rank}(A)$. It follows that $\mathrm{rank}(F) \leq 4$. ■

7.2.3 Tightness of our constraints

Claim 1 provides two constraints on the $3n \times 3n$ matrix F.

- $F = A + A^T$ and $\mathrm{rank}(A) = 3$.
- The diagonal block of F vanishes, i.e., $F_{ii} = 0$.

We now investigate how tight these constraints are in producing fundamental matrices that are consistent with a set of camera parameters. We show that the number of degrees of freedom allowed by these constraints is equal to the number of degrees of freedom in the camera matrices. However, we find that there exist matrices that are allowed by these constraints, but do not produce valid fundamental matrices.

Counting arguments show that our constraints allow $12n - 15$ degrees of freedom (DOFs) in defining F. Specifically, since A is rank 3 it can be written as $A = UV^T$ where U and V are $3n \times 3$, so together they have $18n$ entries. The constraint $F = A + A^T$, however, gives rise to a 15 DOF ambiguity that should be subtracted from the number of entries of U and V, as we explain in the next paragraph. The constraint that $F_{ii} = 0$ requires $U_i V_i^T$ to be skew symmetric, yielding $6n$ more

constraints on the entries of U and V, yielding together $12n - 15$ DOFs.

To calculate the DOFs in the ambiguity of $F = A + A^T$ note that we can write F as $F = [U,V]J[U,V]^T$, where J is a 6×6 permutation matrix defined as $J = \left[\begin{smallmatrix} 0 & I \\ I & 0 \end{smallmatrix}\right]$ (so $J[U,V]^T = [V,U]^T$). With this notation the ambiguity in factorizing F is obtained by introducing a 6×6 matrix Q such that $QJQ^T = J$ so that $[U,V]QJQ^T[U,V]^T = [U,V]J[U,V]^T = F$. Q has 36 entries, but the constraints $QJQ^T = J$ reduces its degrees of freedom to 15. Denote $Q = \left[\begin{smallmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{smallmatrix}\right]$ these constraints restrict the products $Q_{11}Q_{12}$ and $Q_{21}Q_{22}$ to be skew symmetric and the sum $Q_{11}Q_{22} + Q_{12}Q_{21} = I$, providing altogether 21 constraints on the 36 entries of Q, leaving 15 DOFs.

Coincidentally, the number of DOFs in factoring F is equal to the DOFs in defining n cameras. In general, the number of DOFs in defining n perspective cameras is $11n - 15$. However, each camera matrix can be scaled arbitrarily and each choice of scale will (inversely) scale the respective row and column of F. In other words, n camera matrices, $C_1, ..., C_n$, scaled arbitrarily by non zeros $1/s_1, ..., 1/s_n$, produce a collection of equivalent multiview fundamental matrices defined by

$$\{SFS | S = \text{diag}\{s_1, s_1, s_1, s_2, ..., s_n\}, s_i \neq 0\}.$$

The freedom in choosing the entries of S accounts for the n missing DOFs.

We note however that although the DOFs in factoring F with our constraints are equal to the DOFs in defining n camera matrices there exist matrices that satisfy our constraints but cannot be realized with n cameras. Specifically, these constraints do not guarantee that all the pairwise fundamental matrices F_{ij} are rank deficient. The constraint $F_{ii} = 0$ restricts $U_i V_i^T$ to be skew-symmetric, implying that either U_i or V_i is rank deficient. If all U_i's (or equivalently all V_i's) are chosen to be rank deficient then so are all the F_{ij}. If however some of the U_i's and some of the V_i's are chosen to be full rank then they may produce F_{ij} blocks that are rank 3 and so they are not legal

fundamental matrices. Note that the skew-symmetry of U_iV_i guarantees that no more than 1/4 of the F_{ij}'s can be of full rank. Indeed, our experiments (in Section 7.4) often produce F_{ij}'s that are near rank 2; in a typical run the average ratio of the third to second largest singular value $\approx 7 \times 10^{-8}$, presumably because the problem is so over-constrained.

In conclusion, while our constraints provide a necessary but not sufficient conditions for consistency, counting considerations indicate that our constraints are nearly tight. Below we develop an optimization scheme that utilizes these constraints to infer the missing scale factors for collections of estimated pairwise fundamental matrices, to recover missing fundamentals and to correct noisy ones.

7.3 Low-rank Constrained Optimization to Recover Fundamental Matrices

In this section we formulate an optimization problem that uses the constraints derived in section 7.2 to achieve a better recovery of pairwise fundamental matrices. Assume we are given a set of fundamental matrices $\hat{F}_{ij}$, where $(i,j) \in \Omega$ and Ω denotes the subset of image pairs for which fundamental matrices have been estimated. (We will further assume $(i,j) \in \Omega \implies (j,i) \in \Omega$.) We use these matrices to construct our measurement matrix $\hat{F}$ whose (i,j)'s 3×3 block contains $\hat{F}_{ij}$ if $(i,j) \in \Omega$ and is zero otherwise. Note that in the absence of errors each non-zero block is related by an unknown scale factor λ_{ij} to the corresponding block in the sought multiview matrix of fundamentals F, where λ_{ij} depends on the distance between the i'th and j'th cameras. Recovering these scale factors is essential in order to apply our constraints. Our task therefore can be expressed

as:

$$\min_{F,\{\lambda_{ij}\}} \sum_{(i,j)\in\Omega} \|\hat{F}_{ij} - \lambda_{ij} F_{ij}\|_F, \tag{7.6}$$

where F is constrained to satisfy Claim 1. Here we have chosen to minimize over the sum of Frobenius norms of each 3×3 block. Such mixed L1-L2 norm minimization is expected to be robust to outlier estimates of F_{ij}'s.

We note that the formulation (7.6) is bilinear in F and the scale factors. We could avoid this bilinearity by minimizing instead $\|\lambda_{ij}\hat{F}_{ij} - F_{ij}\|_F$. Such minimization, however, is subject to a zero trivial solution and so it requires an additional constraint such as $\sum_{ij} \lambda_{ij}^2 = 1$. Our experience with such a formulation is that it is quite sensitive to errors.

Expressing (7.6) with the constraints results in the following problem:

$$\begin{aligned} \min_{A,\{\lambda_{ij}\}} \quad & \frac{1}{2} \sum_{(i,j)\in\Omega} \|\hat{F}_{ij} - \lambda_{ij}(A_{ij} + A_{ji}^T)\|_F \\ \text{s.t.} \quad & \operatorname{rank}(A) = 3,\ A_{ii} + A_{ii}^T = 0,\ \lambda_{ij} = \lambda_{ji} \end{aligned} \tag{7.7}$$

where A_{ij} denotes each 3×3 sub-block of A. Our solution for F then is $F = A + A^T$.

(7.7) introduces a number of challenges, including the mixed L1-Frobenius norms, the bilinearity, and the rank constraint. This problem is non-convex due to the latter two challenges. Below we describe how we approach these challenges with IRLS and ADMM. Our algorithm is summarized in Algorithm 1.

7.3.1 Handling Outliers with IRLS

We begin by addressing the mixed L1-Frobenius norm in the cost function. We approach this with Iterative Re-weighted Least Squares (IRLS) [61]. IRLS converts the problem to weighted least

squares where the weights are updated from one iteration to the next. At each iteration t of the IRLS we replace the cost function in (7.7) with

$$\min_{A,\{\lambda_{ij}\}} \frac{1}{2} \sum_{(i,j)\in\Omega} w_{ij}^t \|\hat{F}_{ij} - (A_{ij} + A_{ji}^T)\lambda_{ij}\|_F^2, \tag{7.8}$$

where

$$w_{ij}^t = \begin{cases} 1/\max(\delta, \|\hat{F}_{ij} - \lambda_{ij}^{t-1}(A_{ij}^{t-1} + (A_{ji}^{t-1})^T)\|_F), & \text{if } (i,j) \in \Omega \\ 0, & \text{otherwise.} \end{cases}$$

δ is a regularization parameters (we use $\delta = 10^{-3}$).

To clarify presentation we simplify our notations as follows. Let W and Λ be $3n \times 3n$ matrices. Denoting their 3×3 sub-blocks by W_{ij} and Λ_{ij}, we set $W_{ij} = w_{ij}\mathbf{1}$ and $\Lambda_{ij} = \lambda_{ij}\mathbf{1}$, where $\mathbf{1}$ is a 3×3 matrix with all 1's. We further use the subscript *WF* to denote the weighted Frobenius norm, i.e., $\|\mathbf{v}\|_{WF}^2 = \mathrm{trace}(\mathbf{v}^T W \mathbf{v})$ and use $\odot$ to denote element-wise product of matrices. Therefore, in each IRLS iteration we seek to solve

$$\begin{aligned} \min_{A,\Lambda} \quad & \frac{1}{2}\|\hat{F} - \Lambda \odot (A + A^T)\|_{WF}^2 \\ \text{s.t.} \quad & \mathrm{rank}(A) = 3,\ A_{ii} + A_{ii}^T = 0,\ \Lambda_{ij} = \lambda_{ij}\mathbf{1},\ \lambda_{ij} = \lambda_{ji}. \end{aligned} \tag{7.9}$$

7.3.2 Optimization using ADMM

Next, we wish to solve the non-convex optimization problem in (7.9), including the bilinearity and the rank constraint. To this end we will use a scaled version of Alternating Direction Method of Multipliers (ADMM) [18, 50]. We maintain a second copy of A, which we denote as B, and form

the augmented Lagrangian of (7.9) as:

$$\max_{\Gamma} \min_{A,B,\Lambda} \quad \frac{1}{2}\|\hat{F} - \Lambda \odot (A + A^T)\|_{WF}^2 + \frac{\tau}{2}\|B - A + \Gamma\|_F^2$$
$$\text{s.t. } \operatorname{rank}(B) = 3,\ A_{ii} + A_{ii}^T = 0,\ \Lambda_{ij} = \lambda_{ij}\mathbf{1},\ \lambda_{ij} = \lambda_{ji}. \tag{7.10}$$

The last term in this objective, $\frac{\tau}{2}\|B - A + \Gamma\|_F^2$ denotes the Lagrangian penalty; τ is a constant, and Γ is a matrix of Lagrange multipliers of the same size as A that is updated in the ADMM steps. We next describe the ADMM steps, which are applied iteratively.

Step 1: Solving for (A, Λ).

In each iteration, k, we solve the following sub-problems:

$$\min_{A,\Lambda} \quad \frac{1}{2}\|\hat{F} - \Lambda \odot (A + A^T)\|_{WF}^2 + \frac{\tau}{2}\|A - (B + \Gamma)\|_F^2$$
$$\text{s.t. } A_{ii} + A_{ii}^T = 0,\ \Lambda_{ij} = \lambda_{ij}\mathbf{1},\ \lambda_{ij} = \lambda_{ji}. \tag{7.11}$$

Since (7.11) is non-convex we will solve it by alternative minimization of A and Λ

1. Optimize w.r.t. A:

   Because of the form of (7.11) it is useful to separate A into its symmetric and anti-symmetric parts, A_s and A_n, so that $A = \frac{1}{2}(A_s + A_n)$ with $A_s = A + A^T$ and $A_n = A - A^T$. Let $G = B + \Gamma$; G_s and G_n respectively denote its symmetric and anti-symmetric part. We can write (7.11) in

terms of A_s and A_n and separately solve for them as follows:

$$A_s^{(k+1)} = \underset{A_s}{\operatorname{argmin}} \frac{1}{2}\|\hat{F} - \Lambda^{(k)} \odot A_s\|_{WF}^2 + \frac{\tau}{8}\|A_s - G_s^{(k)}\|_F^2 \ \text{ s.t. } \ (A_s)_{ii} = 0, \tag{7.12}$$

$$A_n^{(k+1)} = \underset{A_n}{\operatorname{argmin}} \frac{\tau}{8}\|A_n - G_n^{(k)}\|_F^2 = \ G_n^{(k)}. \tag{7.13}$$

To solve (7.12) we take the derivative w.r.t. A_s and equate to 0. Thus we update A_s according to

$$A_s^{(k+1)} = W \odot \Lambda^{(k)} \odot \hat{F} + \frac{\tau}{4} G_s^{(k)} \oslash (W \odot \Lambda^{(k)} \odot \Lambda^{(k)} + \frac{\tau}{4}) \tag{7.14}$$

$$(A_s^{(k+1)})_{ii} = 0 \tag{7.15}$$

where $\oslash$ denotes element-wise division.

2. Optimize w.r.t. Λ: We minimize the following sub-problem

$$\Lambda^{(k+1)} = \underset{\Lambda}{\operatorname{argmin}} \ \|\hat{F} - \Lambda \odot A_s^{(k+1)}\|_{WF}^2 \quad \text{s.t. } \ \Lambda_{ij} = \lambda_{ij}\mathbf{1}, \ \lambda_{ij} = \lambda_{ji}. \tag{7.16}$$

We can solve (7.16) separately for each block as follows,

$$\lambda_{ij}^{(k+1)} = \underset{\lambda_{ij}}{\operatorname{argmin}} \ \|\hat{F}_{ij} - \lambda_{ij}(A_s^{(k+1)})_{ij}\|_{WF}^2, \ \ i < j = \frac{\operatorname{trace}(\hat{F}_{ij}^T (A_s^{(k+1)})_{ij})}{\|(A_s^{(k+1)})_{ij}\|_F^2} \tag{7.17}$$

Note that $\lambda_{ii}^{(k+1)} = 0$, $\lambda_{ji}^{(k+1)} = \lambda_{ij}^{(k+1)}$ and $\Lambda_{ij}^{(k+1)} = \lambda_{ij}^{(k+1)}\mathbf{1}$.

Step 2: Solving for B.

This part of the ADMM deals with the rank constraint. It requires a solution to

$$B^{(k+1)} = \underset{B}{\operatorname{argmin}} \frac{\tau}{2}\|B - A^{(k+1)} + \Gamma^{(k)}\|_F^2$$
$$\text{s.t. } \operatorname{rank}(B) = 3. \tag{7.18}$$

This is solved by

$$B^{(k+1)} = SVP(A^{(k+1)} - \Gamma^{(k)}, 3), \tag{7.19}$$

where $SVP(X, r)$ denotes the Singular Value Projection (SVP) of X into space the of rank-r matrices. To perform $SVP(X, r)$ we compute the SVD of X and keep its top r singular values and the corresponding singular vectors.

Step 3: Update of Γ. The matrix Γ contains Lagrange multipliers that are used in the saddle-point formulation (7.10) to enforce the equality constraint $A = B$. The following update is a gradient ascent step that acts to maximize the augmented Lagrangian (7.10) for Γ. For details, see [18, 50].

$$\Gamma^{(k+1)} = \Gamma^{(k)} + (B^{(k+1)} - A^{(k+1)}). \tag{7.20}$$

Empirically we observe monotonic convergence of the cost function defined in Equation 7.7 with each iteration of IRLS on a sample problem as shown in Figure 7-2. For every iteration of the IRLS we run ADMM till convergence to optimize Equation 7.10.

Algorithm 1 IRLS-ADMM solver

Input: Estimated fundamentals in $\hat{F}$ and Ω.
Output: Recovered F.
IRLS: Solve (7.7)
Initialize Λ and A .
Create weights for IRLS, $w_{ij}^0 = 1$ if $(i, j) \in \Omega$ and $w_{ij}^0 = 0$ otherwise. Set $t = 1$.
while not converged **do**
  Solve (7.8) using ADMM formulation (7.10).
    Set $k = 0$, $\tau = \sum w_{ij}$, $\Gamma^0 = 0$, $B = A$.
    while not converged **do**
      Alternative minimization of (7.11).
        Update A using (7.13) and (7.15).
        Update Λ using (7.17).
      Update B using (7.19) .
      Update Γ using (8.27) .
      $k = k + 1$.
    end while
  Update Weights w_{ij}^t using (7.8).
  $t = t + 1$.
end while
$F = A + A^T$.

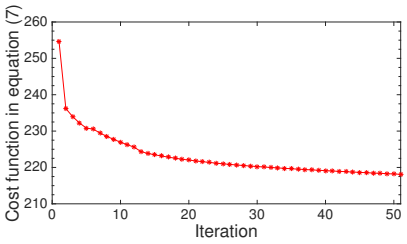


Figure 7-2: Convergence of our optimization algorithm. The cost function is defined in (7.7).

7.4 Experiments

To demonstrate the utility of our method we tested it in the problem of estimating essential matrices and camera locations from multiple images. Current iterative and global approaches to Structure from Motion (SfM) are often tested on large datasets when many pairwise essential matrices can be estimated, achieving outstanding performance. We argue that imposing rank constraints can be useful particularly when the number of images is relatively small. To demonstrate this we run our method on subsets of images of different sizes showing improved performance relative to the existing methods particularly with smaller subsets.

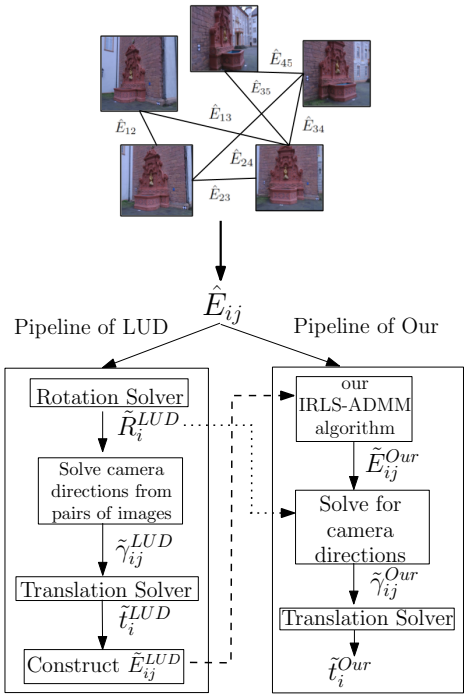


Figure 7-3: SfM pipelines for LUD (left) and our method (right).

We next describe the tested methods:

- **LUD [120]**: Figure 7-3 shows the pipeline used by LUD to estimate camera locations and orientations from pairs of images. Starting from pairwise essential matrices estimated with SIFT [105] and RANSAC [15], this method first solves for camera orientations, denoted by $\tilde{R}_i^{\text{LUD}}$ in Figure 7-3, by iteratively applying [4] while rejecting outliers. Using camera orientations it then returns to the image keypoints to estimate pairwise camera directions, denoted by $\tilde{\gamma}_{ij}^{\text{LUD}}$. Using these pairwise directions it applies IRLS to solve for camera locations ($\tilde{t}_i^{\text{LUD}}$), which we compare to our method. In addition, we use the estimated camera locations and orientations to reconstruct the pairwise essential matrices $\tilde{E}_{ij}^{\text{LUD}}$.

- **ShapeKick [49]**: For this method we use the same pipeline as used with LUD, except that we replace the translation recovery part of LUD with ShapeKick. ShapeKick formulates the location recovery problem as a convex optimization and solves it with ADMM. They achieved

comparable performance to LUD on the dataset of [167].

- **1DSfM [167]**: This method uses a pre-processing technique, based on projection in many random directions, to remove outliers in the original pairwise direction measurements. In our experiments we use their software, which uses the pipeline described in [167] and only provides camera locations.

