Image Layer Separation and Application

by

Renjiao Yi

B.Sc., National University of Defense Technology, 2013

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Spring 2019

Approval

Name:	**Renjiao Yi**
Degree:	**Doctor of Philosophy (Computing Science)**
Title:	**Image Layer Separation and Application**
Examining Committee:	**Chair:** Kangkang Yin Associate Professor
	Ping Tan Senior Supervisor Associate Professor
	Hao (Richard) Zhang Supervisor Professor
	Yasutaka Furukawa Internal Examiner Assistant Professor
	Sing Bing Kang External Examiner Principal Researcher Microsoft Research
Date Defended:	**February 19, 2019**

Abstract

Image layer separation is an important step for image understanding and facilitates many image processing applications. It aims to separate a single image into multiple image layers, decomposing different components of the image. Image layers are either physics-based layers such as the reflectance layer in intrinsic image decomposition, or semantic layers such as the occlusion layer in image de-hazing, raindrop removal problems. Since the number of unknowns is at least twice that of the inputs, image layer separation problems are ill-posed and challenging. In order to solve such ill-posed problems, traditional methods acquire additional constraints based on prior knowledge, and recent deep learning methods rely on training data. In this thesis, we propose an optimization-based method based on handcrafted priors for video de-fencing (separating fence-like occlusion layers from dynamic videos), and an unsupervised deep learning training scheme for utilizing unlabeled real images from the Internet, which is applied on highlight separation and intrinsic image decomposition.

Traditional methods make assumptions based on observations and priors to acquire additional constraints and solve it as an optimization problem. In this thesis, we solve video de-fencing by a novel bottom-up pipeline based on such traditional optimization-based method. We present a fully automatic approach to detect and segment fence-like occluders from a video clip. Unlike previous approaches that usually assume either static scenes or cameras, our method is capable of handling both dynamic scenes and moving cameras.

After that, we introduce the main challenges of recent deep learning methods for image layer separation, which is the lack of real-world training data with ground truth. Thus, we propose an unsupervised training scheme for training the network on unlabeled real images. This unsupervised training scheme is then applied to two image layer separation problems, which are highlight separation for facial images trained from celebrity photos, and non-Lambertian intrinsic image decomposition trained from customer product photos.

Finally, we demonstrate one application from separated image layers, where we use faces as light probes to estimate the environment illumination. It is important for mixed reality applications, such as inserting virtual objects into real photos. Our technique estimates illumination at high precision in the form of a non-parametric environment map, and it works well for both indoor and outdoor scenes.

Keywords: image decomposition; highlight separation; intrinsic image decomposition; video de-fencing; illumination estimation

Acknowledgements

I would like to thank all the people for their help and support during my Ph.D. study. First and foremost, I would like to thank my senior supervisor Dr. Ping Tan. When I started my Ph.D. study, I was an amateur to computer vision. He taught me a lot from the details. It is an honor for me to have such a great supervisor, and I really appreciate and cherish the supervision and mentorship from him. I am also impressed by his enthusiasm for research, which I will always keep in mind.

I also own sincere thanks to Dr. Stephen Lin from Microsoft Research Asia for his mentorship during two projects for over two years. It is a great experience working with Steve, and I learned a lot from him, especially his rigorous attitude towards research. He also provided many exciting ideas, which is a great help for the progress of these projects. I am also grateful to Dr. Jue Wang from Adobe Research, for his mentorship on my first project.

I also want to express my sincere gratitude to the thesis examining committee members. I want to thank my supervisor Dr. Hao Zhang for his constructive suggestions during my Ph.D. study. I appreciate Dr. Yasutaka Furukawa for his insightful comments and discussions in my depth exam as well as serving as the internal examiner of my thesis. It is a great honor to have Dr. Sing Bing Kang as my external examiner, for his insightful comments and suggestions about my thesis, which inspired me a lot.

I want to thank my friends and colleagues from Gruvi Lab for their support and help: Zhaopeng Cui, Rui Huang and Xi Ao, Akshay Gadi Patil, Rui Ma, Honghua Li, Ruizhen Hu, Chengzhou Tang, Luwei Yang, Feitong Tan, Ibraheem Alhashim, Ali Mahdavi-Amiri, Zhiqin Chen, Kangxue Yin, Manyi Li *etc.*

Lastly, I want to thank my fiance Chenyang Zhu for his love and company for the past nine years, and he is always the best partner in life and work. I would also like to thank my parents, grandparents for their all-time love and support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Natural images capture mixes multiple components of the scenes into a single observed image, making the images difficult to understand. Those image components consist of scene appearance, lighting, shadows, occlusions and so on, where each of them only describes a property from one certain aspect. Separating these components is the key to understand and explain the complex virtual world. Image layer separation, or image decomposition, aims to separate a single image into multiple layers, where each layer only describes a single component. These layers can be physics-based layers such as the reflectance layer in intrinsic image decomposition which is related to the physical properties of the object surfaces, or semantic layers such as occlusion layers in image de-hazing, de-fencing or raindrop removal problems. Since image layer separation infers multiple outputs from one single input, they are highly ill-posed, as formulated in Section 2.1. To tackle them, additional information is provided by priors. Traditional methods usually use handcrafted priors based on observations while recent DNN-based ones learn priors automatically from training data by direct supervision. In this thesis, other than these two kinds of solutions, we also combine them where handcrafted priors are used to drive weakly-supervised or unsupervised training in DNNs, when ground truths are not available.

Traditional methods usually build objective functions in optimization based on observations and priors. For example, for the problem of intrinsic image decomposition, assumptions of piecewise constancy of surface colors [41], smoothness of diffuse [66, 103] or specular [60] reflection are enforced in the optimization. Some other methods acquire additional information by requiring extra inputs such as multiple input images or depth. In this thesis, we propose a novel bottom-up framework based on traditional optimization-based methods for the problem of separating fence-like occlusion layers from dynamic videos (video de-fencing), as illustrated in Figure 1.1.

In recent years, deep learning succeeds to solve many ill-posed problems, and it is natural that many recent approaches try to solve image layer separation through deep learning via direct supervision. However, supervised learning requires ground truths of separated image layers and a large scale of training data. Most approaches are trained from synthetic data

Figure 1.1: Illustration of video de-fencing.

via direct supervision, because of the infeasibility of collecting a large real-image dataset. However, it has become known that the mismatch between real and synthetic data may lead to a significant reduction in performance [100]. DNNs trained on synthetic dataset usually have problems on real testing data due to the domain shift between synthetic images and real images. Although domain adaptation aims to solve this problem, it requires the synthetic dataset is large and diverse enough to make sure a very good performance on synthetic images, and such datasets are still expensive to generate. It motivates recent works [58, 65] on developing unsupervised schemes for DNN training on unlabeled real data, where image sequences of a fixed scene under changing illumination are used to enforce constraints such as reflectance consistency in intrinsic image decomposition.

Although such time-lapse image sequences are perfectly aligned so that consistency is easily utilized, they are not easy to get online. On the Internet, there exists an untapped wealth of unconstrained images from random viewpoints which are countless online and easy to collect. In this thesis, we propose an unsupervised training scheme and a low-rank loss for such unconstrained real data collected from the Internet. In detail, because an object's appearance should be consistent, the diffuse chromaticities of aligned images under different illuminations should be ideally rank one, and this property is measured by a low-rank loss. As illustrated in Figure 1.2, we demonstrate the proposed training scheme on face images and it shows the state-of-the-art performance on the task of highlight removal. Furthermore, with further modifications, we apply the proposed unsupervised training scheme on multiview images of general objects for factorizing a single image into highlight, shading and reflectance/albedo layers, as illustrated in Figure 1.3. Due to the fact that misalignments cannot be avoided while aligning these unconstrained multi-view images even by the state-of-the-art algorithms [32, 88], we further improve the low-rank loss. The modified low-rank loss is robust for local misalignment, which is also useful for many other problems.

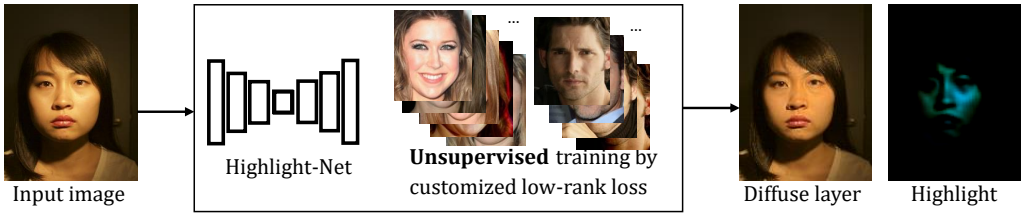


Figure 1.2: Unsupervised deep highlight extraction for face images.

$$Input = Diffuse + Highlight = Albedo \cdot Shading + Highlight$$

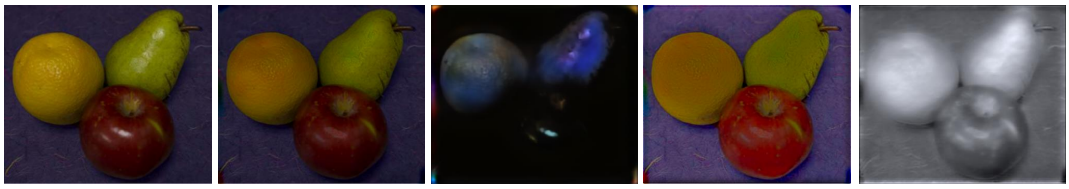


Figure 1.3: Non-Lambertian intrinsic image decomposition. Following the image formation model on the top, a single input image is separated into four image layers.