- **Our method**: Figure 7-3 shows the pipeline used by our method. From the pair-wise essential matrices we minimize (7.7) using the IRLS-ADMM summarized in Algorithm 1. Since our method is not convex it requires a good initialization. We initialize it with essential matrices produced by the LUD method of Ozyesil *et al.* [120], denoted $\tilde{E}_{ij}^{\mathrm{LUD}}$. Specifically $\tilde{E}_{ij}^{LUD}$ is used to initialize Λ and A in Algorithm 1. Our algorithm improves these essential matrix estimates, producing a collection of new pairwise estimates in E, denoted $\tilde{E}_{ij}^{\mathrm{Our}}$. To further produce camera locations we first use $\tilde{E}_{ij}^{\mathrm{Our}}$ and the rotations obtained by the LUD pipeline, $\tilde{R}_i^{\mathrm{LUD}}$, to solve for the pairwise camera directions $\tilde{\gamma}_{ij}^{\mathrm{Our}}$. Then we apply the translation solver of LUD to the $\tilde{\gamma}_{ij}^{\mathrm{Our}}$ with $(i,j) \in \Omega$ to produce camera locations $\tilde{t}_i^{\mathrm{Our}}$. As is shown below, our improved estimates of essential matrices lead in turn to improved estimates of camera locations compared to the LUD pipeline.

We tested these methods on real image collections from [167], which comes with 'ground truth' estimates of camera locations and essential matrices produced with a sequential method similar to [143]. (These ground truth estimates are used also in [167, 120, 49].) For our experiments we used 14 different scenes from the dataset. For each scene we randomly selected 5 different sub-samples of N images from the dataset. We used N = 50, 100, and 150 images, resulting in 70 different trials for each N. In each trial we compared the quality of the essential matrix recovered by our method to that recovered by LUD and ShapeKick. Likewise, we compared the quality of our recovered

camera locations to those obtained by the three competing methods.

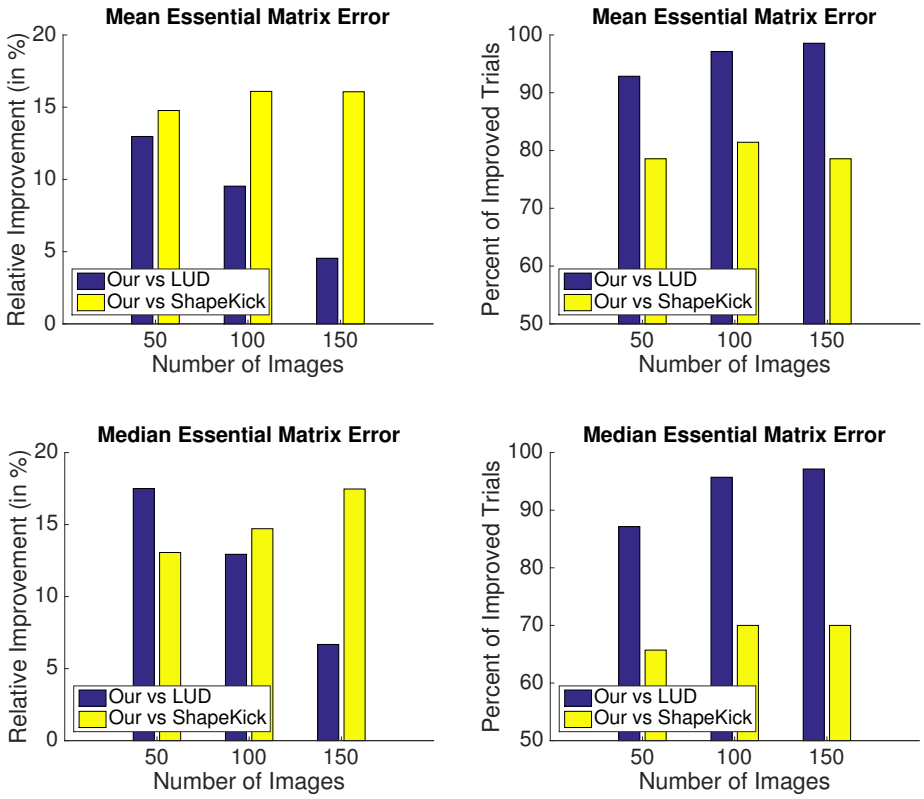


Figure 7-4: These graphs show a comparison of the recovery error of essential matrices achieved with our method compared to LUD (in blue) and ShapeKick (in yellow), for collections of 50, 100, and 150 images from [167], The graphs on the left show the amount of relative improvement and the ones on the right show the fraction of improved trials.

Figures 7-4-7-5 show our results. Each graph summarizes the results of 70 trials with each value of N. Figure 7-4 shows the quality of our essential matrix estimates compared to those obtained with LUD and ShapeKick, and Figure 7-5 shows the quality of our camera location estimates compared to those achieved by the three competing algorithms. We measure these as follows. In each experiment k we consider the collection of pairwise essential matrices produced by our method. We first normalize each matrix and measure its error to the respective (normalized) ground truth matrix. We then take the mean (or median) of this error over all essential matrices. Denote this error by e_k^{Our}. We then produce similar error measures for each competing algorithm, denoted e_k^{Other}. We then report:

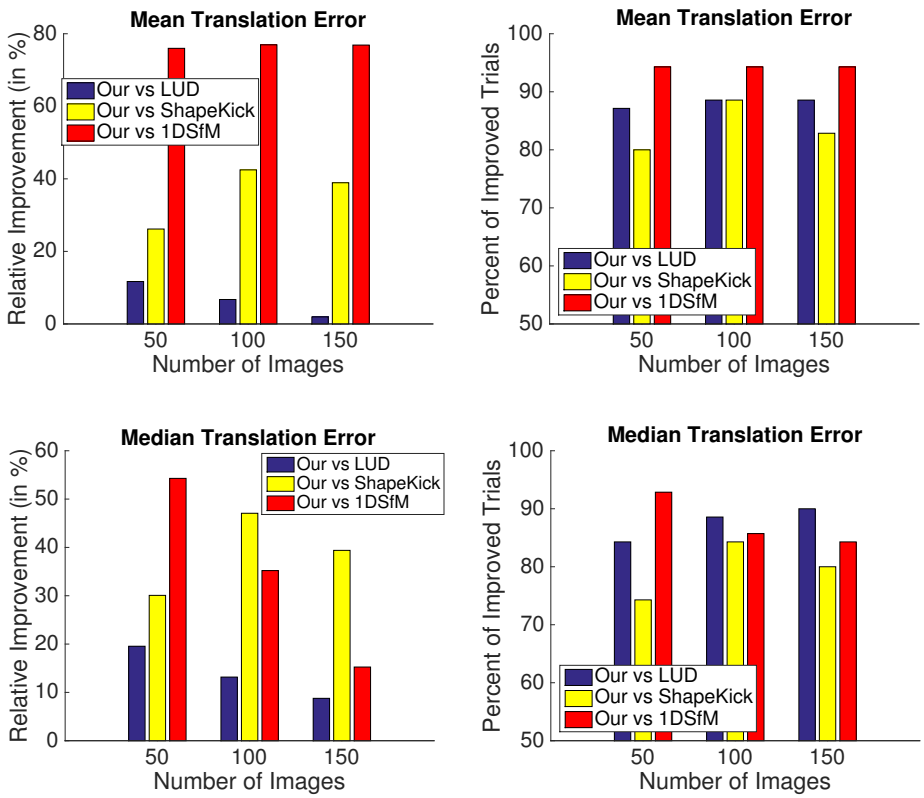


Figure 7-5: A comparison of the recovery error of camera locations achieved with our method compared to LUD (in blue) and ShapeKick (in yellow), and 1DSfM (in red) for collections of 50, 100, and 150 images from [167], The graphs on the left show the amount of relative improvement and the ones on the right show the fraction of improved trials.

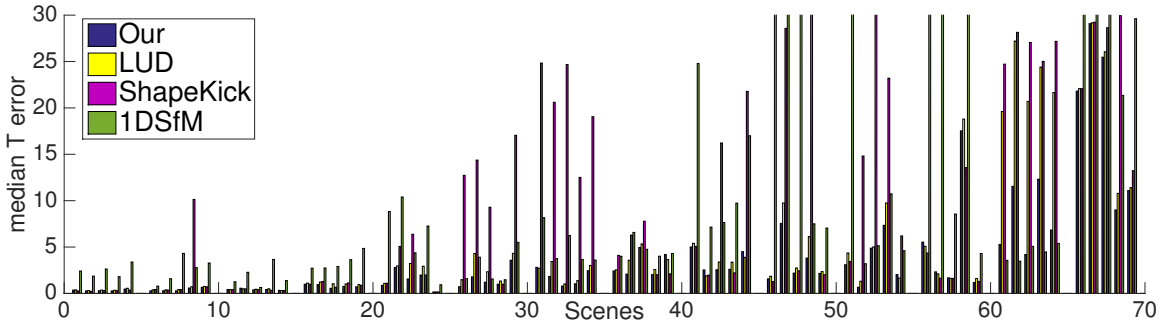


Figure 7-6: Median camera location error obtained by the four algorithms for 5 subsets of 50 images for 14 different scenes ('Notre Dame', 'Montreal Notre Dame', 'Alamo', 'Piazza del Popolo', 'Piccadilly', 'NYC Library', 'Yorkminster', 'Union Square', 'Madrid Metropolis', 'Tower of London', 'Vienna Cathedral', 'Roman Forum' and 'Ellis Island', 'Gendarmenmarkt'). For clarity we terminate the median T error axis at 30.

- **Relative Improvement (in %)**: Here we report for each N and competing algorithm the average of $\frac{e_k^{\text{Other}} - e_k^{\text{Our}}}{e_k^{\text{Other}}}$ over all experiments.

- **Percent of Improved Trials**: This provides the percentage of trials in which our algorithm achieved more accurate results than a competing algorithm, i.e., $\frac{1}{K}\sum_{k=1}^{K}\mathbb{I}(e_k^{\text{Our}} < e_k^{\text{Other}})$, where $\mathbb{I}(.)$ is the indicator function and K denotes the total number of trials.

We provide similar measures to assess the quality of our camera locations estimates. In Figure 7-6 we further show the median error of camera location estimates for all methods in all trials for N = 50.

It can be seen overall that our method leads to improved estimation of essential matrices and of camera locations. With 50 images, compared to, e.g., LUD, our algorithm improves the median essential matrix estimates by 17.69%. With 150 images a smaller overall improvement of 6.68% is achieved. This suggests that our constraints are more effective when smaller numbers of images are used. Interestingly, however, despite this reduction the fraction of trials in which our method achieved more accurate estimates compared to LUD in fact increased slightly from 87% with 50 images to 98% with 150 images, indicating that our method remains effective also with larger number of images (albeit yielding smaller improvement). Similar results are observed for camera location estimation. With 50 and 150 images our algorithm improves the median camera location error by 19.73% and 8.77% respectively, while the fraction of trials in which our method achieved more accurate estimates than LUD increased slightly from 84% with 50 images to 90% with 150 images.

In our previous experiments we applied our optimization algorithm to essential matrices, assuming calibration is given. Below we further apply our algorithm to fundamental matrices in an uncalibrated setting. Since not all the entries of a 3×3 fundamental matrix are of same orders of magnitude, we normalize each of the input pairwise fundamental matrices by centering all the images and scaling them uniformly to within the $[1, 1]$ square and then compute a normalized fundamental matrix. This does not affect our rank constraint and can be inverted at the end of the process. We tested our method on 5 subsamples of 50 images for 14 different scenes and compared

it to LUD. To evaluate the quality of the recovered fundamental matrices we convert them to essential matrices by applying the known calibration matrices and further use these to recover camera locations. The results can be seen in Figure 6. Using our method to recover fundamentals (in blue) yielded comparable accuracies to our results for essential matrix recovery (yellow) and both our approaches improve significantly (10-20%) over LUD as shown in Figure 7-7.

Figure 7-7: Improvement of our method over LUD using fundamental matrix (in blue) and essential matrix (yellow) for 50 images.

We further performed bundle adjustment (using [104]) initialized by the camera parameters obtained with our method and LUD. After bundle adjustment compared to LUD our method improved camera location estimates on average by 11.52%, 3.13% and 5.43%, improving in 70.59%, 64.29% and 63.77% of all trials for 50, 100 and 150 images respectively in terms of median translation error. These results indicate that our method maintains improved accuracies over LUD also after bundle adjustment.

With 50 images the recovery of essential matrices with our method requires roughly 20 iterations of IRLS and 1000 iterations of ADMM. These take overall about 2 minute on a 2.7 GHz Intel Core i5 computer.

To conclude, these experiments indicate that our characterization of essential matrices in multiview settings can be used to improve essential matrix and cameral location estimates. The advantage of these constraints appear to be particularly pronounced when fewer images are available.

Chapter 8

Low Rank Constraints in Photometric Stereo

8.1 Introduction

Uncalibrated photometric stereo (UPS) is the problem of recovering the 3D shape of an object and associated lighting conditions, given images taken with varying, unknown illumination. We assume that we view a lambertian object in multiple images from a fixed viewpoint. In each image the object is illuminated by a single, distant point light source. We represent lighting in image i with $l_i \in R^3$, in which the direction of l_i represents the direction to the lighting, and $\|l_i\|$ represents its magnitude. We represent the object using a set of surface normals $\hat{n}_j \in R^3$, and albedos $\rho_j \in R$ for each pixel. We then obtain images with the equation:

$$M_{ij} = \max(0, \rho_j l_i^T \hat{n}_j) \tag{8.1}$$

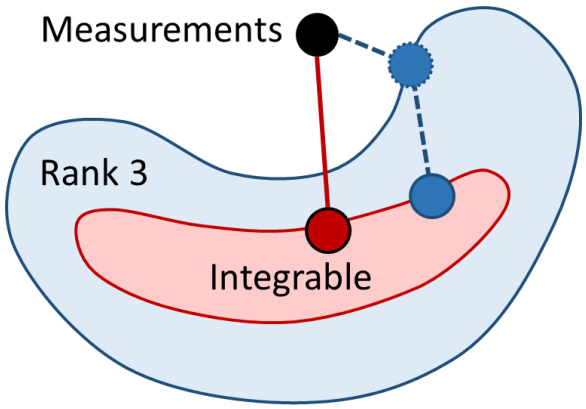


Figure 8-1: A cartoon of our approach. Blue represents the set of rank 3 matrices, while red represents the subset of those that correspond to integrable surfaces. Our optimization seeks to find the integrable matrix (red dot) that is closest to the measurements (black dot). If instead we first find the nearest rank 3 matrix and then select an integrable matrix (the blue dots) we may produce a suboptimal solution.

where M_{ij} represents the j-th pixel of the i-th image. We define the surface normal $\hat{n}_j = \frac{n_j}{\|n_j\|}$, $n_j = (-z_x, -z_y, 1)^T$, where z_x and z_y represent partial derivatives of the surface $z(x, y)$ at pixel j. Negative values of $\rho_j l_i^T \hat{n}_j$ are set to 0; these appear as attached shadows.

We now describe the creation of all images using matrix operations. We define S to be a $3 \times p$ matrix in which column j contains $\rho_j \hat{n}_j$. Given m images, we can stack the light into the matrix L of dimension $m \times 3$, where each row denotes one light per image. We concatenate all the images to form an observation matrix M of dimension $m \times p$, where p is the number of pixels. Now, in the absence of shadows, we can write the equation of UPS as:

$$M = LS. \tag{8.2}$$

Classical work on photometric stereo (e.g. [168], see a recent review in [1]) has assumed that known lighting is obtained by careful calibration. With L known, equation (8.2) can be solved as a linear least squares problem. A more general and challenging case is unconstrained photometric stereo, in which the L is unknown. A common approach, which we use as a baseline algorithm, follows the steps in Algorithm 2.

Algorithm 2 Baseline

Input : M
Output : Z
Factorization : Perform SVD on M to obtain light and scaled surface normals $M = \tilde{L}\tilde{S}$ [56].
Integrability : Follow Yuille and Snow [176] to resolve ambiguity after the factorization using integrability. In $M = LS = \tilde{L}A^{-1}A\tilde{S}$, we solve for A, such that $S = A\tilde{S}$ approximately forms a set of integrable surface normals.
Depth Reconstruction : Obtain the depth map Z from the set of integrable surface normals S as, e.g, in [8] or [43].

In this chapter, I show how we replace the existing pipeline for solving UPS with an integrated approach. Existing methods, pioneered by [56], formulate UPS as the problem of finding a low-rank factorization of the measurements. Specifically, given m images each with p pixels, let M denote the $m \times p$ matrix containing the pixel intensities. These methods optimize

$$\min_{\hat{M}} \|\hat{M} - M\|_F^2 \quad \text{s.t.} \quad \text{rank}(\hat{M}) = 3. \tag{8.3}$$

This problem can be solved by SVD, from which we produce a family of solutions, each consisting of a set of light sources, albedos, and surface normals. These solutions are related by a 3×3 ambiguity matrix. The surface normals provided by SVD are in general inconsistent with the partial derivatives of the surface (i.e. they are not *integrable*). Consequently, existing methods apply an additional sequence of steps aimed at reducing the ambiguity and fitting a surface to the recovered normals. These steps are highlighted in algorithm 2.

In this work we propose instead to optimize:

$$\min_{\hat{M}} \|\hat{M} - M\|_F^2 \tag{8.4}$$
$$\text{s.t.} \quad \hat{M} \text{ is rank 3 and produced by an integrable surface.}$$

Eq. (8.3) optimizes over rank 3 matrices, which can represent sets of images produced by any set of

surface normals. In contrast, in 8.4 we optimize over only those rank 3 matrices that correspond to integrable surfaces.

Intuitively, a single optimization over all constraints will have a better global optimum than a sequence of optimizations in which constraints are used one at a time to increasingly narrow the solution (see illustration in Figure 8-1). A similar intuition has motivated the use of bundle adjustment [53] as the dominant approach to large scale structure-from-motion. Specifically in UPS the measurement matrix may contain many errors due to shadows and specular effects. Therefore, while in theory UPS can be solved with as few as three images, SVD can properly handle these modeling errors only when many images are supplied. Indeed, current methods [3, 40] typically use 10 or more images. With fewer images SVD results tend to provide noisy solutions. Our method incorporates integrability into this estimation, providing valuable additional constraints that can reduce this noise. Our experiments indicate that our method can produce reasonable reconstructions with as few as 4 images and good reconstruction with 6 images, significantly improving over state-of-the-art methods with these few images.

For our approach we optimize a cost function based on (8.4) over the surface, lighting, normals, and (restored) error-free observations. The cost ensures that normals and lighting are consistent with the measurements, which must have low rank. We use constraints that ensure integrability. This is somewhat tricky because rank constraints apply to the measurements while integrability constraints apply to the normals. We show that by constructing a rank 3 matrix that contains normals, measurements, and lighting, we can impose the rank and integrability constraints together. Specifically, we use a truncated nuclear norm approach [62] to enforce the rank constraint, while integrability is represented by linear equalities. This leads to a single non-convex problem that we solve using a series of Alternating Direction Method of Multipliers (ADMM) operations [18, 50].

Our formulation allows us to easily account for missing data in the measurement matrix. This

commonly occurs when pixels are dark due to shadows, or saturated due to specularities. In some of the prior approaches, this can be solved with a preprocessing step, which may lead to a pipeline with yet another optimization [170]. We handle missing data using matrix completion based on the rank constraint. We initialize our optimization using prior approaches, in much the same way that bundle adjustment is initialized using simpler, but non-optimal algorithms [53].

8.2 Our Approach

In this section we introduce our integrated formulation that enforces integrability of surface normals in solving the uncalibrated photometric stereo problem. We recall from (8.2) that the measurement matrix M can be factored into $M = LS$. To access the derivatives of $z(x, y)$ we write S as a product

$$S = N\Lambda, \tag{8.5}$$

where N is a $3 \times p$ matrix whose j'th column is $n_j = (-z_x, -z_y, 1)^T$ and $\Lambda = \mathrm{diag}(\lambda_i, \lambda_2, ..., \lambda_p)$ with $\lambda_j = -\rho_j / \|n_j\|$. We next define the matrix:

$$X = \begin{bmatrix} X^I & X^N \\ X^L & X^M \end{bmatrix} = \begin{bmatrix} I & N \\ L & M\Lambda^{-1} \end{bmatrix}, \tag{8.6}$$

where X is $(3+m)\times(3+p)$. The matrices X, Λ, and the depth values ($z(x, y)$) form the unknowns in our optimization. Note that, because $LN = M\Lambda^{-1}$, the following holds for any 3×3 non-degenerate matrix A

$$X = \begin{bmatrix} A^{-1} \\ LA^{-1} \end{bmatrix} \begin{bmatrix} A & AN \end{bmatrix}. \tag{8.7}$$

This shows that X is rank 3. The matrix A represents a linear ambiguity. However, forcing the normals in N to be integrable will reduce this ambiguity to the GBR.

To force integrability we denote by $\mathbf{z} = (z_1, ..., z_p)^T$ the vector of unknown depth values and require

$$X^N = \begin{bmatrix} D_x\mathbf{z}, & D_y\mathbf{z}, & -\mathbf{1} \end{bmatrix}^T, \tag{8.8}$$

where D_x, D_y denote respectively the x- and y-derivative operators and $\mathbf{1}$ denotes the vector of all 1's.

Additional constraints are obtained by noticing that, because $0 \leq \rho_j \leq 1$ and $\|n_j\| \geq 1$,

$$-1 \leq \lambda_j \leq 0 \tag{8.9}$$

and

$$X^I = I_{3\times 3}. \tag{8.10}$$

We are now ready to define our optimization function. Let W be a binary, $m \times p$ matrix so that $W_{ij} = 0$ if M_{ij} is missing and $W_{ij} = 1$ otherwise, and let

$$f_{data}(X, \Lambda) = \frac{1}{2}\|W \odot (M - X^M\Lambda)\|_F^2, \tag{8.11}$$

where $\odot$ denotes element-wise multiplication. Then (8.4) can be written as

$$\begin{aligned} \min_{X,\Lambda,\mathbf{z}} \quad & f_{data}(X, \Lambda) \\ \text{s.t.} \quad & \text{rank}(X) = 3, \ (8.8), \ (8.9), \text{and } (8.10). \end{aligned} \tag{8.12}$$

Handling the rank-3 constraint: Enforcing the non-convex constraint $\text{rank}(X) = 3$ can be chal-

lenging. In the context of matrix completion a recent paper [62] proposed using the Truncated Nuclear Norm (TNN) regularization term:

$$f_{tnn}(X) = \|X\|_* - \sum_{k=1}^{3} \sigma_k(X), \tag{8.13}$$

where $\|X\|_*$ denotes the nuclear norm of X and $\sigma_k(X)$ is the k-th largest singular value of X. Clearly, $f_{tnn}(X) = 0$ if and only if $\mathrm{rank}(X) \leq 3$. We use f_{tnn} as a regularizer and solve

$$\begin{aligned} \min_{X,\Lambda,\mathbf{z}} \quad & f_{data}(X,\Lambda) + c\, f_{tnn}(X) \\ \text{s.t.} \quad & (8.8),\ (8.9), \text{and } (8.10), \end{aligned} \tag{8.14}$$

where c is a preset scalar.

8.3 Optimization using ADMM

In this section we introduce a method for solving (8.14). This is a challenging problem because both f_{data} and f_{tnn} are non-convex. Specifically, f_{data} (8.11) is bilinear in X and Λ, while f_{tnn} (8.13) is a difference between two convex functions. Our solution is based on a nested iteration in which the outer loop uses majorization to decrease f_{tnn} whereas the inner loop uses the scaled ADMM with alternation to decrease f_{data}.

Outer loop: Following [62] at each iteration of the outer loop we replace $f_{tnn}(X)$ with a majorizer f_{maj}. Specifically, at iteration k let $X^{(k)} = U\Sigma V^T$ be the singular value decomposition of $X^{(k)}$, and let U_3 (and V_3) be the matrices containing the left (right) singular vectors corresponding to the three largest singular values of $X^{(k)}$. U_3 and V_3 are determined in the outer loop and are held

constant throughout the inner loop. We then define

$$f_{maj}^{(k)}(X) = \|X\|_* - \text{trace}(U_3^T X V_3). \tag{8.15}$$

It was shown in [62] that $f_{maj}^{(k)}(X) \geq f_{tnn}(X)$ for all X and that $f_{maj}^{(k)}(X^{(k)}) = f_{tnn}(X^{(k)})$, and so decreasing f_{maj} leads to decreasing f_{tnn}.

Inner loop: In the inner loop we seek to minimize

$$\begin{aligned} \min_{X,\Lambda,\mathbf{z}} \quad & f_{data}(X,\Lambda) + c f_{maj}^{(k)}(X) \\ \text{s.t.} \quad & (8.8),\ (8.9), \text{and } (8.10), \end{aligned} \tag{8.16}$$

We use scaled ADMM, a variant of the augmented Lagrangian method that splits the objective function and aims to solve the different subproblems separately. We maintain a second copy of X, which we denote by Y and form the augmented Lagrangian of (8.16) as follows

$$\begin{aligned} &\max_{\Gamma} \min_{X,\Lambda,\mathbf{z},Y} \quad \frac{1}{2}\|W \odot (M - X^M\Lambda)\|_F^2 + c\left(\|Y\|_* - \text{trace}(U_3^T Y V_3)\right) + \frac{\tau}{2}\|Y - X + \Gamma\|_F^2 \\ &s.t.\ X^I = I_{3\times 3},\ -1 \leq \lambda_j \leq 0\ \forall j,\ X^N = \begin{bmatrix} D_x\mathbf{z}, & D_y\mathbf{z}, & -\mathbf{1} \end{bmatrix}^T, \end{aligned} \tag{8.17}$$

where $\|Y - X + \Gamma\|_F^2$, denotes the Lagrangian penalty; τ is a constant, and Γ is a matrix of Lagrange multipliers the same size as X that is updated by the ADMM steps [18, 50]. We next describe the ADMM steps (applied iteratively).

Step 1: Solving for $(X, \Lambda, \mathbf{z})$.

In each iteration, k, we solve the following sub-problems:

1. Optimize w.r.t. X^I: $X^{I\,(k+1)} = I_{3\times 3}$.

2. Optimize w.r.t. X^L:

$$\begin{aligned} X^{L\,(k+1)} &= \underset{X^L}{\operatorname{argmin}} \|Y^{L\,(k)} - X^L + \Gamma^{L\,(k)}\|_F^2 \\ &= Y^{L\,(k)} + \Gamma^{L\,(k)}. \end{aligned} \tag{8.18}$$

3. Optimize w.r.t. X^N and $\mathbf{z}$:

$$\begin{aligned} (X^{N\,(k+1)}, \mathbf{z}^{(k+1)}) &= \underset{X^N, \mathbf{z}}{\operatorname{argmin}} \|Y^{N\,(k)} - X^N + \Gamma^{N\,(k)}\|_F^2 \\ \text{s.t. } X^N &= \begin{bmatrix} D_x\mathbf{z}, & D_y\mathbf{z}, & -\mathbf{1} \end{bmatrix}^T. \end{aligned} \tag{8.19}$$

The problem is solved by setting the third row of $X^{N\,(k+1)}$ to $-\mathbf{1}$ and by substituting $D_x\mathbf{z}$ and $D_y\mathbf{z}$ for the first two rows of X^N in the objective, obtaining linear least squares equations in $\mathbf{z}$ that can be solved directly.

4. Optimize w.r.t. X^M and Λ:

$$\begin{aligned} (X^{M\,(k+1)}, \Lambda^{(k+1)}) =& \underset{X^M, \Lambda}{\operatorname{argmin}} \frac{1}{2}\|W \odot (M - X^M\Lambda)\|_F^2 + \frac{\tau}{2}\|Y^{M\,(k)} - X^M + \Gamma^{M\,(k)}\|_F^2 \\ \text{s.t.} \quad & -1 \le \lambda_j \le 0 \ \forall j. \end{aligned}$$

We will separate this into the known and unknown pixels based on W. For an **unknown pixel** j in frame i ($W_{ij} = 0$) the first term vanishes and the minimization only determines the respective entry of X^M so that:

$$X_{ij}^{M\,(k+1)} = Y_{ij}^{M\,(k)} + \Gamma_{ij}^{M\,(k)}. \tag{8.20}$$

For the **known pixels**, since Λ is diagonal we can write these equations separately for each

column j (corresponding to the j-th pixel):

$$(X_j^{M\,(k+1)}, \lambda_j^{(k+1)}) = \underset{X_j^M, \lambda_j}{\operatorname{argmin}} \frac{1}{2}\|(W_j \odot (M_j - \lambda_j X_j^M)\|_2^2 + \frac{\tau}{2}\|Y_j^{M\,(k)} - X_j^M + \Gamma_j^{M\,(k)}\|_2^2$$

$$\text{s.t.} \quad -1 \le \lambda_j \le 0. \tag{8.21}$$

The problem (8.21) is non-convex. We will solve it with *alternate optimization*. X^M and Λ are updated by the following steps until convergence.

$\mathbf{X^M}$: Let $\tilde{M}_j = W_j \odot M_j$, $\tilde{X}_j = W_j \odot X_j^M$ and $\tilde{A}_j^{M\,(k)} = W_j \odot (Y_j^{M\,(k)} + \Gamma_j^{M\,(k)})$. Then,

$$\begin{aligned} \tilde{X}_j &= \underset{\tilde{X}_j}{\operatorname{argmin}} \frac{1}{2}\|\tilde{M}_j - \lambda_j \tilde{X}_j\|_2^2 + \frac{\tau}{2}\|\tilde{A}_j^{M\,(k)} - \tilde{X}_j\|_2^2 \\ &= \frac{\lambda_j \tilde{M}_j + \tau \tilde{A}_j^{M\,(k)}}{\lambda_j^2 + \tau}. \end{aligned} \tag{8.22}$$

$\mathbf{\Lambda}$:

$$\begin{aligned} \lambda_j &= \underset{\lambda_j}{\operatorname{argmin}} \frac{1}{2}\|\tilde{M}_j - \lambda_j \tilde{X}_j\|_2^2 \ \text{ s.t. } -1 \le \lambda_j \le 0, \\ &= \min(0, \max(-1, \tilde{X}_j^T \tilde{M}_j / \|\tilde{X}_j\|_2^2)). \end{aligned} \tag{8.23}$$

Step 2: Solving for Y. Solving for Y requires a solution to

$$Y^{(k+1)} = \underset{Y}{\operatorname{argmin}}\, c\left(\|Y\|_* - \operatorname{trace}(U_3^T Y V_3)\right) + \frac{\tau}{2}\|Y - X^{(k+1)} + \Gamma^{(k)}\|_F^2. \tag{8.24}$$

Below we show that this problem can be solved in closed form by applying the shrinkage operator, obtaining

$$Y^{(k+1)} = D_{c/\tau}(X^{(k+1)} - \Gamma^{(k)} + \frac{c}{\tau} U_3 V_3^T), \tag{8.25}$$

where the shrinkage operator $D_t(.)$ is defined as follows. For a scalar s we define $D_t(s) = \text{sign}(s) \times \max(|s| - t, 0)$. For a diagonal matrix $S = \text{diag}(s_1, s_2, ...)$ with non-negative entries we define $D_t(S) = \text{diag}(D_t(s_1), D_t(s_2), ...)$. Finally, for a general matrix Υ, let $\Upsilon = \tilde{U}S\tilde{V}^T$ be its singular value decomposition, then $D_t(\Upsilon) = \tilde{U}D_t(S)\tilde{V}^T$.

To derive (8.25), we rewrite (8.24) as:

$$Y^{(k+1)} = \underset{Y}{\text{argmin}} \|Y\|_* + \frac{\tau}{2c}\|Y - X^{(k+1)} + \Gamma^{(k)} - \frac{c}{\tau}U_3V_3^T\|_F^2 - T, \tag{8.26}$$

where $T = \text{trace}(V_3U_3^T(X^{(k+1)} - \Gamma^{(k)})) + \frac{c}{2\tau}\|U_3V_3^T\|_F^2$ is independent of Y. Equation (8.26) is of the general form $\min_Y \|Y\|_* + \frac{1}{2t}\|Y - C\|_F^2$, for which the solution is $D_t(C)$, as is shown in [24], implying (8.25).

Step 3: Update of Γ. The matrix Γ contains Lagrange multipliers that are used in the saddle-point formulation (8.17) to enforce the equality constraint $X = Y$. The following update is a gradient ascent step that acts to maximize the augmented Lagrangian (8.17) for Γ. For details, see [18, 50].

$$\Gamma^{(k+1)} = \Gamma^{(k)} + (Y^{(k+1)} - X^{(k+1)}). \tag{8.27}$$

The entire optimization process is listed in Algorithm 3.

8.4 Experimental Results

In this section we evaluate and compare the performance of our algorithm with two versions of the baseline algorithm, in both real world and synthetic examples. We compare the following methods:

Baseline: Algorithm 2 described in Section 8.1. This method is used in [3, 40, 137, 30, 152, 38].

RPCA: Images are preprocessed using Robust PCA [170], parameters are chosen as suggested

Algorithm 3 TNN formulation solved with ADMM

Input: M, W.
Output: X, $\mathbf{z}$.
Initialization: Initialize X^L and X^N by running Baseline algorithm (without resolving GBR). Initialize $X^M = -M$, $\Lambda = -I$, and $c = 1$. Set $X^{(0)} = X$, $Y = X$, $\Gamma = 0$, and $\tau = 1$.
$k = 0$.
while not converged **do**
  Perform SVD over $X^{(k)}$ to obtain U_3 and V_3.
  Run ADMM:
  while not converged **do**
    Update of X**,** $\mathbf{z}$ **and** Λ.
    Update $X^{I(k+1)} = I_{3\times 3}$.
    Update $X^{L(k+1)}$ using (8.18).
    Update $X^{N(k+1)}$ and $\mathbf{z}$ using (8.19).
    while not converged **do**
      for each pixel j **do**
        Update $X_j^{M(k+1)}$ using (8.22) and $\lambda_j^{(k+1)}$ using (8.23).
      end for
      for each pixel j in each image i **do**
        if $W_{ij} = 0$ i.e. pixel j is not known **then**
          Update $X_{ij}^{M(k+1)}$ using (8.20).
        end if
      end for
    end while
    Update Y using (8.25).
    Update of Γ using (8.27).
    $k = k + 1$.
  end while
end while

by [40]. Then we apply the baseline algorithm to the obtained matrix. This method is used in [40]. RPCA solves a sparse low rank optimization to detect shadows and other non-Lambertian effects. The method uses L_1 regularization to identify outlier pixels, even when they do not result in intensities near 0 or 1.

Our(NC): Our proposed formulation as described in Section 8.3 using $W = 1$, i.e., no completion. This allows comparison to Baseline, which also does not perform matrix completion.

Our(MC): Our proposed formulation as in Section 8.3 with $w_{ij} \in \{0, 1\}$, allowing for matrix completion. In both versions of our algorithm we use $c = 1$ and use RPCA to initialize optimization. We

identify missing pixels as those with normalized intensity outside the range of $(0.02, 0.98)$.

All the tested methods solve for the surface only up to a GBR ambiguity. To compare the results with ground truth, we find the GBR that optimizes the fit to ground truth, and measure the residual error.

In the presence of a large number of images with noise and non-Lambertian effects, we expect the sequential pipeline of Baseline and RPCA, involving SVD, to produce accurate solutions, because the problem solved by SVD is heavily overconstrained. In the presence of fewer images, our integrated method will be able to produce a more accurate decomposition by using both rank and integrability constraints to find the right linear subspace. Thus we expect our integrated approach to improve over the Baseline and RPCA as we reduce the number of images. In the following sub-section we will show results with synthetic and real world data that supports our claim.

8.4.1 Experiments on Synthetic Data

We use five real objects ("cat", "owl", "rock", "horse", "buddha") to produce synthetic images, their shape is obtained by applying calibrated photometric stereo to a publicly available dataset [73]. We use the normals and albedos from these objects to generate images. Each image is generated by a randomly selected light source which lies at 30 degrees of the viewing direction on average. All images are of size 512 × 340 with objects occupying 29-72K pixels. A segmentation mask is also supplied. To show the variation of performance with the number of images N_I, we use sets of 4, 6, 8, 10, 15, 20, 25 and 30 images respectively. We add Gaussian noise with standard deviation ranging from 1% to 7% (in steps of 2%) of the maximum intensity. For each choice of noise, we run 5 different trials with random noise and lighting to generate the synthetic images. Thus we have 5 objects, 4 levels of noise and 5 random simulations, making a total of 100 experiments for each of the 8 different sets of 4, 6, 8, 10, 15, 20, 25 and 30 images. As a measure of performance,

we calculate the error in the reconstructed depth map. Let the ground truth surface be Z_T and the reconstructed surface be Z_{rec}. We measure error in depth as $Z_{err} = 100 \times \frac{\|Z_T - Z_{rec}\|}{\|Z_T\|}$. To compare two algorithms (say, algorithm A vs. algorithm B), we define the following two terms :

Relative Improvement (in $\%$) : Denote e_k^a and e_k^b as the depth error for each trial k by using algorithm A and B respectively. The Relative Improvement of algorithm B over A is the average of $\frac{(e_k^b - e_k^a)}{e_k^b}$ over all trials K for each choice of N_I expressed in percentage.

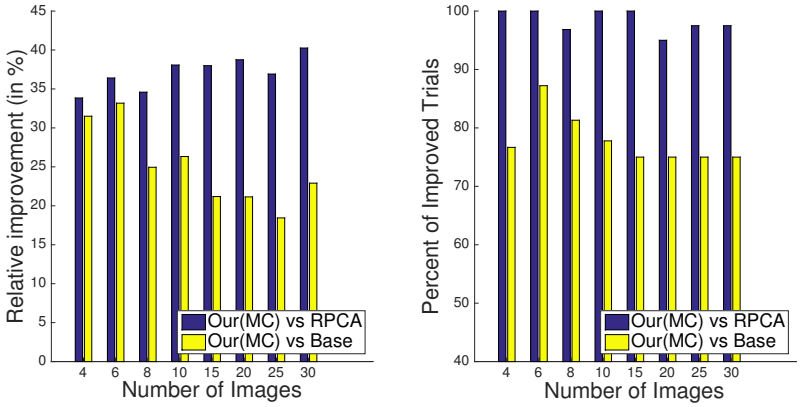
Percent of Improved Trials : This denotes the number of trials in which algorithm B improves over A. In terms of notation introduced previously, this is $\frac{1}{K}\sum_{k=1}^{K}\mathbb{I}(e_k^{\mathrm{a}} < e_k^{\mathrm{b}})$, where $\mathbb{I}(.)$ is in indicator variable and K is the total number of trials for each choice of N_I. The measure is expressed in percentage.

In Figure 8-2a we compare performance of Our(MC) with Baseline and RPCA, on synthetic data in the presence of Gaussian noise. We initialize our methods with RPCA. We observe that as the number of images decreases, our method improves compared to Baseline and RPCA. With simple Gaussian noise RPCA doesn't produce additional advantages as there are no outliers.