There are many applications based on image layer separation. In this thesis, we present an application of illumination estimation from the proposed face highlight removal method. Inserting virtual accessories to selfies become easy to do with mobile augmented reality (AR) apps like Snapchat [101]. While the entertainment value of mobile AR is evident, current results are usually far from realistic because the inserted virtual object is not rendered under the same illumination as in the image scene. For high photorealism in AR, it is thus necessary to estimate the illumination from the image, so that realistic virtual object insertion can be achieved. Illumination estimation from a single image is challenging because lighting is intertwined with geometry and reflectance in the appearance of the scene. Since faces are a common occurrence in photos, together with existed techniques of face geometry reconstructions, we proposed a method to use faces as light probes for estimating the environmental illumination. The estimated illuminations can be used to render visual objects into the photos realistically, as illustrated in Figure 1.4.

In summary, in this thesis, we address three image layer separation problems based on the proposed novel optimization-based framework and unsupervised training scheme respectively. At last, we also present one application which is illumination estimation from separated image layers.

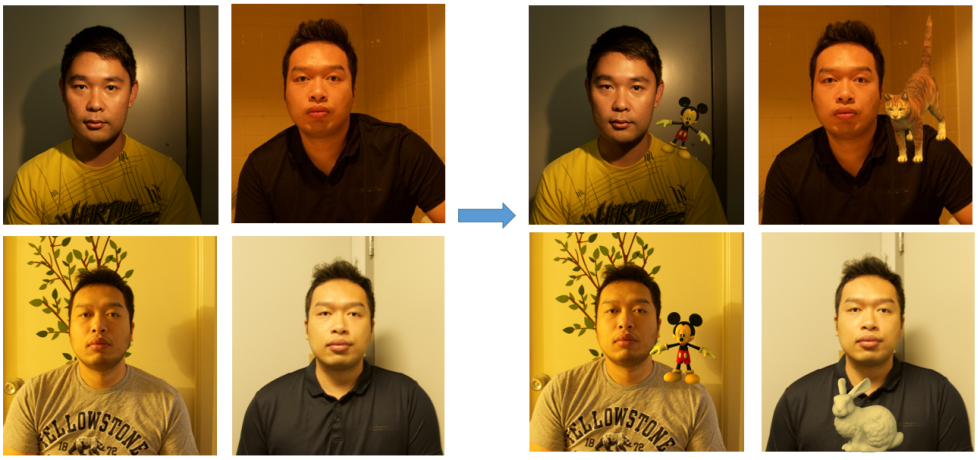


Figure 1.4: Realistic augmented/mixed reality achieved by using faces as light probes.

1.1 Challenges

Image layer separation is challenging as it tries to infer at least two unknown layers L_1 and L_2 from one single image or video frame I. Previous methods make assumptions based on observations or prior knowledge to provide additional constraints. However, these assumptions only work for certain situations, and some priors may not exist in some situations. Proposing a method that works for most scenarios is very challenging. Focusing on the three image layer separation problems we aim to solve in this thesis, we first discuss the main challenges of each problem.

For video de-fencing (removing fence-like occlusions from videos), there are several main challenges. Firstly, the fences are thin and long structures, which is hard to segment even for interactive segmentation tools like GrabCut [90] or Rotobrush [2]. Secondly, there are usually no distinctive colors or strong textures on a fence, making it hard to track, and their repetitive structure patterns often lead to errors in tracking and motion estimation. Thirdly, although some works [108, 114] successfully remove fences from videos, they can only deal with static scenes. For videos of dynamic scenes, the commonly used two-layer motion model breaks down due to the existence of large dynamic objects, which would cause discontinuities in background motion.

For face highlight removal, previous methods try to solve it by assumptions such as white illumination [104, 105, 117], dark channel priors [38] and repeated textures [102], which do not exist for face images. Furthermore, most previous methods do not consider the saturation of pixel intensities. However, most of the images captured by mobile phones are LDR (low dynamic range) and may have the saturation on highlight regions while lighting intensity is strong. For general objects, the problem is also difficult without any constraints of illumination colors or surface colors, which are not accessible for images under natural scenes. For deep learning, the lack of training data is the main obstacle.

Figure 1.5: Examples of images from the Internet. They are difficult to align due to their various backgrounds, illuminations, and views.

For intrinsic image decomposition, previous methods assuming Lambertian surfaces would fail when highlight exists [4,65], and CNN methods trained on synthetic data cannot generalize well for real images [71, 99] due to the domain shift. Since the dataset with ground truth is unavailable for real images, the main challenge is how to utilize real images without ground truth to train deep neural networks. On the other hand, in order to deal with non-Lambertian surfaces, the highlight layer should be considered and solved together. Collecting training data for this task is also challenging. Li *et al.* [58] proposed to use time-lapse video collected on the Internet for unsupervised training, but these scene videos do not contain many glossy surfaces, so the trained network cannot work well for non-Lambertian scenes.

Furthermore, unconstrained images from the Internet are very noisy and bring difficulties to image alignment. As shown in Figure 1.5, images from random viewpoints, captured by unknown cameras, under various illumination and backgrounds are difficult to be aligned well. Most loss functions in deep learning are defined based on pixel-to-pixel correspondences, which will suffer from the misalignment of training images and have poor performance.

1.2 Contributions

In this thesis, we propose methods for three image layer separation problems and summarize all image layer separation problems into several categories, for which we discuss and provide possible solutions. Firstly, we propose an optimization-based method to solve the fence-like occlusion separation in dynamic videos, which is very challenging for previous approaches [35, 69, 108, 114]. Then we propose an unsupervised training scheme for deep learning methods, and it is applied to two image layer separation problems, which are highlight separation and non-Lambertian intrinsic image decomposition. At last, we also present an application of illumination estimation from the separated image layers, which achieves realistic mixed reality using faces as light probes via unsupervised deep highlight extraction.

Fence segmentation of dynamic videos. We present a fully automatic approach to detect and segment fence-like occluders from a video clip. Unlike previous approaches that usually assume either static scenes or cameras [35,69,108,114], our method is capable of handling both dynamic scenes and moving cameras. Under a bottom-up framework,

it first clusters pixels into coherent groups using color and motion features. These pixel groups are then analyzed in a fully connected graph and labeled as either fence or non-fence using graph-cut optimization. Finally, we solve a dense Conditional Random Filed (CRF) constructed from multiple frames to enhance both spatial accuracy and temporal coherence of the segmentation. Once segmented, one can use existing hole-filling methods to generate a fence-free output. This work has been reported in [119].

Unsupervised training scheme on unconstrained real data for highlight separation and intrinsic image decomposition. We propose an unsupervised training scheme on unconstrained real data for training deep neural networks of highlight layer separation. Since real training data for highlight extraction is very limited, we introduce an unsupervised scheme for finetuning the network on real images, based on the consistent diffuse chromaticity of a given face seen in multiple real images. The network is trained on MS-celeb-1M database [27], which contains 100 images for each of 100,000 celebrities. For each celebrity, since his/her facial appearance is consistent, the diffuse chromaticities of aligned facial images under different illuminations should be consistent as well (they should ideally be rank one), and this property is measured by the proposed low-rank loss to drive the unsupervised training. This work has been reported in [120]. After that, we also apply the unsupervised training scheme to non-Lambertian intrinsic image decomposition on general objects. We present an unsupervised approach for factorizing object appearance into the highlight, shading, and albedo layers. In contrast to previous unsupervised learning techniques [58, 65] for reflection separation, which are trained on fixed-view time-lapse image sequences, our method can be trained on multiview image sets such as customer product photos, which are numerous online, facilitate object-level decomposition, and exhibit large illumination variations that make them suitable for training of reflectance separation. The central element of our approach is a proposed image representation based on local color distributions that allows training to be relatively insensitive to misalignment of multi-view images. In detail, we re-rank the pixels in each local grid by their intensities, and re-correspond pixels in roughly aligned images based on the re-ranking. In addition, we present a new guidance cue for unsupervised training that exploits the synergy between highlight separation and intrinsic image decomposition.

Faces as lighting probes via unsupervised deep highlight extraction. We present an application of highlight layer separation for estimating detailed scene illumination using human faces in a single image. In contrast to previous works that estimate lighting in terms of low-order basis functions [4, 24, 36, 42, 54, 84, 86] or distant point lights [57, 76, 91, 92, 109], our technique estimates illumination at a higher precision in the form of a non-parametric environment map. We train a deep neural network for highlight separation and then trace these reflections back to the scene to acquire the environment map. In tracing the estimated highlights to the environment, we reduce the blurring effect of skin reflectance

on reflected light through a deconvolution determined by prior knowledge on face material properties. This work has been reported in [120].

1.3 Thesis organization

This thesis is organized in the following way: in Chapter 2, we survey previous techniques on image layer separation, focusing on three specific problems discussed in this thesis, which are fence layer separation, highlight layer separation, and intrinsic image decomposition. We also survey related works about illumination estimation, as an application of image layer separation. Then we present the proposed video de-fencing method based on traditional optimization-based methods in Chapter 3. After that, we introduce our proposed unsupervised training scheme for deep learning methods in Chapter 4, and it is then applied to face highlight separation in Chapter 5 and non-Lambertian intrinsic image decomposition of general objects in Chapter 6. In Chapter 7, we introduce the illumination estimation method based on the face highlight separation. Finally, Chapter 8 concludes this thesis and discusses limitations and potential future works.