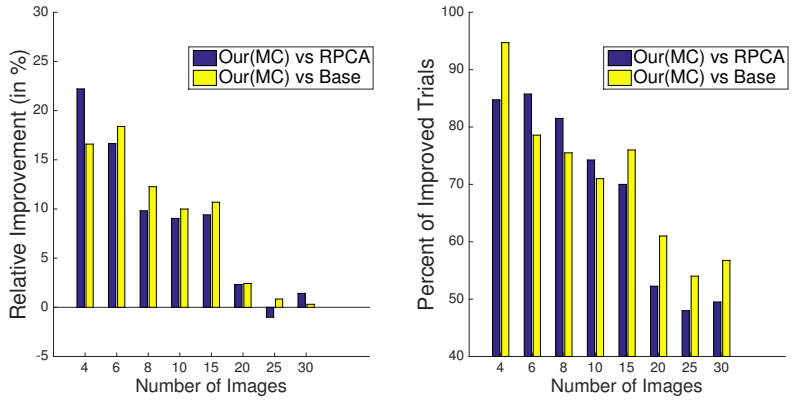
In Figure 8-2b we compare the performance of our methods on synthetic data with Gaussian noise and with specularities generated by the Phong reflectance model [151, 122]. Mathematically each image M_i can be represented as :

$$M_i = L_i S + k_s (VR)^{\alpha}, \tag{8.28}$$

where V is the viewing direction and R denotes the directions of perfect reflection for incoming light L_i for each pixel j. Larger α produces sharper specularities, while larger k_s causes more light to be reflected as specularity. We use $k_s = 0.2$ and $\alpha = 10$. We observe that the advantage of Our(MC) degrades as the number of images increases, as expected. This experiment shows that



(a) Gaussian noise



(b) Gaussian noise with Phong model

Figure 8-2: Performance comparison of Our(MC) algorithm to RPCA (in blue) and Baseline (yellow) for different numbers of input images with gaussian noise under either a pure lambertian model (top) or the Phong model (bottom). The left bar plot shows the amount of relative improvement achieved with our algorithm, and the right plot shows the percent of trials in which our algorithm out performed each one of the competing algorithms.

even though our method is designed specifically for Lambertian objects it can tolerate a certain amount of model irregularities such as specularity. With 4 images our method beats RPCA in 85% of the all trials with a relative improvement of 22.12%.

In Figure 8-3 we compare Our(MC) with Baseline and RPCA with variation of noise for different subsets of images (4, 6, 10 and 15). We can conclude that our method is robust to noise and its advantages do not degrade with an increase in noise.

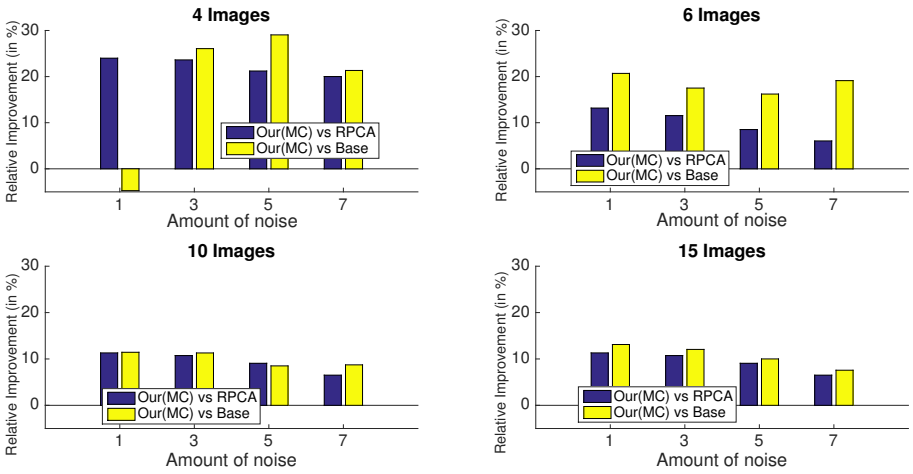


Figure 8-3: Performance comparison of Our(MC) with RPCA and Baseline with varying noise created using the Phong model.

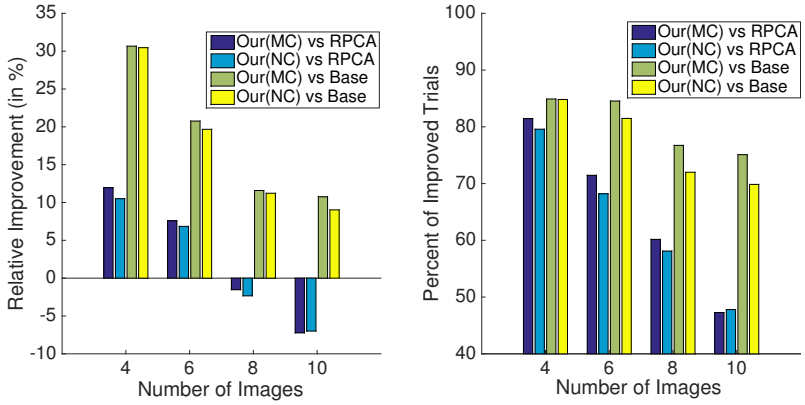


Figure 8-4: Performance comparison of Our (MC) and Our (NC) algorithms to RPCA and Baseline with real images.

8.4.2 Experiments on Real World Data

To test our approach on real data, we used the two publicly available data sets [73] and [171] consisting of 5 and 7 objects respectively. The datasets provide calibrated lighting, which we use to perform calibrated photometric stereo. The obtained depth map, albedo, and surface normals are considered as ground-truth for photometric stereo with unknown lighting similar to [3]. To show the variation of performance with the number of images, we select subset of 4, 6, 8 and 10 images for each object. We perform 10 random selections of subset of images for each of the 12 objects.

Thus we have 120 experiments for every subset of images.

In Figure 8-4 we compare the performance of our methods, Our(MC) and Our(NC), with Baseline and RPCA with variation in the number of images. We see that for fewer images our methods outperform Baseline and RPCA by a significant amount and are comparable to RPCA for more images. For 4 images Our(MC) outperforms Baseline in 84.9% cases with a relative improvement of 30.6% and outperforms RPCA in 81.4% cases with a relative improvement of 12%. However for 10 images we beat Baseline in 75% cases with a relative improvement of 10.7% and beat RPCA in only 47.3% cases with a relative improvement of -7.2%.

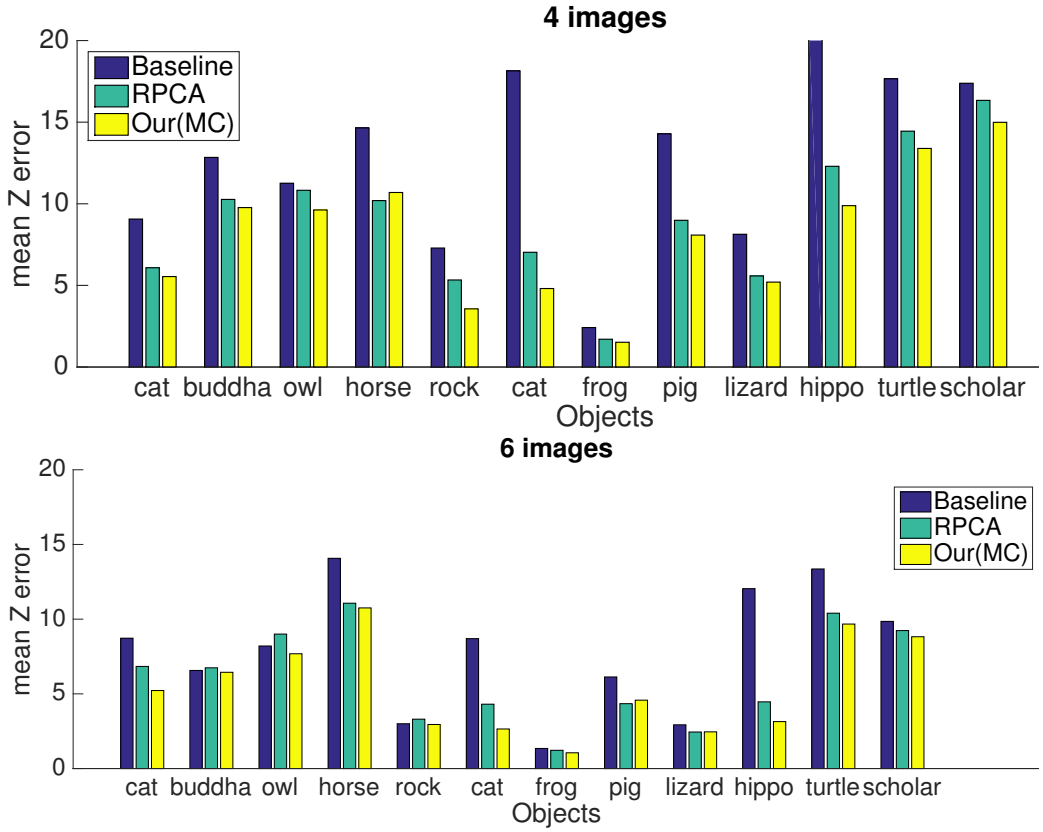


Figure 8-5: Average surface reconstruction error with 4 (top) and 6 (bottom) real images of 12 objects over 10 random trials using Our(MC), RPCA and Baseline.

Figure 8-5 shows the average reconstruction error obtained by Our(MC), RPCA and Baseline on 12 real-world objects over 10 random simulations. We observe that Our(MC) outperforms RPCA

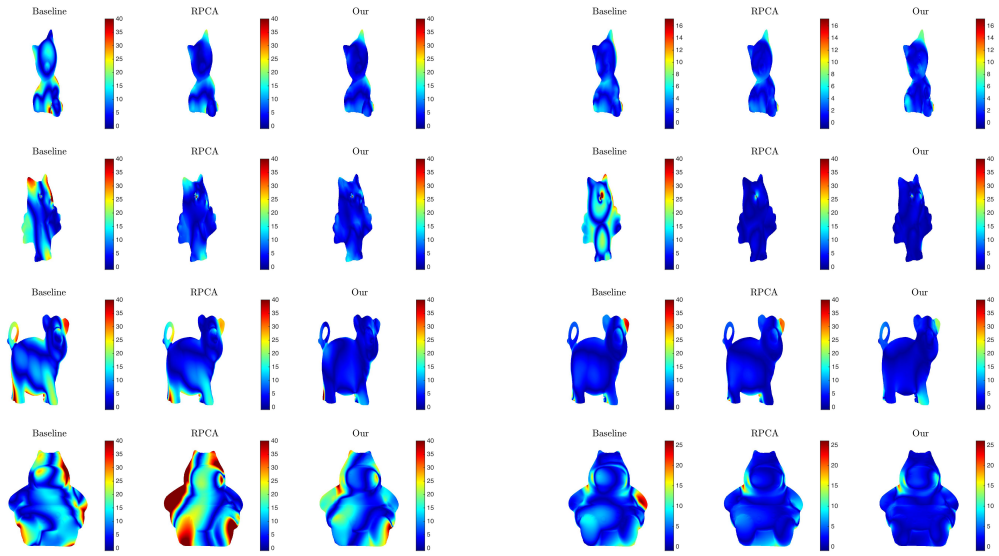


Figure 8-6: Reconstruction error $|Z_T - Z_{rec}|$ for Baseline, RPCA and Our(MC) on “Cat", “Owl", “Pig" and “Hippo” shown in each row. The left column shows results for 4 images, the right shows results for 10.

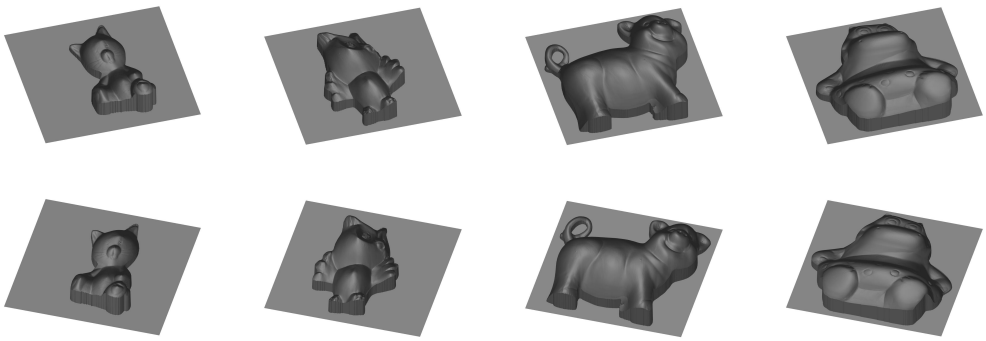


Figure 8-7: Two views of surfaces reconstructed with Our(MC) algorithm for 4 images. Each column shows two images of surfaces reconstructed on “Cat", “Owl", “Pig" and “Hippo” respectively.

on 11 out of 12 objects for 4 images and 10 out of 12 objects for 6 images (and is comparable in 1). With 10 images the average reconstruction error using Our(MC) over all objects and all trials is 4.6%. This increases to 8.1% with four images, and is only 5.4% with six images. This shows that we have reasonable reconstruction with 4 images and good reconstruction with as few as 6 images.

In Figure 8-6 we compare the error in surface reconstruction between Baseline, RPCA, and Our(MC) on some of our real world examples. Figure 8-7 shows two views of surfaces reconstructed using the Our(MC) algorithm using 4 images, showing reasonable surface reconstruction.

These results suggest that our joint approach to enforcing rank and integrability constraints can significantly improve the performance of photometric stereo in the presence of a few images.

In general, we see that incorporating matrix completion into our formulation results in a slight improvement, with Our(MC) somewhat outperforming Our(NC). This indicates that the improvement of our method compared to RPCA or Baseline is mostly due to the joint optimization formulation and not due to matrix completion. We further note that RPCA seems to significantly improve over Baseline. RPCA is able to identify outliers and use that extra information for better recovery. This also suggests that the robust error function used by RPCA is important. However our integrated approach, which does not have a robust cost function like RPCA, still outperforms RPCA for 4 and 6 images and is almost equal for 8 or 10 images. This shows that an integrated approach is very useful for a small number of images and provides similar gain compared to RPCA for more images. It would be an interesting topic of future work to amend the cost function of Our(MC) to include RPCA's robust handling of error, to see if this further improves its performance.

For an image of size 512×340 with an object occupying an area of 30K pixels, our algorithm takes 20 minutes on a 2.7 GHz Intel Core i5 machine.

Chapter 9

Concluding Remarks

In this thesis we aim to solve the problem of Inverse Rendering from limited observations. The goal of Inverse Rendering is to obtain the intrinsic components, i.e. geometry, reflectance and illumination, of an image. In this thesis we study Inverse Rendering as different sub-problems like Structure from Motion, Photometric Stereo and Shape from Shading. We propose two different approaches to study Inverse Rendering. In presence of a fewer images, we aim to enforce low-rank constraints in an optimization framework to ensure accurate recovery. Another direction is to learn data-driven priors over intrinsic components for Inverse Rendering from a single image.

We summarize the major contributions of this thesis as follows:

- We propose a new multi-view rank constraint that connects all cameras capturing a scene. We formulate a constrained low-rank optimization problem that aims at enforcing this constraint to ensure accurate recovery of camera matrices in a Structure from Motion framework. This additional constraint is particularly helpful in presence of a small number of images.

- We propose a joint approach for solving Uncalibrated Photometric Stereo. We aim to simultaneously obtain a low-rank decomposition of a set of images and remove ambiguity by formulating a constrained low-rank optimization problem.

- In pursuit of Inverse Rendering from a single image, we propose a self-supervised learning paradigm that allows us to learn from unlabeled real data while disambiguating the intrinsic components.
- To handle the complex components of the appearance of scenes, we propose Residual Appearance Renderer (RAR) which facilitates self-supervised learning from real world scene images.
- We propose a novel inverse rendering decomposition architecture, SfSNet, which mimics a physical rendering model.
- We introduce a large-scale photo-realistic dataset of indoor scenes with ground-truth labels for geometry, reflectance, illumination and material classes.
- We show that jointly learning all components of an image is significantly better than learning only one of the components of an image as it helps in better generalization.

In this thesis we explore a few directions and provide solutions that advances the goal of achieving Inverse Rendering from unconstrained images. There are some clear next steps that can improve the quality of Inverse Rendering for in-the-wild images. As a concluding remarks, we briefly explore few of these directions.

Combining Priors and Constraints for Multi-view Inverse Rendering. In our thesis we consider two possible setups for Inverse Rendering. In one setup we consider only a single image during inference and provide a learning based solution to this problem. In another setup we consider a few images during inference time and provide an optimization driven framework that can enforce certain constraints. Clearly the next step forward is to combine learning data-driven priors with enforcing constraints to achieve accurate Inverse Rendering from multiple images. Often, it is very convenient for a user to take multiple images or a video of an object or scene of interest. As multi-

view data provides more meaningful information, the goal will be to leverage them for accurate reconstruction.

Unified Geometric and Photometric Inverse Rendering. So far in this thesis, we have separately considered the geometric and photometric aspects of Inverse Rendering. For example, in SfM we consider only geometric aspects in terms of 3D shape and camera parameters. And in case of single image based Inverse Rendering we mostly focus on the photometric aspects. Having multiple images or a video as input will allow us to combine both of these aspects and create a strong self-supervision loss for learning from unlabeled real images

Improved Representations of Reflectance and Illumination. So far we have considered the most simple representations of reflectance and illumination and mostly focused on developing an approach to successfully recovering them. Given that we have a good progress in this direction, it is now important to consider more expressive representations of reflectance and lighting. For example, instead of predicting only albedo we can try to infer more detailed parametric models for reflectance or even learn an improved data-driven BRDF model. Similarly, we can infer local models of illumination as light fields or per-pixel lighting and utilize improved rendering models like Pre-computed Radiance Transfer (PRT).

A Wide Range of Applications. Finally progress in understanding Inverse Rendering will lead to creating and solving many interesting applications. For example, Inverse Rendering will improve different applications in Virtual Shopping, where a user can virtually insert the furniture of their choice in the room and change the material of the furniture to suit their purpose. Inverse Rendering will also be useful in editing images and generating interesting content by inserting objects in the image. Finally, it will be also useful for several AR/VR applications.

Chapter 10

Appendix

10.1 Inverser Rendering of a Face

10.1.1 SfSNet Architecture

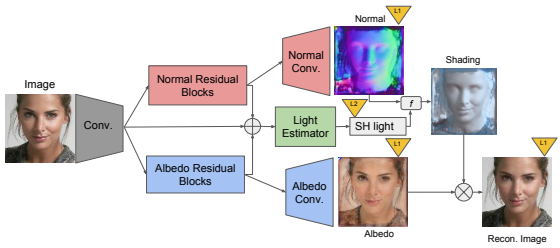


Figure 10-1: **SfSNet Architecture**.

The schematic diagram of our SfSNet is again shown in Figure 10-1 for reference. Our input, normal and albedo is of size 128×128. Below we provide the details of each of the blocks of SfSNet.

'Conv.': C64(k7) - C128(k3) - C*128(k3)

'CN(kS)' denotes convolution layers with N $S \times S$ filters with stride 1, followed by Batch Normalization and ReLU. 'C*N(kS)' denotes only convolution layers with N $S \times S$ filters with stride 2, without batch Normalization. The output of 'Conv' layer produces a blob of spatial resolution

$128 \times 64 \times 64$.

'Normal Residual Blocks': 5 ResBLK - BN - ReLU

This consists of 5 Residual Blocks, 'BesBLK's, all of which operate at a spatial resolution of $128 \times 64 \times 64$, followed by Batch Normalization (BN) and ReLU. Each 'ResBLK' consists of BN - ReLU - C128 - BN - ReLU - C128.

'Albedo Residual Blocks': Same as 'Normal Residual Blocks' (weights are not shared).

'Normal Conv'.: BU - CD128(k1) - C64(k3) - C*3(k1)

'BU' refers to Bilinear up-sampling that converts $128 \times 64 \times 64$ to $128 \times 128 \times 128$.'CN(kS)' represents convolution layers with N $S \times S$ filters with stride 1, followed by Batch Normalization and ReLU. 'C*N(kS)' represents only convolution layer with N $S \times S$ filters with stride 1. The network produces a normal map as output.

'Albedo Conv.': Same as 'Normal Conv.' (weights are not shared).

'Light Estimator': It first concatenates the responses of 'Conv', 'Normal Residual Blocks' and 'Albedo Residual Blocks' to produce a blob of spatial resolution $384 \times 64 \times 64$. This is further processed by 128 1×1 convolutions, Batch Normalization, ReLU, followed by Average Pooling over 64×64 spatial resolution to produce 128 dimensional features. This 128 dimensional feature is passed through a fully connected layer to produce 27 dimensional spherical harmonics coefficients of lighting. Our model and code is available for research purposes at https://senguptaumd.github.io/SfSNet/.

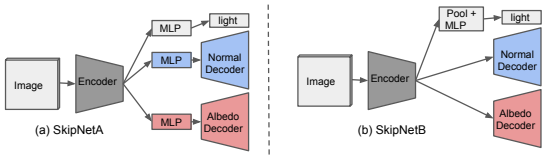


Figure 10-2: **SkipNet and SkipNet+ Network Architectures**.

10.1.2 SkipNet Architecture

The schematic diagram of SkipNet is shown in Figure 10-2(a). SkipNet is based on the network used in [141] with more capacity and skip connections. Similar to SfSNet the input is 128×128; 'Normal Decoder' and 'Albedo Decoder' produces normal and albedo maps. Normal, albedo and 'light' is also used to produce shading and the reconstructed image similar to Figure 10-1. Since that part of the architecture does not contain any trainable parameters we omit them in the figure for clarity. Note that the skip connections between encoder and decoder exist, which is also not shown in the figure. Details of SkipNet are provided below:

Encoder: C*64(k4) - C128(k4) - C256(k4) - C256(k4) - C256(k4) - fc256

'CN(kS)' represents convolution layers with N $S \times S$ filters with stride 2, followed by Batch Normalization and ReLU. 'C*N(kS)' is'CN(ks)' without Batch Normalization. All ReLUs are leaky with slope 0.2. 'fc256' is a fully connected layer that produces a 256 dimensional feature.

MLP: Contains a fully connected layer to take the response of Encoder and separate it into 256 dimensional features for 'Normal Decoder', 'Albedo Decoder' and 'light'. For 'Normal Decoder' and 'Albedo Decoder' a 256 dimensional feature is further up-sampled to form a blob of shape $256 \times 4 \times 4$. For 'light' the 256 dimensional feature is passed through a fully connected network to produce 27 dimensional spherical harmonics coefficients.

Decoder (Normal and Albedo): CD256(k4) - CD256(k4) - CD256(k4) - CD128(k4) - CD64(k4) - C*3(k1) Both 'Normal Decoder' and 'Albedo Decoder' consists of the same architecture without

weight sharing. 'CDN(kS)' represents a de-convolution layer with N $S \times S$ filters operated with stride 2, followed by Batch Normalization and ReLU. 'C*3(k1)' consists of 3 1×1 convolution filters with stride 1 to produce Normal or Albedo. Skip connections are present between encoders and decoders similar to [65, 133].

10.1.3 SkipNet+

SkipNet+ is very similar to SkipNet, but with larger capacity and without a fully connected bottleneck 'MLP' as shown in Figure 10-2(b). The Details of the network are shown below.

Encoder: Co64(k3) - Co64(k1) - C64(k3) - Co64(k1) - C128(k3) - Co128(k1) - C256(k3) - Co256(k1) - C256(k3) - Co256(k1) - C256(k3)

'CN(kS)' represents a convolution layer with N $S \times S$ filters with stride 2, followed by Batch Normalization and ReLU. 'CoN(kS)' is similar to 'CN(kS)' but with stride 1. All ReLUs are leaky with slope 0.3. The output of the Encoder is a feature of spatial resolution $256 \times 4 \times 4$.

Decoder (Normal and Albedo): C256(k1) - CD256(k4) - CD256(k4) - CD256(k4) - CD128(k4) - CD64(k4) - C*3(k1)

'CDN(kS)' represents a de-convolution layer with N $S \times S$ filters with stride 2, followed by Batch Normalization and ReLU. 'CN(kS)' represents a convolution layer with N $S \times S$ filters with stride 1, followed by Batch Normalization and ReLU. 'C*3(k1)' consists of 3 1×1 convolution filters to produce Normal or Albedo. Skip-connections exists between 'CN(k3)' layers of encoder and 'CDN(k4)' layers of decoder.

light: We perform Average pooling over 4×4 spatial resolution of the encoder output to produce a 256 dimensional feature. This feature is then passed through a fully connected layer to produce 27 dimensional spherical harmonics lighting.

10.1.4 Spherical Harmonics

In this section, we define the image generation process under lambertian reflectance following equation (3.1). Let the normal be $n(p) = [x, y, z]^T$ at pixel p. Then the 9 dimensional spherical harmonics basis $Y(p)$ at pixel p is expressed as:

$$Y = [Y_{00}, Y_{10}, Y_{11}^e Y_{11}^0, Y_{20}, Y_{21}^e, Y_{21}^o, Y_{22}^e, Y_{22}^o]^T, \tag{10.1}$$

where

$$\begin{aligned}
Y_{00} &= \tfrac{1}{\sqrt{4\pi}} & Y_{10} &= \sqrt{\frac{3}{4\pi}}z \\
Y_{11}^e &= \sqrt{\tfrac{3}{4\pi}}x & Y_{11}^o &= \sqrt{\frac{3}{4\pi}}y \\
Y_{20} &= \tfrac{1}{2}\sqrt{\tfrac{5}{4\pi}}(3z^2 - 1) & Y_{21}^e &= 3\sqrt{\frac{5}{12\pi}}xz \\
Y_{21}^o &= 3\sqrt{\tfrac{5}{12\pi}}yz & Y_{22}^e &= \frac{3}{2}\sqrt{\frac{5}{12\pi}}(x^2 - y^2) \\
Y_{22}^0 &= 3\sqrt{\tfrac{5}{12\pi}}xy
\end{aligned} \tag{10.2}$$

Then the intensity at pixel p is defined as:

$$I(p) = f_{render}(A(p), N(p), L) = A(p)(Y(p)^T L), \tag{10.3}$$

where $A(p)$ is the albedo at pixel p, and L is the lighting parameter denoting coefficients of spherical harmonics basis. Note that, the above equations are only for one of the RGB channels and can be repeated independently for 3 channels.

Next we define the reconstruction loss. Let $I(p)$ be the original image intensity and $\tilde{N}(p)$, $\tilde{A}(p)$ be the inferred normal and albedo by SfSNet at pixel p. Let $\tilde{L}$ be the 27 dimensional spherical

harmonic coefficients also inferred by SfSNet. The reconstruction loss is defined as:

$$E_{recon} = \sum_{p} |I(p) - f_{render}(\tilde{A}(p), \tilde{N}(p), \tilde{L})|. \tag{10.4}$$

10.1.5 More Qualitative Comparisons

SfSNet on CelebA: In Figures 10-3 and 10-4 we present inverse rendering results on CelebA images with our SfSNet. To visualize the quality of the reconstructed normals, we use directional lights with uniform albedo to produce 'Relit' images.

SfSNet vs Pix2Vertex: In Figure 10-5 we compare SfSNet to Pix2Vertex [133]. These images contain non-ambient illuminations and expressions, where surface normal recovery is much more robust for SfSNet than for Pix2Vertex. Figures 10-6, 10-7 and 10-8 also compares performance of SfSNet and Pix2Vertex on the images showcased by Sela *et. al* in [133]. Since these images mostly contain ambient illumination, SfSNet performs comparable to Pix2Vertex.

SfSNet vs MoFA: We also provide more comparison results with MoFA [156] on the images provided by the authors in Figures 10-10, 10-11 and 10-12. MoFA aims to fit a 3DMM which is limited in its capability to represent real world shapes and reflectance, but can produce a full 3D mesh. Thus SfSNet reconstructs more detailed shape and reflectance than MoFA.

SfSNet vs Neural Face: Similarly comparison with 'Neural Face' [141] in Figure 10-13 on the images showcased by the authors, show that SfSNet obtains more realistic reconstruction than 'Neural Face'.

Figure 10-3: Results of SfSNet on CelebA. 'Relit' images are generated by directional lighting and uniform albedo to highlight the quality of the reconstructed normals. (Best viewed in color)

Figure 10-4: Results of SfSNet on CelebA. 'Relit' images are generated by directional lighting and uniform albedo to highlight the quality of the reconstructed normals. (Best viewed in color)

Figure 10-5: **SfSNet vs Pix2Vertex** [133] on images selected by us with non-ambient illumination and expression. 'Relit' images are generated by directional lighting and uniform albedo selected to highlight the quality of the reconstructed normals. (Best viewed in color)

Figure 10-6: **SfSNet vs Pix2Vertex** [133] on the images showcased by Sela *et. al.* in [133]. 'Relit' images are generated by directional lighting and uniform albedo selected to highlight the quality of the reconstructed normals. (Best viewed in color)

Figure 10-7: **SfSNet vs Pix2Vertex** [133] on the images showcased by Sela *et. al.* in [133]. 'Relit' images are generated by directional lighting and uniform albedo selected to highlight the quality of the reconstructed normals. (Best viewed in color)

Figure 10-8: **SfSNet vs Pix2Vertex** [133] on the images showcased by Sela *et. al.* in [133]. 'Relit' images are generated by directional lighting and uniform albedo selected to highlight the quality of the reconstructed normals. (Best viewed in color)

Figure 10-9: **Light transfer.** Our SfSNet allows us to transfer lighting of the 'Source' image to the 'Target' image to produce 'Transfer' image. 'S' refers to shading. (Best viewed in color)

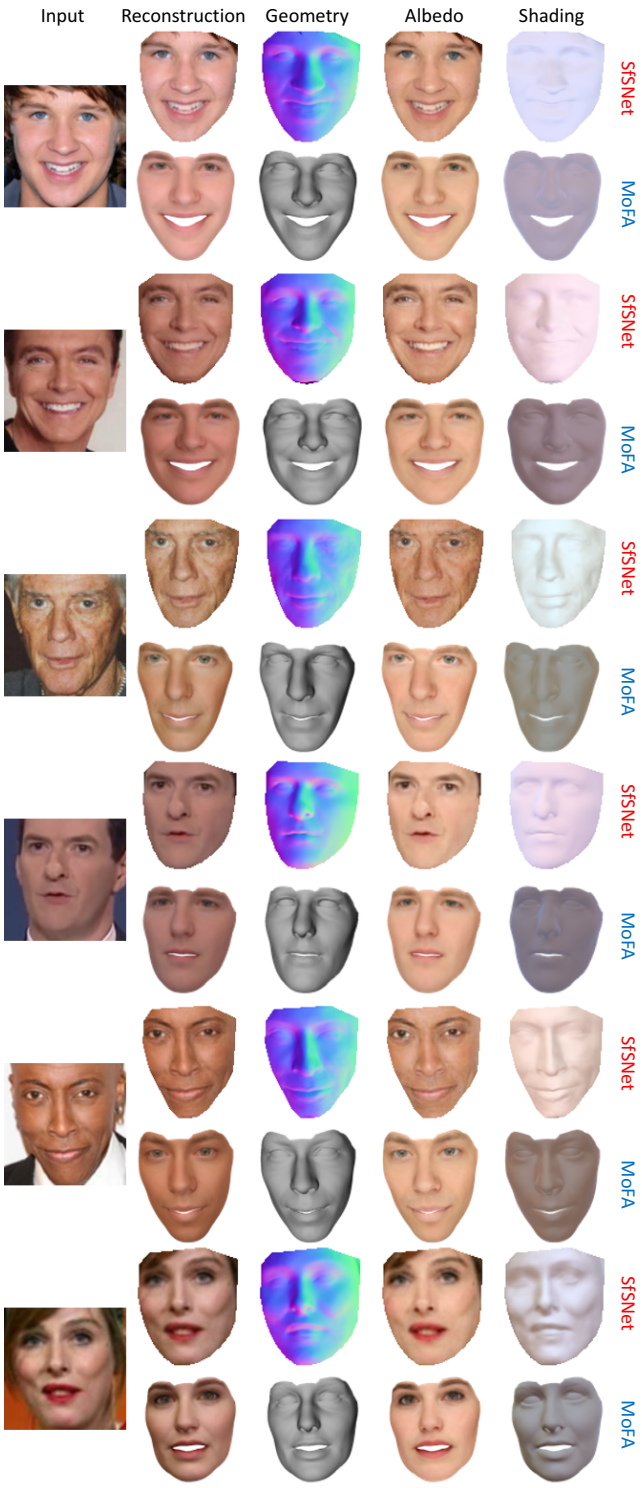


Figure 10-10: **Inverse Rendering. SfSNet vs 'MoFA'** [156] on the data provided by the authors. (Best viewed in color)

Figure 10-11: **Inverse Rendering. SfSNet vs 'MoFA'** [156] on the data provided by the authors. (Best viewed in color)

Figure 10-12: **Inverse Rendering. SfSNet vs 'MoFA'** [156] on the data provided by the authors. (Best viewed in color)

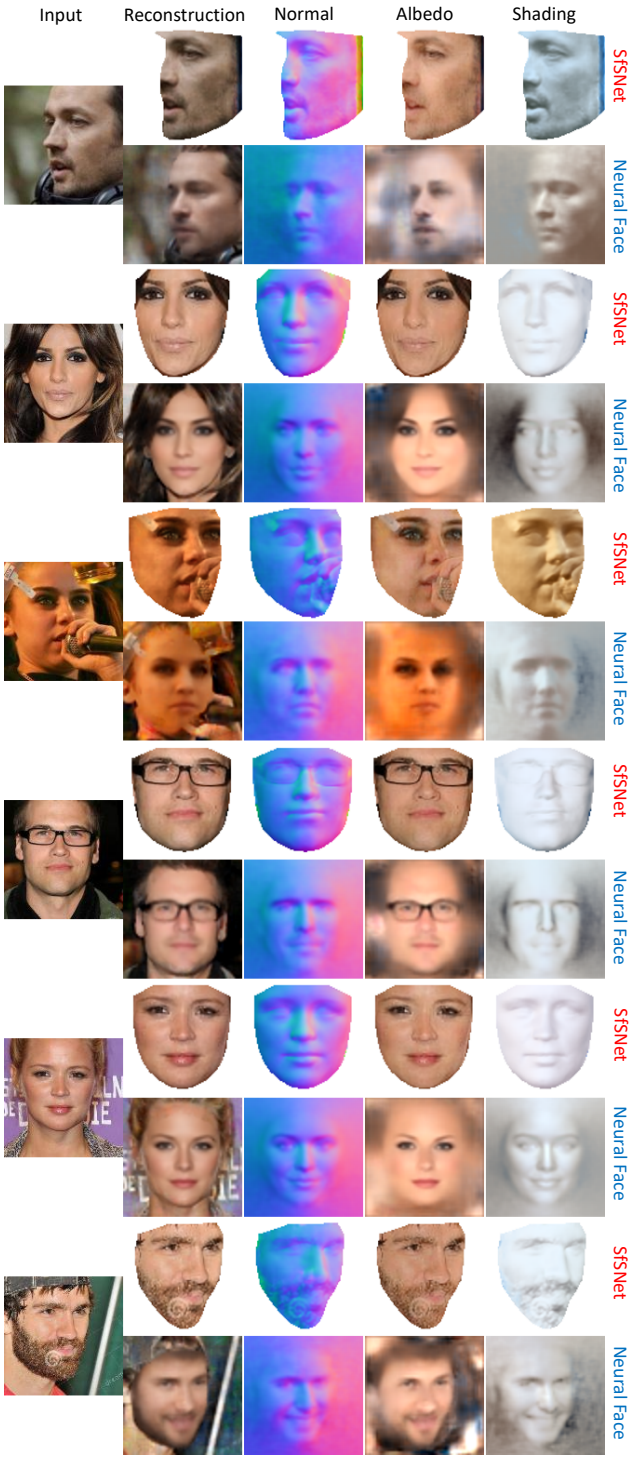


Figure 10-13: **Inverse Rendering. SfSNet vs 'Neural Face'** [141] on the images showcased by the authors. (Best viewed in color)

10.2 Inverse Rendering of an Indoor Scene

In this appendix we provide the details of our network architecture and the loss functions along with more qualitative evaluations. Specifically, in Section 10.2.1 we discuss the details of the IRN and RAR network architectures for reproducibility. Details of our training loss functions on real data are provided in Section 10.2.2. In Section 10.2.5 we present additional qualitative evaluations.

10.2.1 Network Architectures

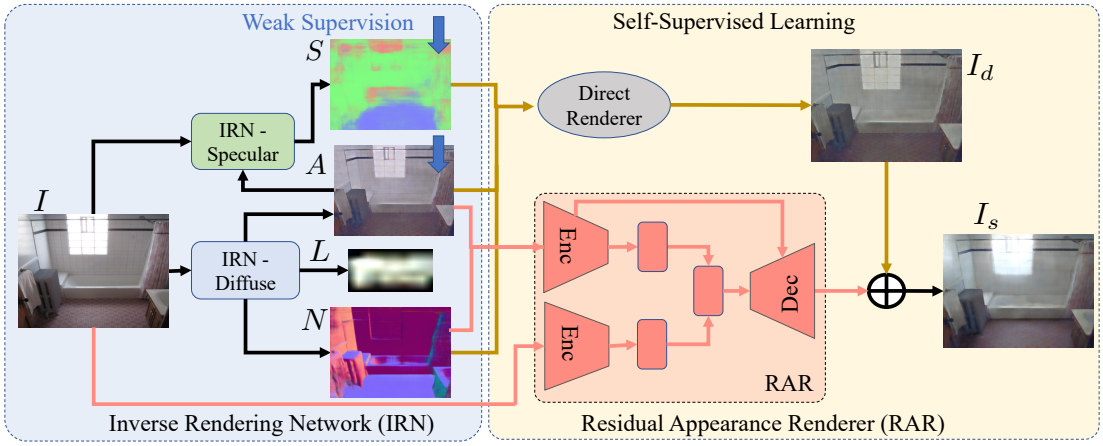


Figure 10-14: **Our Proposed Architecture**.

Our proposed Inverse Rendering Network (IRN), shown again in Figure 10-14 for reference, consists of two modules IRN-Diffuse and IRN-Specular. IRN is trained on real data using the Residual Appearance Renderer (RAR), which learns to capture the complex appearance effects(*e.g.* inter-reflection, cast shadows, near-field illumination, and realistic shading). Next, we describe each of the following modules, IRN-Diffuse, IRN-Specular and RAR.

IRN-Diffuse

In Figure 10-15 we present the network architecture of IRN-Diffuse. The input to IRN-Diffuse is an image of spatial resolution 240×320, and the output is an albedo and normal map of same spatial resolution along with a 18×36 resolution environment map. We provide the details of each of the

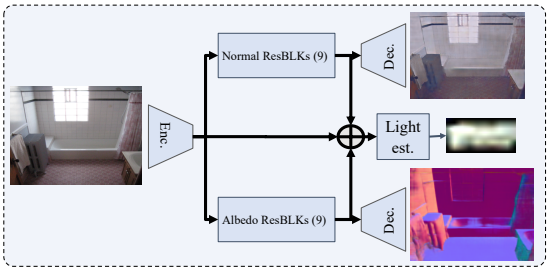


Figure 10-15: **IRN-Diffuse**.

blocks of IRN-Diffuse.