Chapter 2

Background and Related Works

In this chapter, we firstly introduce the image formation model of image layer separation, and related works of three specific problems focused in this thesis, which are image/video de-fencing, highlight separation and intrinsic image decomposition. At last, we also introduce related works of illumination estimation, which is demonstrated as an application of our proposed face highlight separation method.

2.1 Image formation models

Image layer separation, or image decomposition, aims to separate one single input image into multiple image layers. The image layers are different depending on the applications. For example, in image de-fencing, the image layers are the background scene layer and the fence layer, and in intrinsic image decomposition, the image layers are the reflectance layer and the shading layer.

The general form of layer separation from a single-image can be written as:

$$I = L_1 + L_2, \tag{2.1}$$

where I is the observed image, and L_1 and L_2 are separated image layers, usually one single image is separated into two layers at a time.

For each pixel p, since the layers can be semi-transparent, the observed intensity can be written as the sum of two image layers:

$$I(p) = L_1(p) + L_2(p), \tag{2.2}$$

For some problems, image layers are non-transparent and opaque in color, and each pixel p belongs to either L_1 or L_2. $P(L_1)$ is the collection of pixels belonging to L_1, and $P(L_2)$ is the collection of pixels belonging to L_2. Thus at pixel p, the observed intensity can be written as:

$$I(p) = \begin{cases} L_1(p), & if\ p \in P(L_1), \\ L_2(p), & if\ p \in P(L_2). \end{cases} \tag{2.3}$$

For image/video de-fencing, given a single image or a video sequence, certain regions are occluded by fence-like obstructions. We aim to detect and separate the occlusion layer automatically, to achieve better video quality.

This problem can be formulated as:

$$I = L_{Bg} + L_{Fence} \tag{2.4}$$

where I is the observed image, and L_{Bg} is the background layer and L_{Fence} is the fence-like occlusion layer. Since both layers are opaque in color, where each pixel in the input image either belongs to fence layer or the background layer as in Equation 2.3. Separating opaque layers are simpler than separating semi-transparent layers, and can be solved as a segmentation problem, where every pixel is labelled as background or occlusions.

After separating the occlusion layer, we can remove it from the input images or videos, and fill in the occluded regions by in-painting methods.

For highlight separation, given a image of a non-Lambertian scene, as illustrated in Equation 2.5, we aim to separate the diffuse layer I_d and the specular highlight layer H.

$$I = I_d + H \tag{2.5}$$

For intrinsic image decomposition, given a observed image, as illustrated in Equation 2.6, we aim to decompose the reflectance layer R and the shading layer S.

$$I = A \cdot S \tag{2.6}$$

This can be reformulated as the form of Equation 2.1 in the log-domain:

$$log(I) = log(A) + log(S) \tag{2.7}$$

Intrinsic image decomposition often assumes a Lambertian scene, while most natural scenes contain many glossy surfaces. Thus, combining highlight separation and intrinsic image decomposition, we can solve a non-Lambertian intrinsic image decomposition where the highlight layer H, the reflectance layer R and the shading layer S can be solved altoghther as illustated in Equation 2.8, and the joint optimization can also improve the results of both problems.

$$I = A \cdot S + H \tag{2.8}$$

Other than the above problems solved in this thesis, there are many other image layer separation problems which we do not solve in this thesis but they are categorized and discussed in Chapter 8, where possible solutions are provided for each category.

2.2 Image layer separation

All image layer separation problems aim to infer multiple outputs from a single input image, which make them highly ill-posed. Additional information needs to be added in order to tackle these problems. Traditional methods use handcrafted priors based on observations, such as repetitive patterns of fences [29, 61, 79] in image de-fencing, dark channel prior in image de-hazing [30] and smoothness in reflectance [46, 47, 68] in intrinsic image decomposition.

However, these handcrafted priors are difficult to define and they do not work for all scenarios. Recent data-driven methods use DNNs to learn priors from synthetic data [33,71,99] or labeled real-world data [18, 83] automatically. When labeled training data is infeasible to capture, and rendering synthetic data is expensive, unsupervised or weakly-supervised training methods are useful for many problems. Recent methods [58, 65] propose unsupervised training on fixed-view image sequences, where the training losses are defined by handcrafted priors as in traditional methods, such as the reflectance consistency in intrinsic image decomposition. Focusing on the three image layer separation problems solved in this thesis, we review recent works of each of them.

2.2.1 Image and video de-fencing

Image de-fencing is the problem of separating the fence-like occlusion layer and the background scene layer from a single image. The imaging model is illustrated in Figure 2.1, the occlusion layer is often between the camera and the background scene layer, so the occlusions cannot be avoided by simply translating or rotating the camera.

Hays *et al.* [29] and Liu *et al.* [61] detect fence structures from a single image by extracting near regular repetitive texture patterns. Park *et al.* [79] enhance the repetitive structure detection to deal with deformations due to perspective camera projection and non-planar underlying shapes. Online learning and classification are adopted to further enhance the detection [78]. Generally speaking, these methods rely on the success of the challenging task of repetitive structure detection, which is difficult to handle certain types of fence structures such as window blinds and tree branches.

Fence detection and removal can be easier when multiple input images or a video clip is available. Yamashita *et al.* [115] use flash and non-flash images together with multi-focus images to detect and remove fence. Khasare *et al.* [37] manually label fence pixels with existing interactive segmentation tools. Mu *et al.* [69] detect and remove fence using parallax cues from video clips under the assumption of a *static* scene. Xue *et al.* [114] separate fence

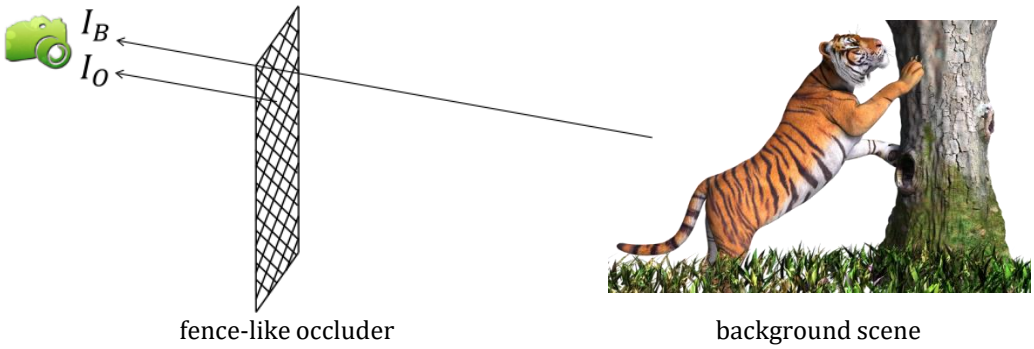


Figure 2.1: The image formation model of capturing a image through fence-like occlusions [114].

from the background using motion cues through an optimization process. This approach achieves high quality results, but is limited to static scenes.

From another aspect, some works [35,108] attempt to solve the fence separation by multi-view stereo or structure-from-motion. After dense stereo is reconstructed, multiple layers of the images can be separated by thresholding depth. However, it is hard to automatically decide the proper depth threshold and depth provided by the reconstructed stereo can be noisy. Thus the problem is usually defined as a segmentation problem by enforcing pixels in each layer having small color variance and pixels in multi-view images having the same assignments. Once fence layer is separated, the originally images can be completed by rendering occluded pixels back to each view by median values in multi-view images. Recent works [18, 35] also apply deep neural networks to detect fence joints or estimating the disparity to facilitate the segmentation.

Image inpainting [7] [16] [6] techniques can fill in small image regions given their masks. Video inpainting [72] [112] can recover missing structures on the current frame by transferring pixels from neighboring frames. The success of these methods rely on accurate segmentation masks as input, which are hard to achieve for fences even with advanced interactive segmentation tools [2, 56, 90]. Our segmentation approach provides such masks automatically.

2.2.2 Highlight layer separation

Highlight layer separation involves separating the diffuse and specular reflection components in an image. It is an ill-posed problem that has been made tractable through the use of different priors. Among them are priors on piecewise constancy of surface colors [41], chromatic information [38, 104–106, 117], smoothness of diffuse [66, 103] or specular [60] reflection, diffuse texture coherence [102], low diffuse intensity in a color channel [38], sparsity of highlights [1, 26, 60], and a low-rank representation of diffuse reflection [26]. These tech-

niques are limited in the types of surface textures that can be handled, and they assume that the illumination color is uniform or known. In recent work [53], these restrictions are avoided for the case of human faces by utilizing additional constraints derived from physical and statistical face priors.

Instead of crafting priors for highlight extraction by hand, they can be learned in a statistical fashion from images using neural networks. This was first investigated together with intrinsic image decomposition through supervised learning on a large collection of rendered images [99]. An unsupervised approach was later presented for the case of human faces in our recent work [120], where a set of images of the same face is aligned using detected facial landmark points, and training guidance is provided by a low-rank constraint on diffuse chromaticity across the aligned image. In Chapter 6, we also present an unsupervised learning approach but deal with image sets of general objects which are difficult to align accurately. Since misaligned images violate the low-rank property assumed in [120], we propose a technique that is relatively robust to such misalignments, thus enabling unsupervised training over a much broader range of objects.

2.2.3 Intrinsic image decomposition

Recent works are mostly built on deep learning frameworks. Due to there is no large-scale real image dataset with ground truth, supervised methods [33, 71, 99] are trained on synthetic data, by re-synthesizing animation movie sequences like the MPI Sintel dataset [12], or synthesing their own dataset [33, 99] by 3D models from ShapeNet [14]. However these networks trained by synthetic data cannot generalize well on real scenes, so weakly supervised methods [44, 70, 123] are proposed to train on real images where sparse annotations are available, such as the IIW dataset (Intrinsic Image in the Wild). Unsupervised methods [58] are also proposed recently, and they are trained on fixed-position, time-lapse videos. While some others [65] proposed to train on small-scale real image dataset such as MIT intrinsics dataset [25]. For these unsupervised methods, the loss function is mostly based on reflectance consistency in multiple images, as well as shading smoothness.

Previous to the deep-learning approaches of recent years, intrinsic image decomposition was primarily addressed as an optimization problem constrained by various prior assumptions about natural scenes. These priors have been used to classify image derivatives as either albedo or shading change [9, 22, 40, 50, 107], to prescribe texture coherence [97, 122], and to enforce sparsity in the set of albedos [89, 98]. Decomposition constraints have also been derived using additional input data such as image sequences [46, 68, 111, 118], where temporal coherence can be enforced for multiple images from the same view [118], or multi-view stereo is used to reconstruct the whole scene for multi-view images [47], depth measurements [3, 15, 34, 51], where they can enforce pixels having similar normals to have similar shading intensities, and user input [9, 10, 96].

These earlier methods have been surpassed in performance by deep neural networks which learn statistical priors from training data. Some of these networks are trained with direct supervision, in which the ground-truth albedo and shading components are provided for each training image [39,71,99]. To obtain ground truth at a large scale for training deep networks, these methods utilize synthetic renderings, which can lead to poor generalization of the networks to real-world scenes. This issue is avoided in several methods by training on sparse annotations of relative reflectance intensity [5] or relative shading [44] in real images [21, 44, 70, 123]. However, these manual labels provide only weak supervision, and the need for supervision reduces the scalability of the training data.

Most recently, unsupervised methods have been presented in which the training is performed on image sequences taken from fixed-position, time-lapse video with varying illumination [58, 65]. In these networks, a major source of guidance for unsupervised training is the temporal consistency of reflectance for static regions within a sequence. The networks are configured so that they can be applied to just a single input image at inference time.

We note that multiview images have previously been used for intrinsic image decomposition of outdoor scenes [19]. The decomposition is solved by an inverse rendering approach, where shading is inferred from an approximate multiview stereo reconstruction and an illumination environment estimated given the known sun direction. The multiview images are required to be taken under the same lighting conditions. By contrast, in Chapter 6, we address a problem where no knowledge about the illumination is given, the lighting can differ from image to image, and differences in image backgrounds would be disruptive to multiview stereo.

2.3 Illumination estimation

Illumination estimation from a single image is difficult, many previous methods assume known geometry and estimate illumination from shading [57, 81, 84, 109] or shadows [57, 74, 76, 91, 92]. Some methods [4, 63, 73, 85] infer the geometry, BRDF, and illumination jointly, or by fitting a model for a specific kind of objects such as human faces [24, 36, 42, 54, 86].

For the representation model of illuminations, although an illumination environment can be arbitrarily complex, nearly all previous works employ a simplified parametric representation as a practical approximation. Earlier techniques mainly estimate a set of point lights [57,76,91,92,109]. More recently, low-order spherical harmonics [4,24,36,42,54,84,86] or Haar wavelets [74] are also used to represent denser illuminations while keeping a small number of parameters. The relatively small number of parameters in these models simplifies optimization but provides limited precision in the estimated lighting. Greater precision has been obtained by utilizing lighting models specific to a certain type of scene. For outdoor scenes, a sky and sun model are proposed and can be used for accurate recovery of outdoor illuminations [13, 31, 48, 49]. For indoor scenes, a CNN-based method [23] is proposed

to infer environment illumination from a image with a limited front-of-view, training from a large-scale HDR dataset with ground truth, as well as manually labeled light sources locations. Highlight reflections have been used together with diffuse shading to jointly estimate non-parametric lighting and an object's reflectance distribution function [62]. In that work, priors on real-world reflectance and illumination are utilized as constraints to improve inference in an optimization-based approach. The method employs an object with known geometry, uniform color, and a shiny surface as a probe for the illumination.

Chapter 3

Traditional Optimization-based Methods for Video De-fencing

Traditional methods of image layer separation problems are usually based on optimization defined on observations or priors, due to the properties of ill-posed problems. In this chapter, we present a novel bottom-up framework to solve video de-fencing of dynamic scenes by handcrafted priors.

3.1 Introduction

It is a common case that one has to shoot an interesting scene through fences or wires. For instance, capturing a video of a walking tiger behind an enclosing fence in a zoo, or a building through wires or tree branches. Such videos are usually unpleasant to watch due to the strong distraction caused by the occluders. A common photography trick to alleviate this problem is to adjust the focus length and aperture of the camera to make the fence out-of-focus, thus less distracting when watching the video. However its effectiveness is limited and is only applicable to relatively advanced cameras, excluding most mobile phone cameras. Removing fence from videos at the postprocessing stage is thus highly desirable.

Despite a few recent attempts [37, 69, 115], removing fence from videos with unconstrained scene dynamics and camera movement is largely an open problem. In particular, it is hard to automatically detect and segment fence in videos. Fences contain very thin structures, which are difficult to segment even for interactive tools such as GrabCut [90] or Rotobrush [2]. Furthermore, there is usually no distinctive colors or strong textures on a fence, making it hard to track, Their repetitive structure patterns often lead to tracking and motion estimation errors. A recent work [114] successfully removes fence from videos, but only for static scenes. For videos capturing dynamic scenes, the commonly used two-layer motion model breaks out due to the existence of large dynamic objects, rendering methods that rely on static scene reconstruction insufficient, as we will show in the experimental section.

In this thesis, we present a new method for automatic wire fence segmentation from casual videos capturing dynamic scenes or objects. Users can capture the input videos using a handhold camera, and the camera is preferred to be moved in a circular motion in a plane which is roughly parallel to the fences. By allowing dynamic scenes, our approach has a much wider application range than previous work that are constrained to static ones. Our approach can also deal with videos shot with a moving camera, which is quite common for novice users capturing with hand-held mobile devices. We show that, while introducing object and camera motion brings new challenges to the task, they in turn provide additional information that can facilitate fence detection and segmentation. Specifically, the camera motion gives the fence a rigid motion in the video that is usually quite distinctive from the object motion behind it, allowing better segmentation using local motion contrast.

Our method takes a bottom-up approach. It begins by computing optical flow between neighboring frames, and grouping pixels in each frame according to color and motion. In the first round, we treat each group as a super-pixel and consider labeling each one as either fence or non-fence. Each group's probability of being fence is evaluated according to its structural and appearance features. The compatibility between two neighboring groups are computed from their color, motion, and structural similarities. We then solve a graph-cut optimization to produce initial labeling. The initial labeling, done on a per-frame basis, suffers from imprecise fence localization and poor temporal coherence. It is further refined by a spatio-temporal dense Conditional Random Field (CRF) optimization [45], which improves fence segmentation in both spatial accuracy and temporal coherence.

We evaluate the proposed approach on various videos, including mobile phone videos captured by ourselves, and Youtube video clips with completely unknown camera setting. Our segmentation results are quantitatively evaluated on a new dataset with manually labeled ground truth. The results show that our method achieves much better precision and recall than previous approaches. Finally, we demonstrate simple hole-filling with existing inpainting techniques [16] to remove detected fences.

3.2 Fence segmentation

Our fence segmentation includes three major steps. Firstly, pixels in each frame are clustered into a fixed number of groups based on color and motion information. Secondly, each of these groups is labeled as fence or non-fence by a graph-cut optimization applied to each video frame individually. Finally, a dense condition random field (CRF) is optimized over all frames simultaneously to label each pixel as fence or non-fence to improve the temporal coherence and spatial accuracy of fence segmentation. As an example, the fence segmentation results after per-frame graph-cut and multi-frame CRF is shown in Figure 3.1 (b) and (c) respectively, where the input frame is in Figure 3.1 (a).

(a) (b) (c)

Figure 3.1: (a) One frame in input video; (b) initial fence segmentation by graph-cut; (c) final fence segmentation by dense CRF.

3.2.1 Pixel grouping

Fences have distinctive structural features, *e.g.* they typically (but not necessarily) have two sets of thin wires pointing at two nearly perpendicular directions. This inspires us to form pixel groups to exploit spatial structural features for fence detection. We apply K-means clustering to pixels at each frame according to color and motion information. This clustering is based on the observation that fences pixels often have similar colors, and distinctive motion from the background due to their short distances to the camera. Even in dynamic scenes, the moving objects in background tend to have quite different motion from the fence.