'Enc': C64(k7) - C*128(k3) - C*256(k3)

'CN(kS)' denotes convolution layers with N $S \times S$ filters with stride 1, followed by Batch Normalization and ReLU. 'C*N(kS)' denotes convolution layers with N $S \times S$ filters with stride 2, followed by Batch Normalization and ReLU. The output of 'Enc' layer produces a blob of spatial resolution $256 \times 60 \times 80$.

'Normal ResBLKs': 9 ResBLK

This consists of 9 Residual Blocks, 'ResBLK's, which operate at a spatial resolution of $256 \times 60 \times 80$. Each 'ResBLK' consists of Conv256(k3) - BN -ReLU - Conv256(k3) - BN, where 'ConvN(kS)' and 'BN' denote convolution layers with N $S \times S$ filters of stride 1 and Batch Normalization.

'Albedo ResBLKs': Same as 'Normal Residual Blocks' (weights are not shared).

'Dec.': CD*128(k3)-CD*64(k3)-Co3(k7)

'CD*N(kS)' denotes Transposed Convolution layers with N $S \times S$ filters with stride 2, followed by Batch Normalization and ReLU. 'CN(kS)' denotes convolution layers with N $S \times S$ filters with stride 1, followed by Batch Normalization and ReLU. The last layer Co3k(7) consists of only convolution layers of 3 7×7 filters, followed by Tanh layer.

'Light Est.': It first concatenates the responses of 'Enc', 'Normal ResBLKs' and 'Albedo ResBLKs' to produce a blob of spatial resolution $768 \times 60 \times 80$. It is further processed by the following module:

C256(k1) - C*256(k3) - C*128(k3) - C*3(k3) - BU(18,36)

'CN(kS)' denotes convolution layers with N $S \times S$ filters with stride 1, followed by Batch Normalization and ReLU. 'C*N(kS)' denotes convolution layers with N $S \times S$ filters with stride 2, followed by Batch Normalization and ReLU. BU(18,36) upsamples the response to produce $18 \times 36 \times 3$ resolution environment map.

IRN-Specular

IRN-Specular consists of an U-Net architecture with image and albedo predicted by the IRN-Diffuse as it's input. Like U-Net, skip connections exist between encoder and decoder of the U-Net.

'Encoder': C64(k3) - C64(k1) - C*64(k3) - C64(k1) -C*128(k3) - C128(k1) - C*256(k3) - C256(k1) - C*512(k3)

'Decoder': CU512(k3) - CU256(k3) - CU128(k3) - CU64(k3) - Co3(k1)

'CN(kS)' denotes convolution layers with N $S \times S$ filters with stride 1, followed by Batch Normalization and ReLU. 'C*N(kS)' denotes convolution layers with N $S \times S$ filters with stride 2, followed by Batch Normalization and ReLU. 'CUN(kS)' represents a bilinear up-sampling layer , followed by convolution layers with N $S \times S$ filters with stride 1, and Batch Normalization and ReLU. 'Co3(k1)' consists of 3 1×1 convolution filters, followed by Tanh layer, to produce Normal or Albedo. Skip-connections exists between 'C*N(k3)' layers of encoder and 'CUN(k3)' layers of decoder.

RAR

As shown in Figure 10-14, Residual Appearance Renderer (RAR) consists of a U-Net architecture and a convolution encoder. The U-Net consists of the following architecture, with normals and albedo as its input:

'Encoder': C64(k3) - C*64(k3) - C*128(k3) - C*256(k3) - C*512(k3)

'Decoder': CU512(k3) - CU256(k3) - CU128(k3) - CU64(k3) - Co3(k1)

'CN(kS)' denotes convolution layers with N $S \times S$ filters with stride 1, followed by Batch Normalization and ReLU. 'C*N(kS)' denotes convolution layers with N $S \times S$ filters with stride 2, followed by Batch Normalization and ReLU. 'CUN(kS)' represents a bilinear up-sampling layer , followed by convolution layers with N $S \times S$ filters with stride 1, followed by Batch Normalization and ReLU. 'Co3(k1)' consists of 3 1×1 convolution filters to produce Normal or Albedo. Skip-connections exists between 'C*N(k3)' layers of encoder and 'CUN(k3)' layers of decoder. The encoder 'Enc' that encodes image features to a latent $D = 300$ dimensional subspace is given by: 'Enc': C64(k7) - C*128(k3) - C*256(k3) - C128(k1) - C64(k3) - C*32(k3) - C*16(k3) - MLP(300)

'CN(kS)' denotes convolution layers with N $S \times S$ filters with stride 1, followed by Batch Normalization and ReLU. 'C*N(kS)' denotes convolution layers with N $S \times S$ filters with stride 2, followed by Batch Normalization and ReLU. MLP(300) takes the response of the previous layer and outputs a 300 dimensional feature, which is concatenated with the last layer of the U-Net 'Encoder'.

Environment Map Estimator

As discussed in Section 3.1 of the main paper, the ground-truth environment map is estimated from the image, ground-truth albedo and normal using a deep network $h_e(\cdot, \Theta_e)$. The detailed architecture of this network is presented below:

C64(k7) - C*128(k3) - C*256(k3) - 4 ResBLKS - C256(k1) - C*256(k3) - C*128(k3) - C*3(k3) -

BU(18,36),

where, 'CN(kS)' denotes convolution layers with N $S \times S$ filters with stride 1, followed by Batch Normalization and ReLU. 'C*N(kS)' denotes convolution layers with N $S \times S$ filters with stride 2, followed by Batch Normalization and ReLU. BU(18,36) upsamples the response to produce $18 \times 36 \times 3$ resolution environment map. Each 'ResBLK' contains Conv256(k3) - BN -ReLU - Conv256(k3) - BN, where 'ConvN(kS)' denotes convolution layers with N $S \times S$ filters of stride 1, 'BN' denoted Batch Normalization.

10.2.2 Training Details

10.2.3 Training with weak supervision over albedo

IIW dataset presents relative reflectance judgments from humans. For any two points R_1 and R_2 on an image, a weighted confidence score classifies R_1 to be same, brighter or darker than R_2. We use these labels to construct a hinge loss for sparse supervision based on WHRD metric presented in [11]. Specifically, if users predict R_1 to be darker than R_2 with confidence w_t, we use a loss $w_t \max(1 + \delta - R_2/R_1, 0)$. If R_1 and R_2 are predicted to have similar reflectance, we use $w_t[\max(R_1/R_2 - 1 - \delta, 0) + \max(R_2/R_1 - 1 - \delta, 0)]$. We observed empirically that this loss function performs better than WHRD metric, which is an L0 version of our loss. We train on real data with the following losses: (i) Psuedo-supervision loss over albedo (L_a), normal (L_n) and lighting (L_e) based on [134], (ii) Photometric Reconstruction loss with RAR (L_u) (iii) Pair-wise weak supervision (L_w). Thus the net loss function is defined as:

$$L = 0.5 * L_a + 0.5 * L_n + 0.1 * L_e + L_u + 30 * L_w. \tag{10.5}$$

Training with weak supervision over normals

We also train on NYUv2 dataset with weak supervision over normals, obtained from Kinect depth data of the scene. We train with the following losses: (i) Psuedo-supervision loss over albedo (L_a) and lighting (L_e) based on [134], (ii) Photometric Reconstruction loss with RAR (L_u) (iii) Supervision (L_w) over kinect normals. Thus the net loss function is defined as:

$$L = 0.2 * L_a + 0.05 * L_e + L_u + 20 * L_w. \tag{10.6}$$

10.2.4 Our SUNCG-PBR Dataset

We present more example images of our SUNCG-PBRS dataset in Figure 10-16. We also compare the renderings of our SUNCG-PBR Dataset with that of PBRS [180], under same illumination condition in Figure 10-17 and 10-18. SUNCG-PBR provides more photo-realistic and less noisy images with specular highlights. Both SUNCG-PBR and PBRS is rendered with Mitsuba [68]. We will release the dataset upon publication.

10.2.5 More Experimental Results

Comparison with SIRFS. We present more detailed qualitative evaluations in this supplementary material. In Figure 10-19 we compare the results of our algorithm with that of SIRFS [6]. SIRFS is an optimization-based method for inverse rendering, which estimates surface normals, albedo and spherical harmonics lighting from a single image. Compared to SIRFS we obtain more accurate normals and better disambiguation of reflectance from shading.

Comparison with CGIntrinsic. In Figure 10-20 we compare the albedo predicted by our method with that of CGIntrinsics [95], which performs intrinsic image decomposition of an image. Intrinsic image decomposition methods do not explicitly recover geometry, illumination or glossiness of the

material, but rather combine them together as shading. In contrast, our goal is to perform a complete inverse rendering which has a wider range of applications in AR/VR.

Evaluation of lighting estimation. In Figure 10-21 we present a qualitative evaluation of lighting estimation by inserting a diffuse hemisphere into the scene and rendering it with the inferred light from the image. We compare this with our implementation of the method proposed by Gardner *et. al.* [45], which also estimates an environment map from a single indoor image. $h_e(\cdot, \Theta_e)$ is a deep network that predicts the environment map given the image, normals, and albedo. 'GT+$h_e(\cdot)$' estimates the environment map given the image, ground-truth normals and albedo, and thus serves as an achievable upper-bound in the quality of the estimated lighting. 'Ours' estimates environment map from an image with IRN. 'Ours+$h_e(\cdot)$' predicts environment map by combining the inferred albedo and normals from IRN to predict lighting with $h_e(\cdot)$. Both 'Ours' and 'Ours+$h_e(\cdot)$' outperform Gardner *et. al.* [45] as they seem to produce more realistic environment maps. 'Ours+$h_e(\cdot)$' improves lighting estimation over 'Ours' by utilizing the predicted albedo and normals to a greater degree.

Our Results and Ablation study. Figure 10-22 shows examples of our results, with the albedo, glossiness segmentation, normal and lighting predicted by the network, as well as the reconstructed image with the direct renderer and the proposed Residual Appearance Renderer (RAR). In Figure 10-23 and 10-24, we perform a detailed ablation study of different components of our method. We show that it is important to train on real data, as networks trained on synthetic data fails to generalize well on real data. We also show that training on real data using Residual Appearance Renderer (RAR), to capture complex appearance effects, significantly improves performance. Finally, incorporating weak supervisions from relative reflectance judgments helps the network to predict uniform albedo across large objects.

Figure 10-16: **Our SUNCG-PBR Dataset.** We provide 235,893 images of a scene assuming specular and diffuse reflectance along with ground truth depth, surface normals, albedo, Phong model parameters, semantic segmentation and glossiness segmentation.

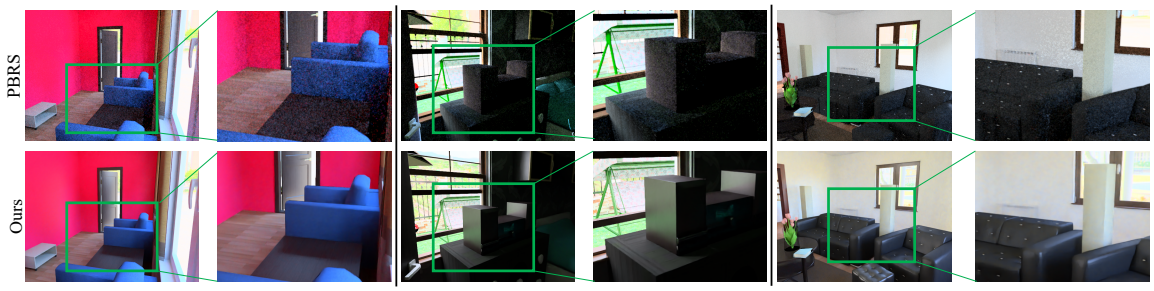


Figure 10-17: **Comparison with PBRS [180].** Our dataset provides more photo-realistic and less noisy images with specular highlights under multiple lighting conditions.

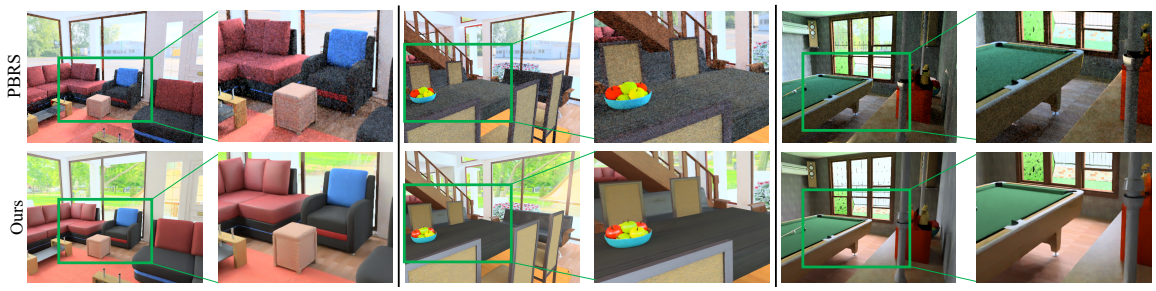


Figure 10-18: **Comparison with PBRS [180].** Our dataset provides more photo-realistic and less noisy images with specular highlights under multiple lighting conditions.

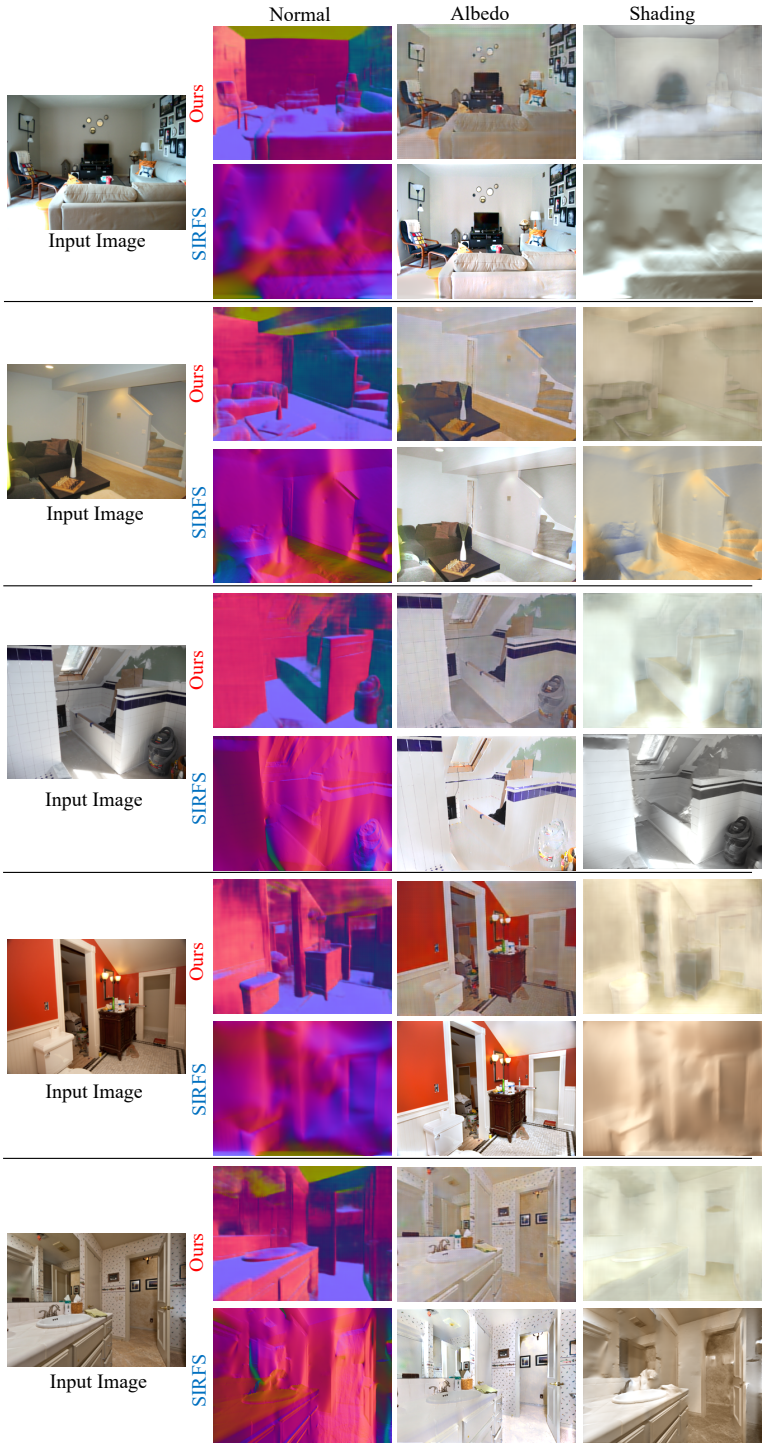


Figure 10-19: **Comparison with SIRFS [6].** Using deep CNNs our method performs better disambiguation of reflectance from shading and predicts better surface normals.

Figure 10-20: **Comparison with CGI (Li *et. al.* [95]).** In comparison with CGI [95], our method performs better disambiguation of reflectance from shading and preserves the texture in the albedo.

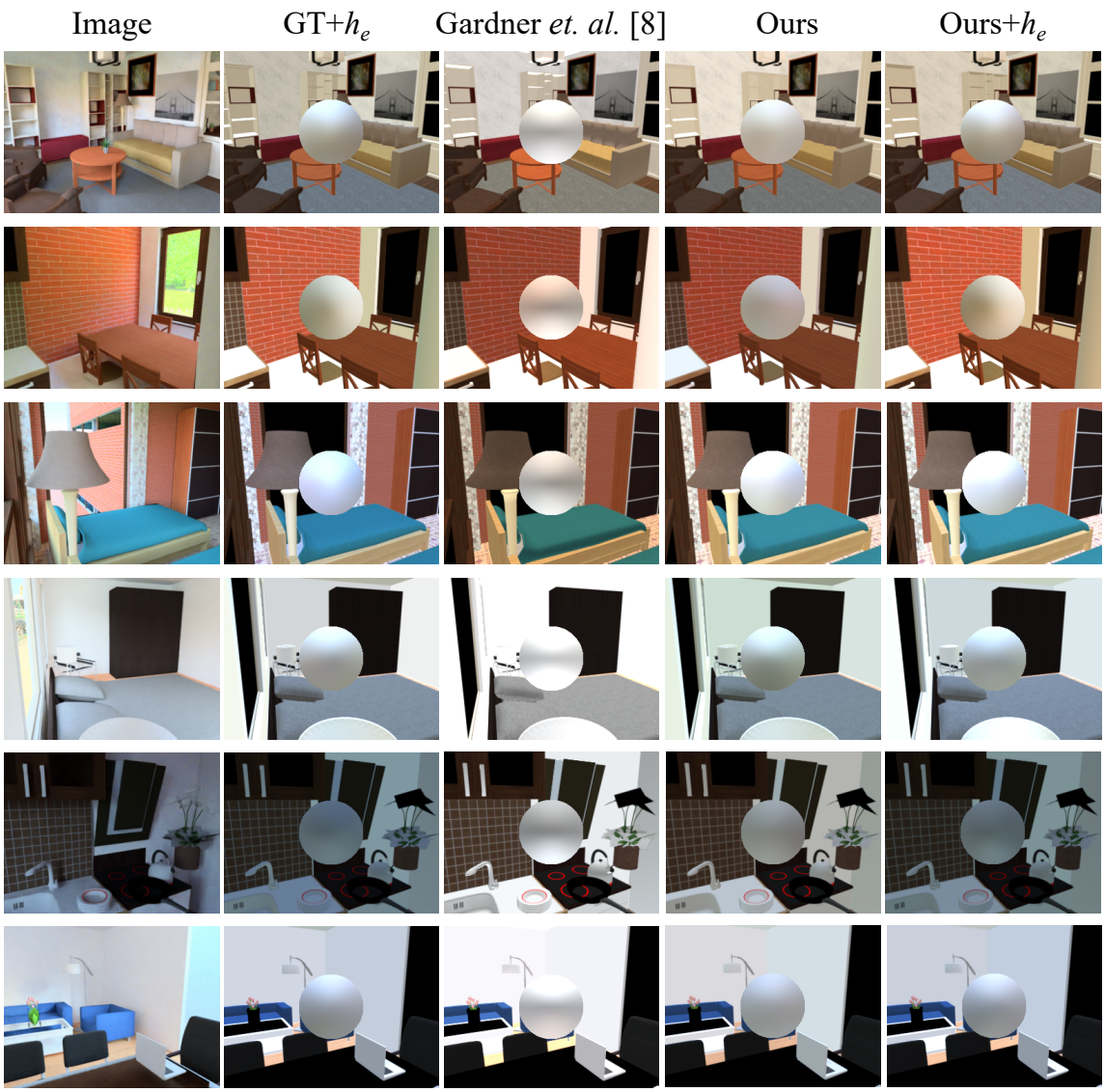


Figure 10-21: **Evaluation of lighting estimation.** We compare with our implementation of Gardner *et. al.* [45]. 'GT+$h_e(\cdot)$' predicts lighting conditioned on the ground-truth normals and albedo. 'Ours+$h_e(\cdot)$' predicts the environment map by conditioning it on the albedo and normals inferred by IRN.