We apply the optical flow algorithm in [59] to compute local motion between neighboring frames. One example of computed flow field is showed in Figure 3.2 (b). The flow vectors in each frame are normalized by subtracting the minimum value and then divided by their value range (*i.e.* the difference between the maximum and minimum values). For each pixel, we concatenate its RGB color (in $[0, 1]$) and the normalized flow vector to form a 5D feature. K-means is applied to generate 50 groups for each frame: examples are shown in Figure 3.2 (c) - (f). Typically, fence pixels and background pixels are separated into different groups due to their difference in either color or motion. In the following, we seek to identify fence pixel groups according to fence structural features.

3.2.2 Initial fence segmentation

On each frame, we form a fully-connected graph where each pixel group is a vertex. We optimize a fence or non-fence label at each vertex by graph-cut, which minimizes the following objective function:

$$E = \sum_i D(c_i, l_i) + \sum_{(i,j)} S(c_i, l_i; c_j, l_j). \tag{3.1}$$

Here, c_i, c_j indicates the i-th and j-th pixel group, l_i, l_j are the binary fence labels on c_i, c_j respectively. The data term $D(\cdot)$ measures the probability of a pixel group being fence, define as:

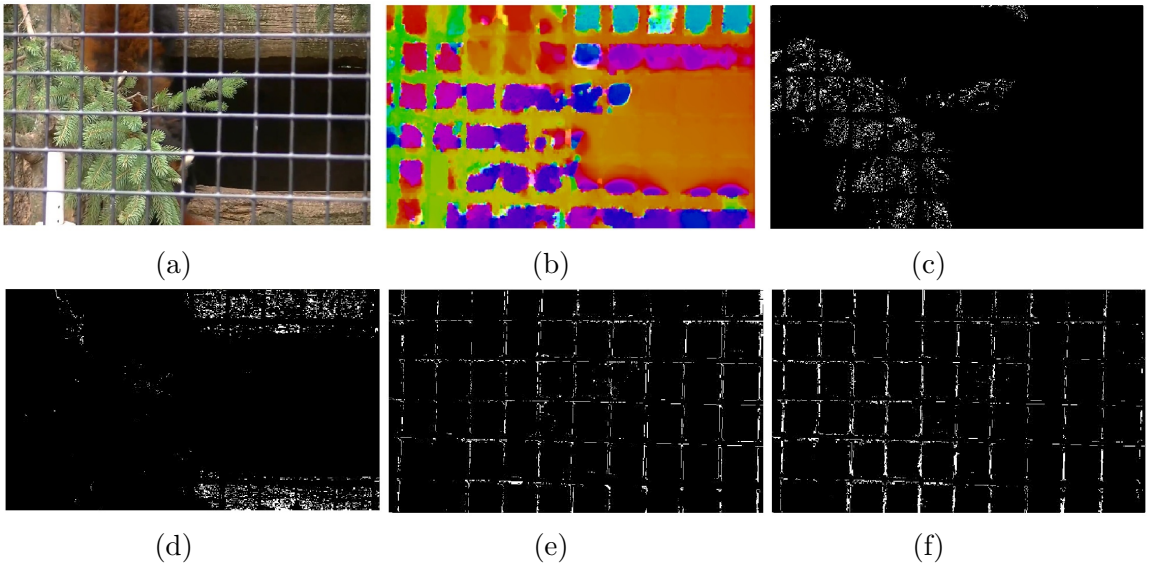


Figure 3.2: (a) input frame; (b) optical flow field; (c)–(f) some representative pixel groups. Note that fence and background pixels are largely separated into different groups due to color and/or motion difference.

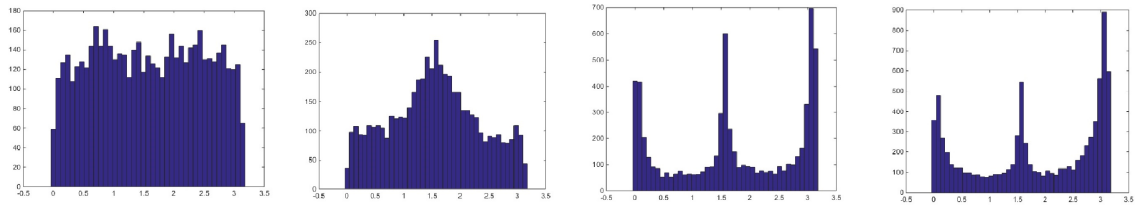
$$D(c_i, l_i) = l_i \cdot (1 - P(c_i)) + (1 - l_i) \cdot P(c_i), \tag{3.2}$$

where $P(c_i)$ is the probability that c_i being fence. It includes a gradient-based term and a geometry-based term:

$$P(c_i) = (1 - D_1(c_i)) \cdot (1 - D_2(c_i)). \tag{3.3}$$

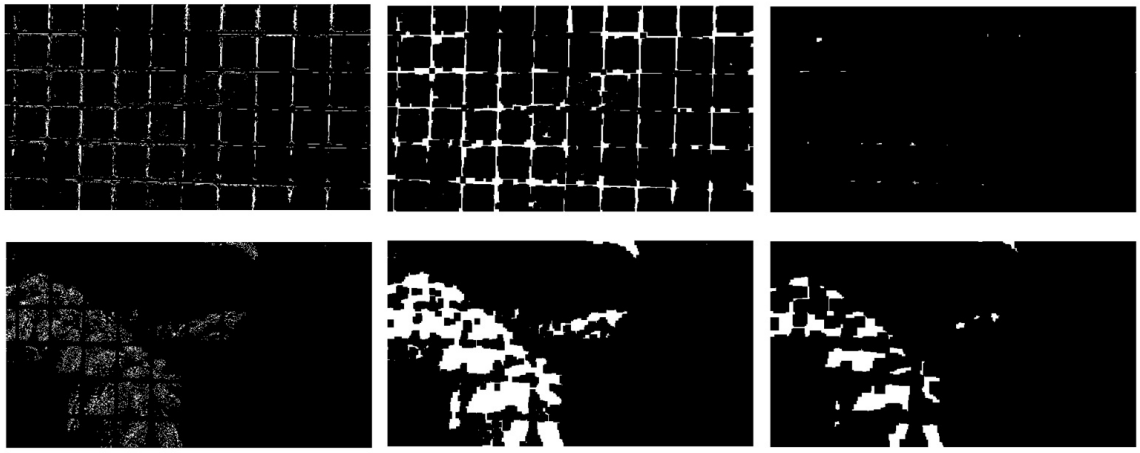
The gradient-based term $D_1(\cdot)$ exploits the fact that fences typically contain two sets of nearly perpendicular wires. We build a gradient orientation histogram for all pixels in a group. The histogram of a fence group should have two dominant peaks in two nearly perpendicular orientations. In contrast, a non-fence group tends to have a flat histogram. Some example are showed in Figure 3.3, where Figure 3.3 (a), (b) and (c), (d) are histograms of non-fence and fence groups, respectively. Their corresponding pixel groups are shown in Figure 3.2 (c), (d) and (e), (f), respectively. To exploit this observation, for each histogram, we firstly search the global highest peak c, and then search another local peak in an interval centered at $c+\pi/2$ with width $\pi/5$. We then take the histogram value at the middle point of these two peaks. For fence groups, this middle point is often associated with a low histogram value, in the valley between two histogram peaks in Figure 3.3 (c)-(d). D_1 is computed as the ratio of the histogram value at the middle point over that at the two peaks. Sometimes, the occluder contains multiple wires of similar orientations, which leads to a single dominant peak in the gradient orientation histogram. Our definition of $D_1(\cdot)$ can deal with such cases.

The geometry-based term D_2 exploits the fact that fences are usually thin structures. A morphological erosion should remove most of pixels in a fence group. In contrast, a non-



(a) (b) (c) (d)

Figure 3.3: (a)–(b) gradient orientation histograms of two background clusters (see Figure 3.2 (c) and (d)); (c)–(d) gradient orientation histograms of two fence clusters (see Figure 3.2 (e) and (f)).



(a) (b) (c)

Figure 3.4: A fence and non-fence pixel group after (a) initial K-means grouping, (b) 'close operator', and (c) erosion.

fence group usually has many more remaining pixels after this operation. To be robust to noisy grouping results, we first apply a 'close operator' to connect nearby isolated pixels. Figure 3.4 (a), (b), and (c) show results for a fence and non-fence group by initial K-means grouping, 'close operator', and erosion respectively, where morphological masks are 10×10. D_2 is computed as the percentage of pixels remained after the erosion. Both D_1 and D_2 are then linearly normalized to $[0, 1]$.

The smoothness term $S(\cdot;\cdot)$ in Equation 3.1 measures similarities between pixel groups based on their color, gradients orientation histogram, and dominant gradient orientations (the two histogram peaks selected when evaluating D_1). It is defined as:

$$S(c_i, l_i; c_j, l_j) = \mu(l_i, l_j) \cdot (1 - S_1(c_i, c_j)) \cdot (1 - S_2(c_i, c_j)) \cdot (1 - S_3(c_i, c_j)). \tag{3.4}$$

Figure 3.5: Initial segmentation by graph-cut optimization.

Figure 3.6: Fence segmentation by the multi-frame dense CRF optimization on same frames in Figure 3.5.

Here, $\mu(l_i, l_j)$ is the Pott model: 1 when $l_i \neq l_j$, and 0 otherwise. S_1 is the L_1 color histogram distance of two groups, computed in ab channels only in Lab space in order to be roust to illumination variations. S_2 is the L_1 distance between two gradient orientation histograms. S_3 is the difference of dominant gradient orientations. Suppose $g_1(\cdot), g_2(\cdot)$ are the two dominant gradient orientations of a pixel group, we measure S_3 as:

$$\min(dis_{g_1}, \pi - dis_{g_1}) + \min(dis_{g_2}, \pi - dis_{g_2}).$$

Here, $\min(dis_{g_1}, \pi - dis_{g_1})$ and $\min(dis_{g_2}, \pi - dis_{g_2})$ compute the closest peak in c_j to the first and second peaks in c_i respectively. Specifically, we compute them as the following:

$$dis_{g_1} = \min\left\{|g_1(c_i) - g_1(c_j)|, |g_1(c_i) - g_2(c_j)|\right\}, \tag{3.5}$$

$$dis_{g_2} = \min\left\{|g_2(c_i) - g_1(c_j)|, |g_2(c_i) - g_2(c_j)|\right\}. \tag{3.6}$$

S_1, S_2, and S_3 are all linearly normalized to be in $[0, 1]$.

We use graph-cut [11] to solve for a fence or non-fence label at each group. Some results are showed in Figure 3.5. Note that fence segmentation at this stage is roughly correct but inaccurate, fence boundaries do not align well with image edges. There are also occasional frames with poor segmentation results. This is because K-means clustering fails to produce correct low-level clustering results for frames with very little motion. Next, we build a dense CRF over all video frames to further improve the segmentation result.

3.2.3 Spatio-temporal segmentation refinement

In our dense CRF, each pixel on each frame is a vertex, and it connects to all other vertices. This spatio-temporal graph construction gives us a chance to enhance both temporal coherence and spatial accuracy of the segmentation. The total energy is defined in the same

Figure 3.7: Rotobrush [2] results on two examples. From left to right: manually-labeled keyframe; results after propagating 5 frames; 10 frames; 15 frames and 20 frames. The segmentation results deteriorate quickly in the temporal propagation process.

way as Equation 3.1, with data and smoothness terms defined differently. The data term is defined as:

$$\mathbb{D}(x, l_x) = l_x \cdot (1 - \mathbb{P}(x)) + (1 - l_x) \cdot \mathbb{P}(x).$$

where $\mathbb{P}(x)$ is evaluated as:

$$\mathbb{P}(x) = \mathbb{P}_1(x) \cdot \mathbb{P}_2(x). \tag{3.7}$$

Here, the term $\mathbb{P}_1(x)$ encourages the result from CRF optimization to be consistent with the initial graph-cut labeling result, defined as:

$$\mathbb{P}_1(x) = \begin{cases} 1 - \alpha, & L_0(x) = 0 \\ \alpha, & L_0(x) = 1 \end{cases} \tag{3.8}$$

where α is a parameter determining the confidence of initial graph-cut segmentation. In our system we simply use a constant probability at 0.8, although one could further make it adaptive according to the features of each pixel group. $L_0(x)$ is the initial label of pixel x, which is 1 or 0 for fence and non-fence pixels, respectively. The term $\mathbb{P}_2(x)$ is defined as

$$\mathbb{P}_2(x) = P(c_i), \qquad x \in c_i. \tag{3.9}$$

Here, $P(c_i)$ is the probability evaluated in Equation 3.3 in the previous step. $x \in c_i$ means that pixel x is in the i-th group.

The smoothness term $\mathbb{S}$ ensures similar pixels to have similar label. It is defined as:

$$\mathbb{S}(x, l_x; y, l_y) = \mu(l_x, l_y) \cdot k(x, y). \tag{3.10}$$

Here, μ is again the Pott model. Following [45], the similarity function $k(x, y)$ is defined as:

$$\begin{aligned} k(x,y) = w_1 exp\left(-\frac{|Dis(x,y)|}{2\theta_1^2} - \frac{|I_x - I_y|}{2\theta_2^2}\right) \\ + w_2 exp\left(-\frac{|Dis(x,y)|}{2\theta_3^2}\right). \end{aligned} \tag{3.11}$$

Figure 3.8: Alpha mattes (bottom) extracted by the method proposed in [114] on some examples (top) in our dataset.

which describes the similarities between x and y in their spatial position and color. Here, I_x, I_y are the RGB colors at x, y. In all our experiments,we set $\theta_1 = 6, \theta_2 = 2$, $\theta_3 = 1.7$, the weights of two kernels are $w_1 = 10, w_2 = 3$.

The color difference between two pixels is computed as the L_1 distance between two color vectors. The spatial difference $Dis(x, y)$ for pixels in different frames requires some special handling. For a pixel x in the t_x-th frame and a pixel y in the t_y-th frame, we use optical flow to track x to the frame t_y. The spatial distance is then evaluated as:

$$Dis(x, y) = \left| x + m_{t_x \to t_y}(x) - y \right|, \tag{3.12}$$

where $m_{t_x \to t_y}$ is the motion of pixel x from frame t_x to the frame t_y. This motion vector is obtained by concatenating optical flow vectors from adjacent frames.

Once the graph is constructed, we used the method proposed in [45] to minimize the total energy. Solving the multi-frame CRF enforces temporal coherence. If a pixel is temporally connected to pixels in other frames that have high fence probabilities, optimizing this CRF will help correct its label even its original fence probability is low. Figures 3.6 shows some fence segmentation results after dense CRF optimization. Comparing with the initial segmentation shown in Figures 3.5, the refined segmentation is more accurate on individual frames, and also maintains better temporal coherence.

3.3 Experiments

The dataset. We evaluate our method on a dataset of 18 video clips. Seven of them (the first seven data shown in Figure 3.9) were downloaded from Youtube. The following three (the eighth to tenth shown in Figure 3.9) are from [114]. The rest were captured by ourselves with a mobile phone. Nine of these videos contain moving objects of various sizes. The example "Blue Fence" captures a dynamic scene, and "Running Lion" captures a dynamic scene with large perspective distortion. All videos except "Jaguar" are captured with a

moving camera. Our dataset also includes two examples that contain non-fence occluders: "Wire" and "Tree branch", to test the robustness and generalization of each method.

Figure 3.9 shows some representative frames and their final fence segmentation results. For each example, we show two sample frames, where the segmentation results are overlaid on the input frame. The results show that our method generates accurate and temporally coherent segmentation for most examples.

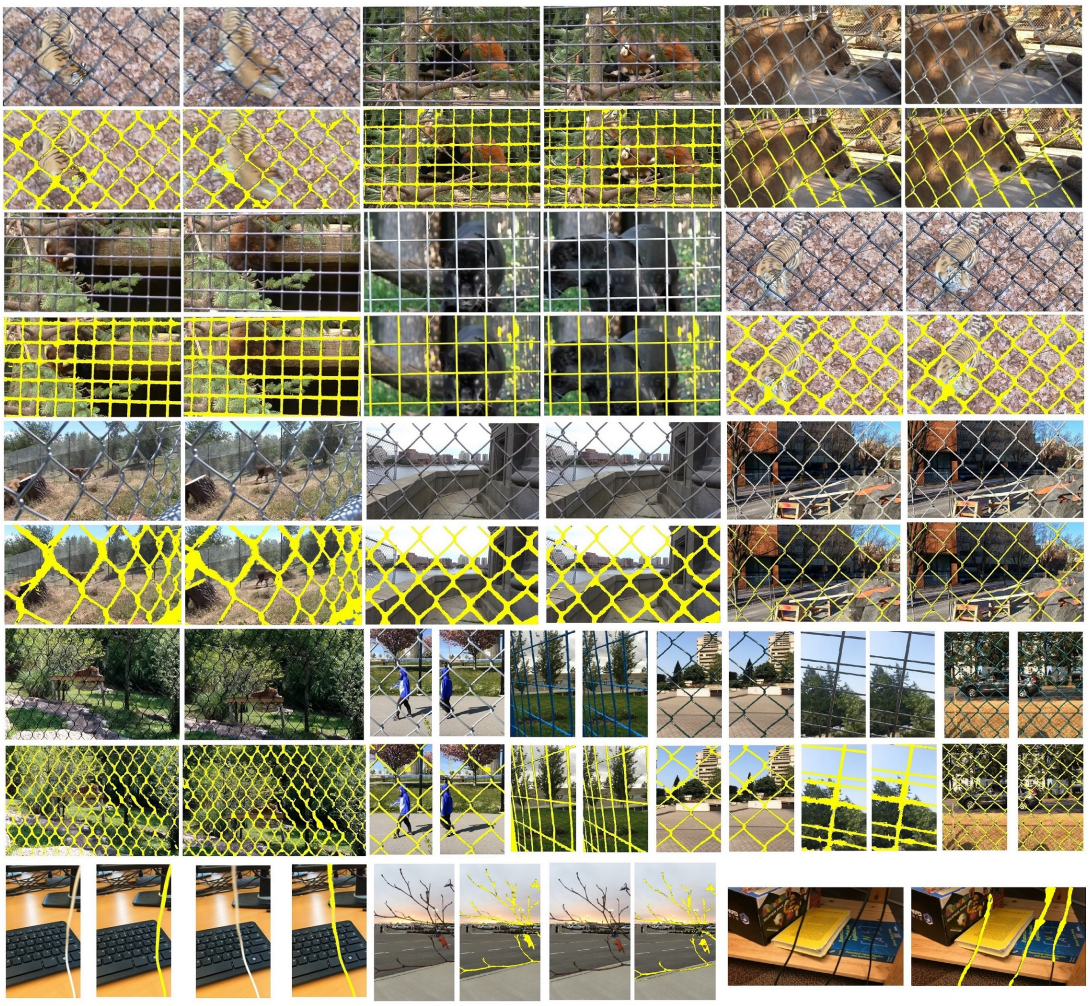


Figure 3.9: More fence segmentation results. For each example, we show two frames with the fence segmentation overlaid.