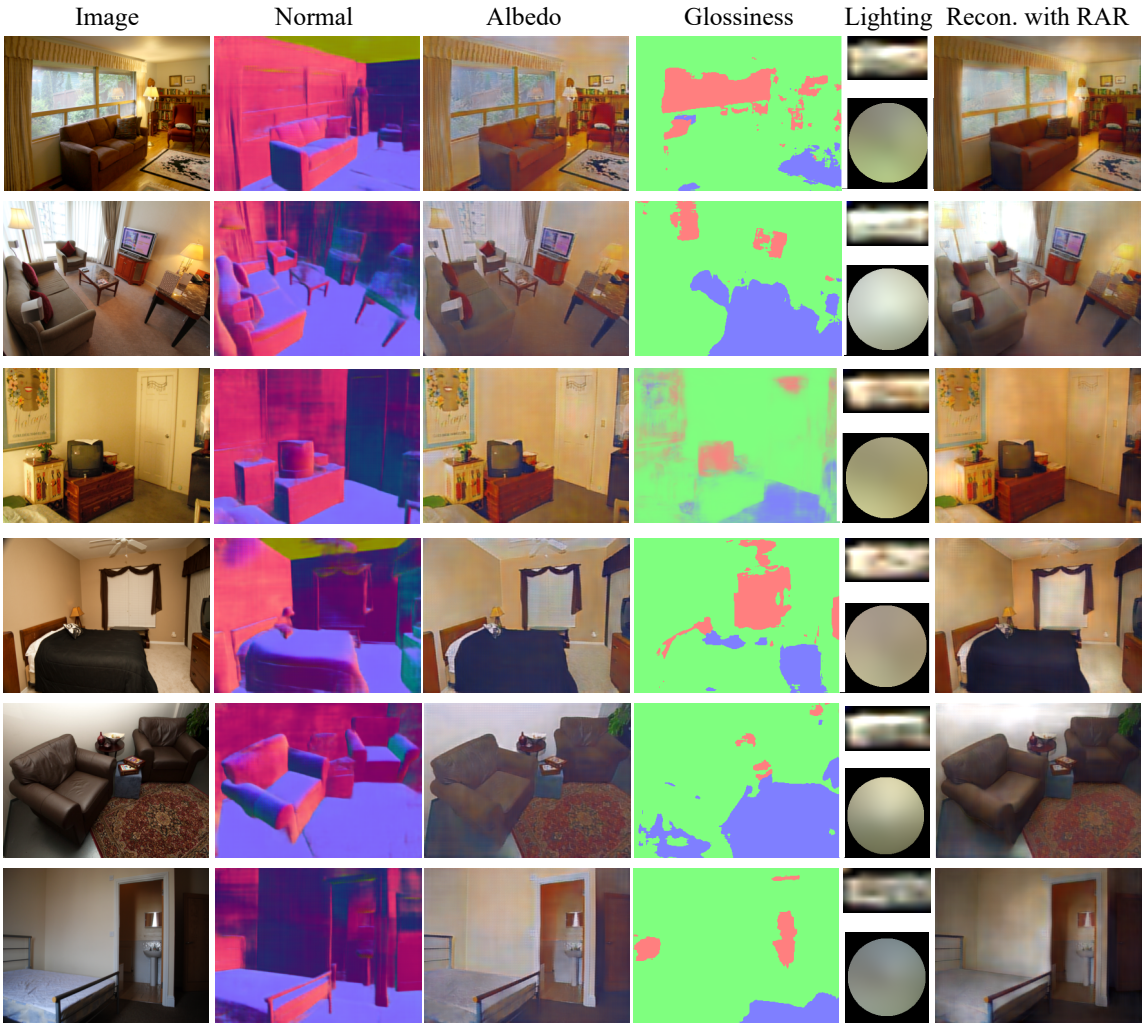


Figure 10-22: **Our Result.** We show the estimated intrinsic components; normals, albedo, glossiness segmentation (matte-blue, glossy-red and semi-glossy-green) and lighting predicted by the network, along with the reconstructed image with our direct renderer and the RAR.

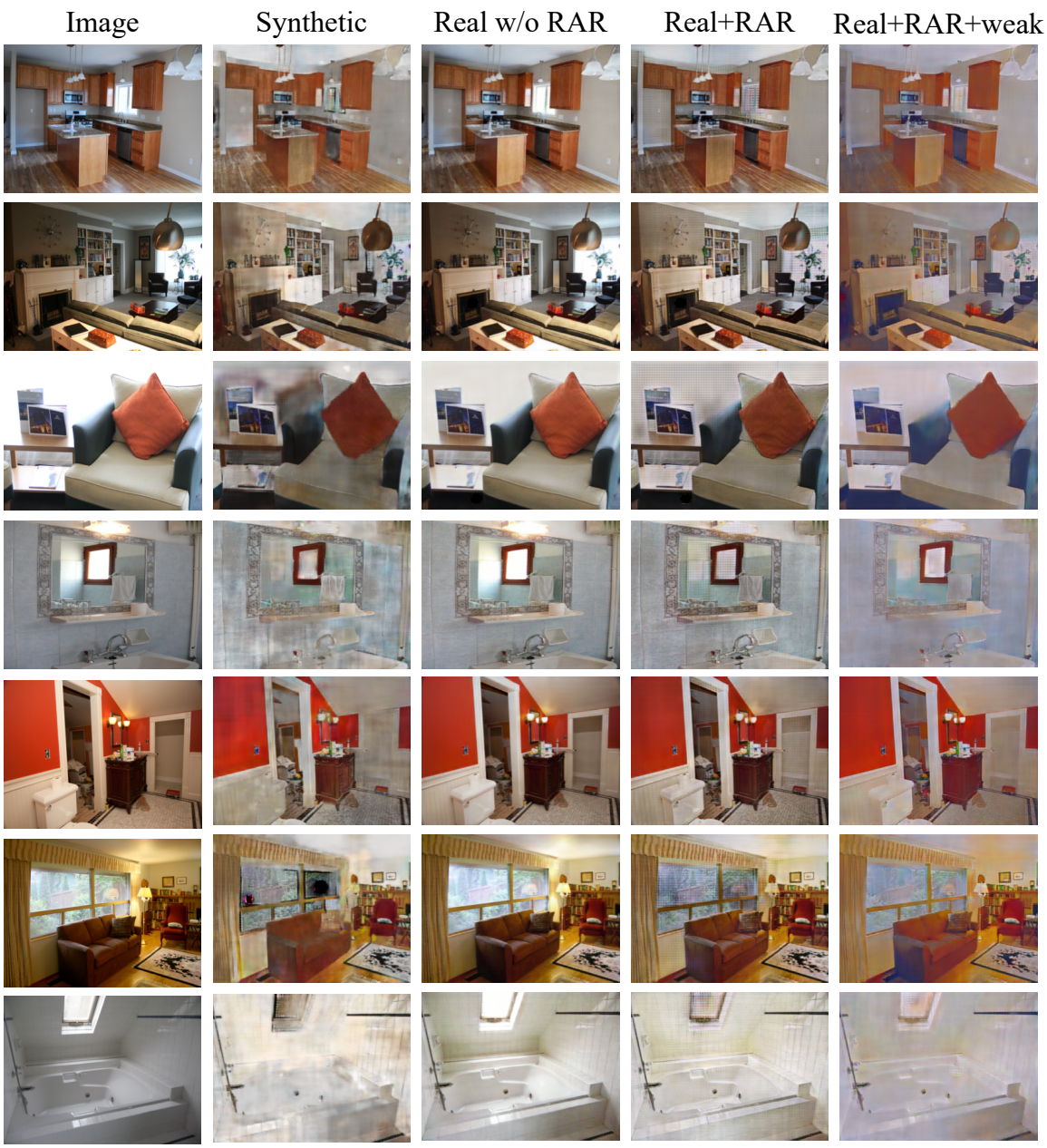


Figure 10-23: **Ablation Study**. We present the predicted albedo for each input image (in column 1) in column 2-5. We show the albedo predicted by IRN trained on our SUNCG-PBR only in column 2. In column 3 and 4 we show the albedo predicted by IRN finetuned on real data without and with RAR respectively. We present the albedo predicted by IRN, trained on real data with RAR and weak supervision, in column 5.

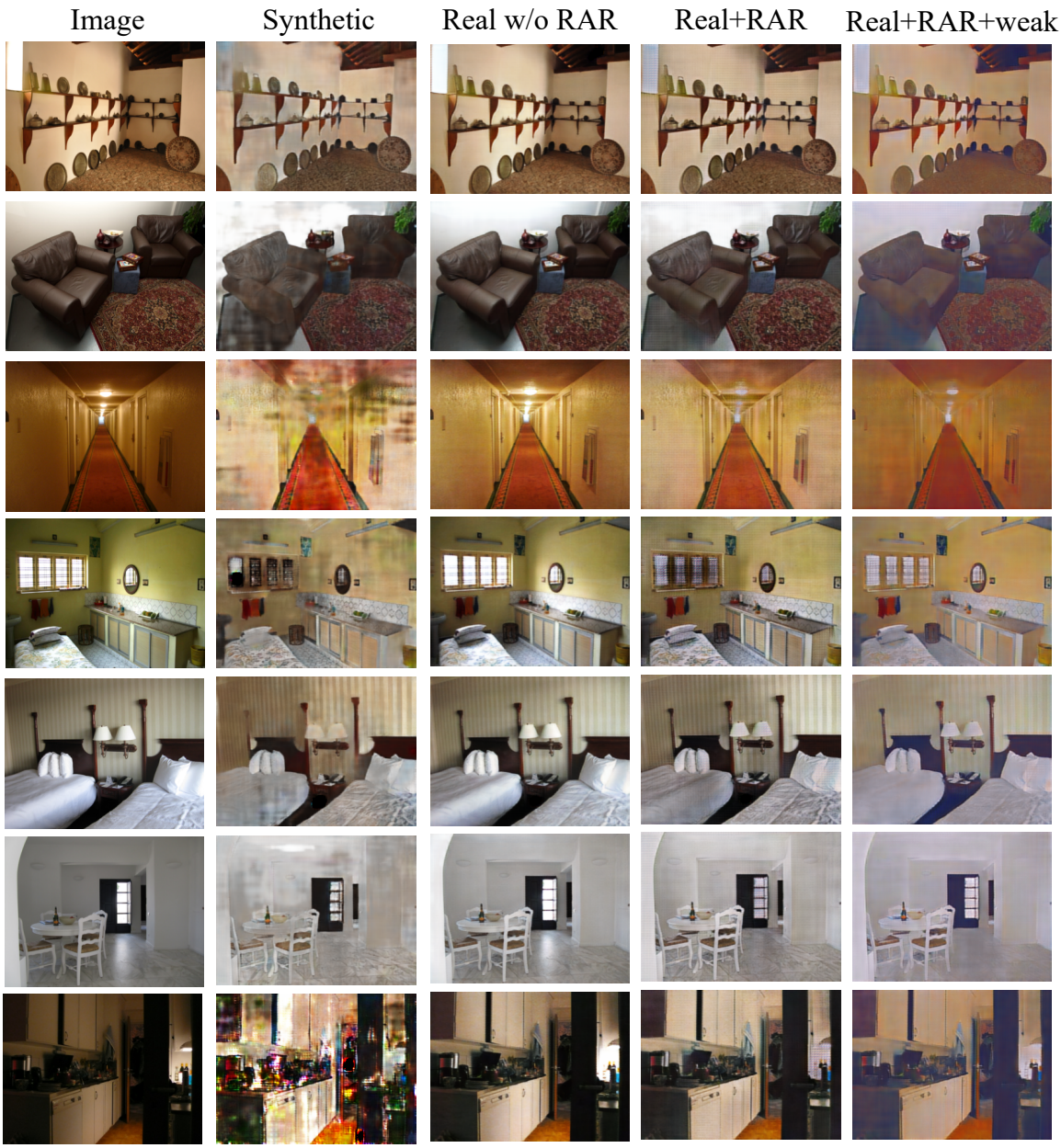


Figure 10-24: **Ablation Study**. We present the predicted albedo for each input image (in column 1) in column 2-5. We show the albedo predicted by IRN trained on our SUNCG-PBR only in column 2. In column 3 and 4 we show the albedo predicted by IRN finetuned on real data without and with RAR respectively. We present the albedo predicted by IRN, trained on real data with RAR and weak supervision, in column 5.

Bibliography

[1] Jens Ackermann and Michael Goesele. A survey of photometric stereo techniques. *Foundations and Trends® in Computer Graphics and Vision*, 9(3-4):149–254, 2015. 81

[2] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski. Building rome in a day. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 72–79. IEEE, 2009. 2, 57, 60

[3] Neil G Alldrin, Satya P Mallick, and David J Kriegman. Resolving the generalized bas-relief ambiguity by entropy minimization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007. 50, 58, 82, 89, 94

[4] Mica Arie-Nachimson, Shahar Z Kovalsky, Ira Kemelmacher-Shlizerman, Amit Singer, and Ronen Basri. Global motion estimation from point matches. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 81–88. IEEE, 2012. 58, 60, 63, 74

[5] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5965–5974, 2016. 25, 26

[6] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015. ix, xi, 12, 34, 42, 123, 127

[7] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1670–1687, 2015. 16, 27

[8] Ronen Basri, David Jacobs, and Ira Kemelmacher. Photometric stereo with general, unknown lighting. *International Journal of Computer Vision*, 72(3):239–257, 2007. 59, 81

[9] Ronen Basri and David W Jacobs. Lambertian reflectance and linear subspaces. *IEEE transactions on pattern analysis and machine intelligence*, 25(2):218–233, 2003. 19

[10] Peter N Belhumeur, David J Kriegman, and Alan L Yuille. The bas-relief ambiguity. *International journal of computer vision*, 35(1):33–44, 1999. 58

[11] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics (TOG)*, 33(4):159, 2014. vii, ix, 14, 15, 34, 39, 42, 43, 44, 46, 47, 122

[12] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. OpenSurfaces: A richly annotated catalog of surface appearance. *ACM Transactions on Graphics (SIGGRAPH)*, 32(4), 2013. 14, 34, 39

[13] Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Global optimality of local search for low rank matrix recovery. *arXiv preprint arXiv:1605.07221*, 2016. 55

[14] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999. 12, 15, 22

[15] Robert C Bolles and Martin A Fischler. A ransac-based approach to model fitting and its application to finding cylinders in range data. In *IJCAI*, volume 1981, pages 637–643, 1981. 73

[16] Ruediger Borsdorf. *Structured matrix nearness problems: Theory and algorithms*. PhD thesis, Citeseer, 2012. 57

[17] Nicolas Boumal and P-A Absil. Low-rank matrix completion via preconditioned optimization on the grassmann manifold. *Linear Algebra and its Applications*, 475:200–239, 2015. 54

[18] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011. 51, 61, 69, 72, 83, 86, 89

[19] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 54

[20] Aeron M Buchanan and Andrew W Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 316–322. IEEE, 2005. 53

[21] Ricardo Cabral, Fernando De La Torre, João P Costeira, and Alexandre Bernardino. Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2488–2495, 2013. 50

[22] Ricardo S Cabral, Fernando Torre, João P Costeira, and Alexandre Bernardino. Matrix completion for multi-label image classification. In *Advances in Neural Information Processing Systems*, pages 190–198, 2011. 50, 59

[23] James A Cadzow. Signal enhancement-a composite property mapping algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(1):49–62, 1988. 56

[24] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956–1982, 2010. 54, 89

[25] Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010. 55

[26] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009. 55, 58

[27] Menglei Chai, Linjie Luo, Kalyan Sunkavalli, Nathan Carr, Sunil Hadap, and Kun Zhou. High-quality hair modeling from a single portrait photo. *ACM Transactions on Graphics (TOG)*, 34(6):204, 2015. 12

[28] Chakravarty R Alla Chaitanya, Anton S Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. Interactive reconstruction of monte carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics (TOG)*, 36(4):98, 2017. 33, 41

[29] Manmohan Chandraker, Sameer Agarwal, and David Kriegman. Shadowcuts: Photometric stereo with shadows. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 59

[30] Manmohan Krishna Chandraker, Chandraker Fredrik Kahl, and David J Kriegman. Reflections on the generalized bas-relief ambiguity. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 788–795. IEEE, 2005. 58, 89

[31] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 20

[32] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 521–528, 2013. 58

[33] Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. Inverse transport networks. *arXiv preprint arXiv:1809.10820*, 2018. 14

[34] Ching-Hui Chen, Vishal M Patel, and Rama Chellappa. Matrix completion for resolving label ambiguity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4110–4118, 2015. 50

[35] Moody T Chu, Robert E Funderlic, and Robert J Plemmons. Structured low rank approximation. *Linear algebra and its applications*, 366:157–172, 2003. 56

[36] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 864–872, 2015. 58

[37] Mark A Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):608–622, 2016. 53

[38] Ondrej Drbohlav and M Chaniler. Can two specular pixels calibrate photometric stereo? In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1850–1857. IEEE, 2005. 58, 89

[39] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *International Conference on Computer Vision (ICCV)*, pages 2650–2658, 2015. 13, 33

[40] Paolo Favaro and Thoma Papadhimitri. A closed-form solution to uncalibrated photometric stereo via diffuse maxima. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 821–828. IEEE, 2012. 58, 59, 82, 89

[41] M Fazel, E Candes, B Recht, and P Parrilo. Compressed sensing and robust recovery of low rank matrices. In *2008 42nd Asilomar Conference on Signals, Systems and Computers*, pages 1043–1047. IEEE, 2008. 55

[42] Maryam Fazel. *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University, 2002. 54

[43] Robert T. Frankot and Rama Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 10(4):439–451, 1988. 81

[44] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 13

[45] 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):176, 2017. ix, xi, 13, 33, 34, 44, 45, 123, 129

[46] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. *arXiv preprint arXiv:1605.07272*, 2016. 55

[47] Athinodoros S Georghiades. Incorporating the torrance and sparrow model of reflectance in uncalibrated photometric stereo. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 816–823. Ieee, 2003. 59

[48] Stamatios Georgoulis, Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Luc Van Gool, and Tinne Tuytelaars. DeLight-Net: Decomposing reflectance maps into specular materials and natural illumination. *arXiv preprint arXiv:1603.08240*, 2016. 13