Evaluation and comparison. In order to quantitatively evaluate the segmentation result, for each video sequence, we manually label "ground truth" segmentation on evenly-distributed ten keyframes. We compare our method with the Rotobrush [2] video segmen-

(a) (b) (c) (d)

Figure 3.10: Comparison with [69] on their data. (a) selected frames from the original video; (b) de-fencing results from [69]; (c) our fence segmentation results; (d) our fence removal results.

tation tool in Adobe After Effect, and the recent method proposed in [114] [1]. Rotobrush is an interactive segmentation tool that needs an manually segmented keyframe as additional input. We thus manually segment the first frame, and use Rotobrush to propagate this segmentation to the next 20 frames for comparison. We limit the propagation to 20 frames, because after that the results are severely deteriorated. Figure 3.7 shows two examples of the manually-segmented keyframes and the automatically segmented results of Rotobrush.

The precision and recall of three methods are shown in tab:evaluation. To demonstrate the effectiveness of the CRF-based refinement, we also compare the initial segmentation computed by graph-cut optimization with the final result produced by the CRF refinement. The results show that our method in general outperforms previous approaches: the average precision and recall for our method are 80.92% and 82.31%, respectively, which are significantly higher than those of the other two methods. Fence segmentation is difficult even for interactive tools such as the Rotobrush. Its average precision and recall are 57.43% and 52.25%, much lower than ours. Looking at individual examples, for videos containing dynamic scenes, the best result is achieved on the "Jaguar" example, due to the fact that its fence color is most distinctive from the background. Our method also achieves reasonable results on the "wire" and "tree branch" example, demonstrating the generalization of our method to non-fence occludes. Furthermore, the dense CRF improves both precision and recall in all examples.

[1] The authors of [114] have kindly generated the alpha matte on one frame for each of our input video. The evaluation of their method is based on that given frame.

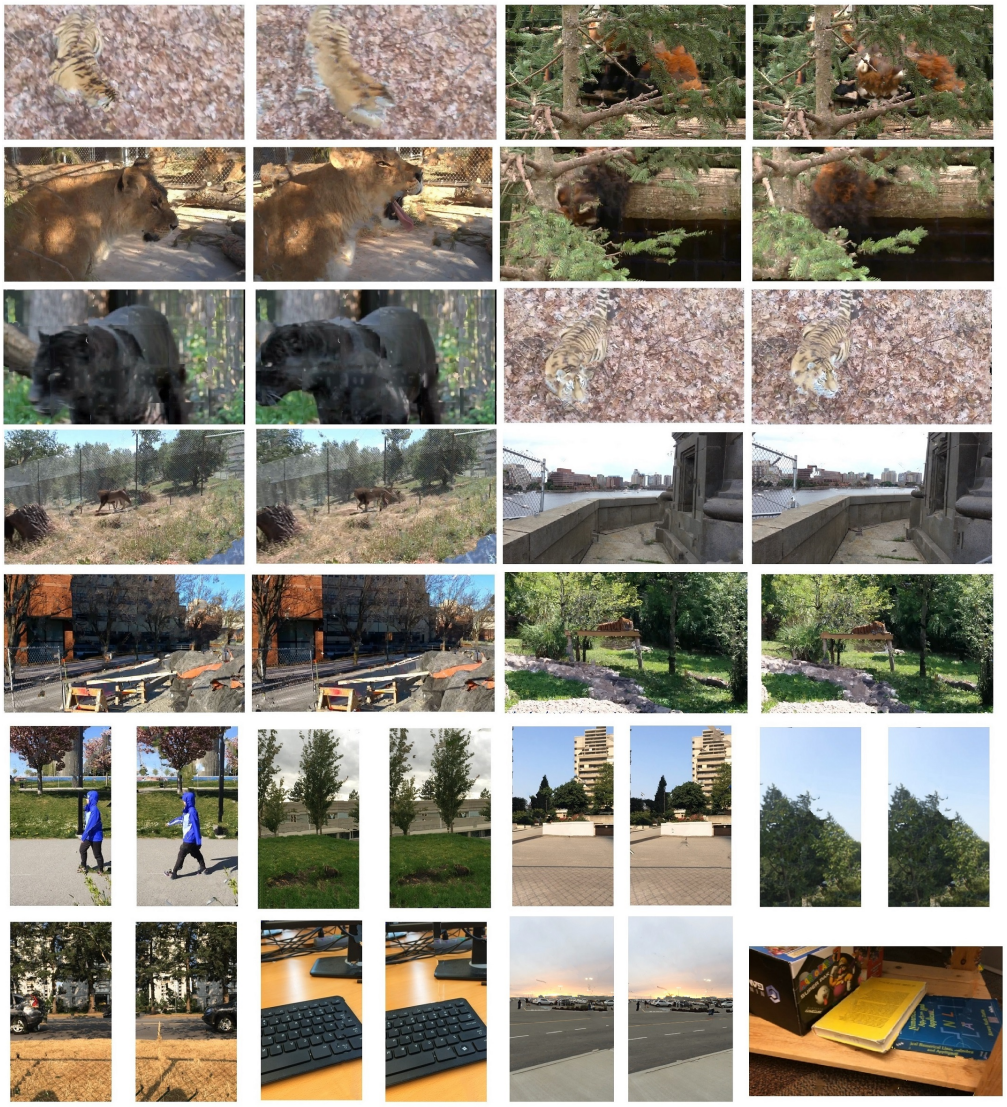


Figure 3.11: Fence removal results on some selected frames.

The method described in [114] produces an alpha matte of the fence for one frame of the input video: some are shown in Figure 3.8. Given that this method is designed for videos with static background, its results are poor on many examples with dynamic backgrounds (e.g. "Tiger"). It also produces poor results on examples captured with a static camera (e.g. "Jaguar"), which violate the underlying assumption of this method. To evaluate their precision and recall, we search through $[0, 1]$ for an optimal threshold that gives the largest value on (precision$\times$recall). The average precision and recall computed in this way are 46.34 and 69.61, respectively, which is significantly lower than ours. Moreover, our method considers both color and motion to form pixel groups. So it can largely tolerate optical flow errors. For example in Figure 3.2, though the flow is quite poor as in Figure 3.2 (b), the pixel-groups in (c)-(f) are quite reasonable.

Data	Rotobrush [2] Precision (%)	Precision (%) of method in [114]	Initial Precision (%)	Final Precision (%)	Rotobrush [2] Recall (%)	Recall (%) of method in [114]	Initial Recall (%)	Final Recall (%)
Tiger1	51.43	15.73	76.00	**78.97**	58.22	**95.61**	59.93	77.42
Little Panda1	51.30	19.63	78.13	**80.05**	**82.36**	64.56	75.04	78.91
Lion	78.08	53.61	67.49	**80.49**	49.78	69.89	61.11	**80.58**
Little Panda2	37.36	20.94	77.10	**80.19**	24.68	66.77	77.38	**78.92**
Jaguar	**96.47**	11.59	81.09	86.93	47.04	59.27	77.60	**90.54**
Tiger2	57.22	18.78	78.05	**84.09**	44.20	39.00	75.65	**82.70**
Running Lion	32.27	70.27	71.54	**79.75**	64.74	81.06	70.15	**86.72**
Gray Fence1	**85.84**	76.88	71.44	78.48	32.86	**88.25**	70.13	83.88
Gray Fence2	32.56	53.47	77.03	**80.76**	11.94	**85.56**	75.52	80.37
Zoo	31.22	59.01	73.57	**82.39**	2.16	69.57	72.23	**84.72**
Walking Person	70.57	51.73	74.19	**78.49**	**82.79**	67.90	67.50	79.95
Blue Fence	56.89	20.64	84.21	**89.41**	62.80	75.66	65.61	**91.51**
Building	61.77	84.89	77.29	**85.02**	74.82	66.96	80.36	**82.82**
Tree	76.14	49.24	77.82	**82.19**	69.97	41.95	78.73	**81.76**
Car	41.75	59.79	75.81	**81.51**	81.61	74.49	61.80	**83.93**
Wire and Keyboard	67.46	56.81	62.67	**90.58**	76.33	87.82	67.56	**89.25**
Tree Branch	67.68	\	50.80	**74.19**	39.62	\	58.80	**73.44**
Wires	34.74	**78.99**	74.27	63.01	34.56	49.13	62.41	**79.22**
Average Value	57.43	46.34	73.81	**80.92**	52.25	69.61	69.86	**82.31**

Table 3.1: Precision and recall of initial segmentation and final segmentation.

Meanwhile, we also tested our methods on data from [69] to provide a direct comparison. Since there are no video fence segmentation results provided, we compared with it by fence removal results as shown in Figure 3.10 On the second example, our method produces superior results in the red circles which suggests better fence segmentation. Please note that [69] can only deal with static scenes.