[49] Thomas Goldstein, Paul Hand, Choongbum Lee, Vladislav Voroninski, and Stefano Soatto. Shapefit and shapekick for robust, scalable structure from motion. In *European Conference on Computer Vision*, pages 289–304. Springer, 2016. 74, 75

[50] Tom Goldstein, Brendan O'Donoghue, Simon Setzer, and Richard Baraniuk. Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences*, 7(3):1588–1623, 2014. 51, 69, 72, 83, 86, 89

[51] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010. 23, 27

[52] Christian Grussler, Anders Rantzer, and Pontus Giselsson. Low-rank optimization with convex constraints. *arXiv preprint arXiv:1606.01793*, 2016. 57

[53] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 82, 83

[54] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International journal of computer vision*, 103(3):267–305, 2013. 58

[55] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 62

[56] Hideki Hayakawa. Photometric stereo under a light source with arbitrary motion. *JOSA A*, 11(11):3079–3089, 1994. 81, 82

[57] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016. 22

[58] Gim Hee Lee, Friedrich Faundorfer, and Marc Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2746–2753, 2013. 3

[59] Jared Heinly, Johannes L Schonberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3287–3295, 2015. 2, 57

[60] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 13

[61] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977. 61, 68

[62] Yao Hu, Debing Zhang, Jieping Ye, Xuelong Li, and Xiaofei He. Fast and accurate matrix completion via truncated nuclear norm regularization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(9):2117–2130, 2013. 50, 55, 56, 59, 83, 85, 86

[63] Yao Hu, Debing Zhang, Jieping Ye, Xuelong Li, and Xiaofei He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2117–2130, 2013. 58

[64] Mariya Ishteva, Konstantin Usevich, and Ivan Markovsky. Factorization approach to structured low-rank approximation with applications. *SIAM Journal on Matrix Analysis and Applications*, 35(3):1180–1204, 2014. 56

[65] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016. 14, 20, 30, 103

[66] David Jacobs. Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 206–212. IEEE, 1997. 50, 58

[67] David W Jacobs. Linear fitting with missing data for structure-from-motion. *Computer Vision and Image Understanding*, 82(1):57–81, 2001. 58

[68] Wenzel Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 6, 40, 123

[69] Hui Ji, Chaoqiang Liu, Zuowei Shen, and Yuhong Xu. Robust video denoising using low rank matrix completion. In *CVPR*, pages 1791–1798. Citeseer, 2010. 50

[70] Shuiwang Ji and Jieping Ye. An accelerated gradient method for trace norm minimization. In *Proceedings of the 26th annual international conference on machine learning*, pages 457–464. ACM, 2009. 55

[71] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 14

[72] Micah K Johnson and Hany Farid. Exposing digital forgeries in complex lighting environments. *IEEE Transactions on Information Forensics and Security*, 2(3):450–461, 2007. 3

[73] Neil Joshi, Ira Kemelmacher, and Ian Simon. Photometric stereo dataset. url=http://courses.cs.washington.edu/courses/cse455/10wi/projects/project4/, 2015. 91, 94

[74] Arnav Kapur, Kshitij Marwah, and Gil Alterovitz. Gene expression prediction using low-rank matrix completion. *BMC bioinformatics*, 17(1):243, 2016. 50

[75] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics (TOG)*, 30(6):157, 2011. 3

[76] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering synthetic objects into legacy photographs. In *ACM Transactions on Graphics (SIGGRAPH Asia)*, pages 157:1–157:12, 2011.

[77] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3907–3916, 2018. 14

[78] Ira Kemelmacher-Shlizerman and Ronen Basri. 3d face reconstruction from a single image using a single reference face shape. *IEEE transactions on pattern analysis and machine intelligence*, 33(2):394–405, 2011. 12, 17, 20

[79] Ira Kemelmacher-Shlizerman and Steven M Seitz. Face reconstruction in the wild. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1746–1753. IEEE, 2011. 12, 20

[80] Christian Kerl, Jurgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2100–2106. Citeseer, 2013. 3

[81] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(Jul):2057–2078, 2010. 54

[82] Erum Arif Khan, Erik Reinhard, Roland W Fleming, and Heinrich H Bülthoff. Image-based material editing. *ACM Transactions on Graphics (TOG)*, 25(3):654–663, 2006.

[83] Kihwan Kim, Jinwei Gu, Stephen Tyree, Pavlo Molchanov, Matthias Nießner, and Jan Kautz. A lightweight approach for on-the-fly reflectance estimation. In *International Conference on Computer Vision (ICCV)*, pages 20–28, 2017. 13, 33

[84] Iasonas Kokkinos. Ubernet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 25, 26

[85] Vladimir Koltchinskii, Karim Lounici, and Alexandre B Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, pages 2302–2329, 2011. 55

[86] Nikos Komodakis. Image completion using global optimization. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 442–452. IEEE, 2006. 50

[87] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008. 50

[88] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015. 12

[89] Rajiv Kumar, Curt Da Silva, Okan Akalin, Aleksandr Y Aravkin, Hassan Mansour, Benjamin Recht, and Felix J Herrmann. Efficient matrix completion for seismic data reconstruction. *Geophysics*, 80(5):V97–V114, 2015. 50

[90] Eric P Lafortune and Yves D Willems. Using the modified Phong reflectance model for physically based rendering. 1994. 33, 35, 41

[91] Samuli Laine, Tero Karras, Timo Aila, Antti Herva, Shunsuke Saito, Ronald Yu, Hao Li, and Jaakko Lehtinen. Production-level facial performance capture using deep convolutional neural networks. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, page 10. ACM, 2017. 13

[92] Louis Lettry, Kenneth Vanhoey, and Luc Van Gool. DARN: a deep adversarial residual network for intrinsic image decomposition. In *IEEE Workshop on Applications of Computer Vision (WACV)*, 2018. 15

[93] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. 37(6):222:1–222:11, 2018. 14

[94] Z. Li, Z. Xu, R. Ramamoorthi, K. Sunkavalli, and M.K. Chandraker. Learning to reconstruct shape and spatially-varying reflectance with a single image. In *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2018. 13

[95] Zhengqi Li and Noah Snavely. CGIntrinsics: Better intrinsic image decomposition through physically-based rendering. *European Conference on Computer Vision (ECCV)*, 2018. ix, xi, 15, 34, 39, 43, 44, 123, 128

[96] Zhengqi Li and Noah Snavely. Learning intrinsic image decomposition from watching the world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 15, 34, 44

[97] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: SVBRDF acquisition with a single mobile phone image. In *European Conference on Computer Vision (ECCV)*, 2018. 13, 33

[98] Guilin Liu, Duygu Ceylan, Ersin Yumer, Jimei Yang, and Jyh-Ming Lien. Material editing using a physically based rendering network. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2280–2288. IEEE, 2017. 3

[99] Guilin Liu, Duygu Ceylan, Ersin Yumer, Jimei Yang, and Jyh-Ming Lien. Material editing using a physically based rendering network. In *International Conference on Computer Vision (ICCV)*, 2017. 13, 33, 38

[100] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013. 50

[101] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017. 14

[102] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 15, 18, 22

[103] Stephen Lombardi and Ko Nishino. Reflectance and illumination recovery in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(1):129–141, 2016. 12

[104] Manolis IA Lourakis and Antonis A Argyros. Sba: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software (TOMS)*, 36(1):2, 2009. 78

[105] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 73

[106] Daniel Martinec and Tomas Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 58, 60

[107] Roberto Mecca, Ariel Tankus, Aaron Wetzler, and Alfred M Bruckstein. A direct differential approach to photometric stereo with perspective viewing. *SIAM Journal on Imaging Sciences*, 7(2):579–612, 2014. 59

[108] Abhimitra Meka, Maxim Maximov, Michael Zollhoefer, Avishek Chatterjee, Hans-Peter Seidel, Christian Richardt, and Christian Theobalt. LIME: Live intrinsic material estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 13, 33

[109] Kaushik Mitra, Sameer Sheorey, and Rama Chellappa. Large-scale matrix factorization with missing data under additional constraints. In *Advances in Neural Information Processing Systems*, pages 1651–1659, 2010. 58

[110] Takuya Narihira, Michael Maire, and Stella X Yu. Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *International Conference on Computer Vision (ICCV)*, pages 2992–2992, 2015. 15

[111] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision (ECCV)*, 2012. 14, 44

[112] Sahand Negahban and Martin J Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13(May):1665–1697, 2012. 55

[113] Thomas Nestmeyer and Peter V Gehler. Reflectance adaptive filtering improves intrinsic image estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 4, 2017. 15, 34, 43, 44

[114] Tae-Hyun Oh, Hyeongwoo Kim, Yu-Wing Tai, Jean-Charles Bazin, and In So Kweon. Partial sum minimization of singular values in rpca for low-level vision. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 145–152. IEEE, 2013. 59

[115] Takahiro Okabe, Imari Sato, and Yoichi Sato. Attached shadow coding: Estimating surface normals from shadows under unknown reflectance and lighting conditions. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1693–1700. IEEE, 2009. 59

[116] Takayuki Okatani and Koichiro Deguchi. On the wiberg algorithm for matrix factorization in the presence of missing components. *International Journal of Computer Vision*, 72(3):329–337, 2007. 54, 58

[117] Takayuki Okatani, Takahiro Yoshida, and Koichiro Deguchi. Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms. In *2011 International Conference on Computer Vision*, pages 842–849. IEEE, 2011. 54

[118] Takayuki Okatani, Takahiro Yoshida, and Koichiro Deguchi. Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 842–849. IEEE, 2011. 59

[119] Geoffrey Oxholm and Ko Nishino. Shape and reflectance from natural illumination. In *European Conference on Computer Vision (ECCV)*, pages 528–541. Springer, 2012. 12, 16

[120] Onur Ozyesil and Amit Singer. Robust camera location estimation by convex programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2674–2683, 2015. 58, 60, 73, 75

[121] Onur Ozyesil, Amit Singer, and Ronen Basri. Stable camera motion estimation using convex programming. *SIAM Journal on Imaging Sciences*, 8(2):1220–1262, 2015. 58, 60

[122] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975. 93

[123] Emmanuel Prados and Olivier Faugeras. Shape from shading. In *Handbook of mathematical models in computer vision*, pages 375–388. Springer, 2006. 12, 16

[124] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D Castillo, and Rama Chellappa. An all-in-one convolutional neural network for face analysis. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 17–24. IEEE, 2017. 22

[125] Christopher Rasmussen and Thommen Korah. Spatiotemporal inpainting for recovering texture maps of partially occluded building facades. In *IEEE International Conference on Image Processing 2005*, volume 3, pages III–125. IEEE, 2005. 50

[126] Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(Dec):3413–3430, 2011. 55

[127] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010. 55

[128] Jasson DM Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719. ACM, 2005. 50

[129] 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 (MICCAI)*, pages 234–241. Springer, 2015. 7, 14, 20, 36, 39

[130] Joseph Roth, Yiying Tong, and Xiaoming Liu. Adaptive 3d face reconstruction from unconstrained photo collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4197–4206, 2016. 2, 12

[131] Shunsuke Saito, Tianye Li, and Hao Li. Real-time facial segmentation and performance capture from rgb input. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 12

[132] Éric Schost and Pierre-Jean Spaenlehauer. A quadratically convergent algorithm for structured low-rank approximation. *Foundations of Computational Mathematics*, 16(2):457–492, 2016. 56

[133] Matan Sela, Elad Richardson, and Ron Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. *arXiv preprint arXiv:1703.10131*, 2017. viii, x, 13, 17, 18, 19, 23, 26, 27, 29, 103, 106, 109, 110, 111, 112

[134] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David W. Jacobs. Sfsnet: Learning shape, refectance and illuminance of faces in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 33, 36, 38, 40, 122

[135] Sohil Shah, Abhay Kumar, David Jacobs, Christoph Studer, and Tom Goldstein. Biconvex relaxation for semidefinite programming in computer vision. *arXiv preprint arXiv:1605.09527*, 2016. 57

[136] Evan Shelhamer, Jonathan T Barron, and Trevor Darrell. Scene intrinsics and depth from a single image. In *International Conference on Computer Vision, Workshops (ICCV-W)*, pages 37–44, 2015. 12

[137] Boxin Shi, Yasuyuki Matsushita, Yichen Wei, Chao Xu, and Ping Tan. Self-calibrating photometric stereo. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1118–1125. IEEE, 2010. 58, 89

[138] Jian Shi, Yue Dong, Hao Su, and X Yu Stella. Learning non-lambertian object intrinsics across shapenet categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5844–5853, 2017. 13, 15

[139] Jian Shi, Yue Dong, Hao Su, and X Yu Stella. Learning non-lambertian object intrinsics across shapenet categories. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5844–5853. IEEE, 2017. 14, 20, 30

[140] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, 2013. 44

[141] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras. Neural face editing with intrinsic image disentangling. In *Computer Vision and Pattern Recognition, 2017. CVPR 2017. IEEE Conference on*, pages –. IEEE, 2017. viii, xi, 13, 14, 18, 19, 22, 23, 24, 29, 30, 102, 106, 117

[142] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5444–5453, 2017. 33, 38

[143] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006. 2, 57, 60, 75

[144] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. ix, 14, 33, 40, 41

[145] Marco Speicher, Sebastian Cucerca, and Antonio Krüger. Vrshop: A mobile interactive virtual reality shopping environment combining the benefits of on-and offline shopping. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):102, 2017. 3

[146] Harald Steck. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 713–722. ACM, 2010. 50

[147] Peter Sturm and Bill Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European conference on computer vision*, pages 709–720. Springer, 1996. 58

[148] Kalyan Sunkavalli, Todd Zickler, and Hanspeter Pfister. Visibility subspaces: Uncalibrated photometric stereo with shadows. In *European Conference on Computer Vision*, pages 251–264. Springer, 2010. 59

[149] Supasorn Suwajanakorn, Ira Kemelmacher-Shlizerman, and Steven M Seitz. Total moving face reconstruction. In *European Conference on Computer Vision*, pages 796–812. Springer, 2014. 2

[150] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Matrix factorization and neighbor based algorithms for the netflix prize problem. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 267–274. ACM, 2008. 50

[151] Ping Tan. Phong reflectance model. In *Computer Vision*, pages 592–594. Springer, 2014. 93

[152] Ping Tan, Satya P Mallick, Long Quan, David J Kriegman, and Todd Zickler. Isotropy, reciprocity and the generalized bas-relief ambiguity. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 58, 89

[153] Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. Deep lambertian networks. In *Proceedings of the 29th International Conference on Machine Learning, 2012, Edinburgh, Scotland*, 2012. 12

[154] Tatsunori Taniai and Takanori Maehara. Neural inverse rendering for general reflectance photometric stereo. In *Intl. Conf. on Machine Learning (ICML)*, pages 20–28, 2017. 13

[155] Marshall F Tappen, William T Freeman, and Edward H Adelson. Recovering intrinsic images from a single image. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1367–1374, 2003. 12, 16

[156] Ayush Tewari, Michael Zollöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. viii, x, 13, 18, 19, 23, 24, 25, 29, 106, 114, 115, 116

[157] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016. 13

[158] Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 6(615-640):15, 2010. 55

[159] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gerard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. *arXiv preprint arXiv:1612.04904*, 2016. 13, 17

[160] George Trigeorgis, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. Normal Estimation For "in-the-wild" Faces Using Fully Convolutional Networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 13, 18, 19, 23, 25, 26

[161] Bill Triggs. Factorization methods for projective structure and motion. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 845–851. IEEE, 1996. 58

[162] Roberto Tron and René Vidal. Distributed 3-d localization of camera sensor networks from 2-d image measurements. *IEEE Transactions on Automatic Control*, 59(12):3325–3340, 2014. 58, 60

[163] B Tunwattanapong and P Debevec. Interactive image-based relighting with spatially-varying lights. In *ACM Transactions on Graphics (SIGGRAPH)*, 2009.

[164] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017. 3

[165] Tuanfeng Wang, Tobias Ritschel, and Niloy Mitra. Joint material and illumination estimation from photo sets in the wild. In *International Conference on 3D Vision (3DV)*, pages 22–31, 2018. 13

[166] Yang Wang, Lei Zhang, Zicheng Liu, Gang Hua, Zhen Wen, Zhengyou Zhang, and Dimitris Samaras. Face relighting from a single image under arbitrary unknown lighting conditions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1968–1984, 2009. 17

[167] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *European Conference on Computer Vision*, pages 61–75. Springer, 2014. ix, 58, 60, 74, 75, 76, 77

[168] Robert J Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):191139–191139, 1980. 81

[169] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009. 55

[170] Lun Wu, Arvind Ganesh, Boxin Shi, Yasuyuki Matsushita, Yongtian Wang, and Yi Ma. Robust photometric stereo via low-rank matrix completion and recovery. In *Computer Vision–ACCV 2010*, pages 703–717. Springer, 2011. 50, 59, 83, 89

[171] Ying Xiong, Anandaroop Chakrabarti, Ronen Basri, Steven J Gortler, David W Jacobs, and Todd Zickler. From shading to local shape. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(1):67–79, 2015. 94

[172] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (TOG)*, 37(4):126, 2018.

[173] Hongyang Xue, Shengming Zhang, and Deng Cai. Depth image inpainting: Improving low rank matrix completion with low gradient regularization. *arXiv preprint arXiv:1604.05817*, 2016. 50

[174] Gao Yan. *Structured low rank matrix optimization problems: a penalty approach*. PhD thesis, 2010. 57

[175] Miaolong Yuan, Ishtiaq Rasool Khan, Farzam Farbiz, Susu Yao, Arthur Niswar, and Min-Hui Foo. A mixed reality virtual clothes try-on system. *IEEE Transactions on Multimedia*, 15(8):1958–1968, 2013. 3

[176] Alan Yuille and Daniel Snow. Shape and albedo from multiple images using integrability. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 158–164. IEEE, 1997. 81

[177] Stefanos Zafeiriou, Mark Hansen, Gary Atkinson, Vasileios Argyriou, Maria Petrou, Melvyn Smith, and Lyndon Smith. The photoface database. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 132–139. IEEE, 2011. viii, 18, 23, 25, 26

[178] Edward Zhang, Michael F Cohen, and Brian Curless. Emptying, refurnishing, and relighting indoor spaces. *ACM Transactions on Graphics (TOG)*, 35(6):174, 2016. 12

[179] Edward Zhang, Michael F Cohen, and Brian Curless. Discovering point lights with intensity distance fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6635–6643, 2018. 13

[180] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. ix, xi, 14, 33, 34, 40, 41, 44, 45, 122, 126

[181] Ying Zhang. Restricted low-rank approximation via admm. *arXiv preprint arXiv:1512.01748*, 2015. 57

[182] Zhengyou Zhang and Gang Xu. A general expression of the fundamental matrix for both perspective and affine cameras. In *Proceedings of the Fifteenth international joint conference on Artifical intelligence-Volume 2*, pages 1502–1507. Morgan Kaufmann Publishers Inc., 1997. 63

[183] Keke Zhao and Zhenyue Zhang. Successively alternate least square for low-rank matrix factorization with bounded missing data. *Computer Vision and Image Understanding*, 114(10):1084–1096, 2010. 53

[184] Hao Zhou, Jin Sun, Yaser Yacoob, and David W Jacobs. Label denoising adversarial network (LDAN) for inverse lighting of face images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 13, 19, 23, 27

[185] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017. 3

[186] Tinghui Zhou, Philipp Krahenbuhl, and Alexei A Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *International Conference on Computer Vision (ICCV)*, pages 3469–3477, 2015. 15, 34, 39, 43, 44

[187] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017. 14

[188] Wei Zhuo, Mathieu Salzmann, Xuming He, and Miaomiao Liu. Indoor scene structure analysis for single image depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 614–622, 2015. 13