Fence removal: Once the fence is segmented, we can apply existing image and video inpainting techniques, such as [16], to remove the fence from video frames. Figure 3.11 shows some frames with fence removed using the method in [16]. We believe better fence removal can be achieved by exploiting multiple frame information such as in [112], which is our future work.

3.4 Conclusion

We present a fully-automatic method to detect and segment fence-like occluders from a video clip to generate a fence-free photo. The main advantage of our method over previous work is that it handles both dynamic scenes and moving cameras. Our method first groups pixels according to their motion and color similarity. It then exploits spatial structural features in a graph-cut optimization framework to produce initial segmentation. The initial segmentation is further refined by solving a dense CRF to achieve better spatial accuracy and temporal coherence. Fence removal is demonstrated with existing inpainting techniques, which shows that our method is a promising building block towards a fully automatic, high quality fence removal solution for general videos.

Lastly, we would like to discuss the handcrafted features proposed in this method. We build the gradient-based feature based on the assumption that fences should have two nearly perpendicular dominant peaks. Thus this feature can deal with most fences and regular tree branches. For other occluders that does not satisfy the assumption, we can modify the features accordingly, such as round fences or other shape of occluders. Similarly, the

geometry-based term in this proposed method is based on the assumption of thin structures. For scenarios of non-thin structures, we should modify this term accordingly by other assumptions of geometry to accommodate a broader range of inputs.

Chapter 4

Unsupervised Training Scheme for Deep Learning Methods

In this chapter, we introduce the main challenges of previous deep learning methods, which is the lack of real-image training data with ground truth. In order to solve this problem, we proposed an unsupervised training scheme for utilizing real-image training data for training deep neural networks for layer separation problems. We demonstrate two applications using the proposed unsupervised training scheme, which are highlight separation of face images, and non-Lambertian intrinsic image decomposition (end-to-end separation of multiple reflectance layers, which are the highlight, diffuse, reflectance/albedo, and shading). The experiments in Chapter 5-6 show that these networks trained by real training data outperformed previous methods trained on synthetic data, and achieved the state-of-art performance on both tasks.

4.1 Challenges

Factorizing an image into multiple image layers is an ill-posed problem that is best solved at present through deep learning. The main challenge in this task is the lack of ground truth separation data on real images. Although ground truth separations can be generated synthetically using graphics models [99], it has become known that the mismatch between real and synthetic data can lead to significant reductions in performance [100]. Although for some tasks, like highlight separation, it is possible to capture ground truth data in a lab setting by cross-polarization, but it can only cover linear illuminations, while most of the natural illumination in real scenes are nonlinear. Capturing a dataset that is large enough for training DNNs also requires a heavy workload. Generating movie-quality synthetic data will also require a lot of computational resources and a long time.

Thus, obtaining large-scale ground-truth realistic data for training deep neural networks remains a challenge, and this has motivated recent work on developing unsupervised schemes for the image layer separation problems like intrinsic image decomposition. The

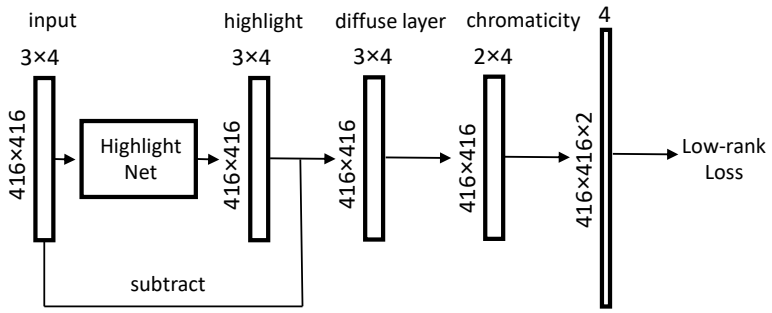


Figure 4.1: Network structure for the unsupervised training of Highlight-Net.

unsupervised techniques that have been presented thus far all take the same approach of training a network on image sequences of a fixed scene under changing illumination [58,65]. With images from such a sequence, these methods guide network training by exploiting the albedo consistency that exists for each scene point throughout the sequence. However, these works require image sequences of fixed scenes, which is relatively difficult to get. Thus unsupervised training scheme for unconstrained image sequences from random views is worthy to explore.

4.2 Proposed unsupervised training scheme on unconstrained real data

In order to make use of unconstrained real data without ground truth, we present an unsupervised strategy for training networks for separating image layers. Taking highlight layer separation as the example here, we aim to train a deep neural network called Highlight-Net to predict highlight layers from the observed image. This unsupervised training strategy is based on the observation that an object's surface features should remain the same, so the diffuse chromaticity over a given object should be consistent in different images from the same viewpoint, regardless of illumination changes. Thus, a matrix constructed by stacking the aligned diffuse chromaticity maps of an object should be low rank. In place of ground-truth highlight layers of real object images, we use this low-rank property of ground-truth diffuse layers to train our Highlight-Net.

This unsupervised training is implemented using the network structure shown in Figure 4.1, where Highlight-Net is augmented with a low-rank loss, assuming we have a set of aligned images under random illuminations for each object. The data preparation will be described in specific problems in Chapter 5-6.

During training, four aligned images of the same object are randomly selected for each batch. A batch is fed into Highlight-Net to produce the estimated highlight layers for the

four images. These highlight layers are subtracted from the original images to obtain the corresponding diffuse layers. For a diffuse layer I_d, its diffuse chromaticity map is computed per-pixel as

$$chrom(I_d) = \frac{1}{(I_d(r) + I_d(g) + I_d(b))} (I_d(r), I_d(g)) \tag{4.1}$$

where r, g, and b denote the color channels. Each diffuse chromaticity map is then reshaped into a vector I^{dc}, and the vectors of the four images are stacked into a matrix $D = \left[I_1^{dc}, I_2^{dc}, I_3^{dc}, I_4^{dc}\right]^T$. With a low-rank loss enforced on D, Highlight-Net is trained through backpropagation.

Since the diffuse chromaticity of the same object should be consistent among images, the rank of matrix D should ideally be one. So we define the low-rank loss as its second singular value, during backpropagation the partial derivative of σ_2 with respect to each matrix element is evaluated according to [77]:

$$\begin{aligned} &D = U\Sigma V^T, \qquad \Sigma = diag(\sigma_1, \sigma_2, \sigma_3, \sigma_4), \\ &loss_{lowrank} = \sigma_2, \qquad \frac{\partial \sigma_2}{\partial D_{i,j}} = U_{i,2} \times V_{j,2}. \end{aligned} \tag{4.2}$$

The proposed unsupervised training scheme is applied to the task of highlight separation for face images in Chapter 5, and the task of end-to-end highlight separation and intrinsic image decomposition for general objects, as described in Chapter 6. In the second task, the low-rank loss is further improved to be misalignment-robust. Details are described in Chapter 6.

Chapter 5

Unsupervised Face Highlight Separation

In this chapter, we present a highlight separation method for face images, based on the proposed unsupervised training scheme presented in Chapter 4.

5.1 Introduction

Specular highlights removal is the task of removing specular highlights from images of non-Lambertian surfaces, which is an important preprocessing step for many following tasks, such as object recognition and detection. These tasks usually assume a Lambertian surface and treat specular highlights as noises, while the majority of real-life objects are non-Lambertian and exhibit specular highlights. Therefore, methods assuming Lambertian surfaces may fail due to the undesired discontinuities caused by highlights, highlights need to be removed beforehand to remove noises caused by such discontinuities. Furthermore, due to the brightness of highlight regions, the image pixels may be saturated or cause the reduction of contrast.

Previous work of specular highlight removal usually relies on additional observations or priors, such as white illumination and repetitive patterns. However, these observations only work for specific scenarios, for images captured under natural lighting, some surface properties may not exhibit. Recently, methods based on deep learning are proposed by direct supervision by rendered synthetic dataset [99]. However, networks trained by synthetic data are not working well for real images due to the domain shift between them. Generating movie-quality synthetic data will be expensive while capturing a real dataset with ground truth by cross-polarization is also impractical due to the complexity of light sources in natural scenes.

Focusing on the highlight removal of facial images, photos of human faces often exhibits strong specular highlights due to the oily skin surfaces. Specular highlights removal from

face images is desirable for photo editing or facilitate other tasks, such as facial landmark detection or face recognition.

In this work, we present a deep neural network for separating specular highlights from diffuse reflections in face images. The main challenge in this task is the lack of ground truth separation data on real face images for use in network training. Although ground truth separations can be generated synthetically using graphics models [99], it has become known that the mismatch between real and synthetic data can lead to significant reductions in performance [100]. We deal with this issue by pretraining our network with a small set of synthetic images and then finetuning the network using an unsupervised strategy with real photos. Since there is little real image data on ground truth separations, we instead take advantage of the property that the diffuse chromaticity values over a given person's face are relatively unchanged from image to image, aside from a global color rescaling due to different illumination colors and sensor attributes. From this property, we show that the diffuse chromaticity of multiple aligned images of the same face should form a low-rank matrix. We utilize this low-rank feature in place of ground truth separations to finetune the network using multiple real images of the same face, downloaded from the MS-celeb-1M database [27]. This unsupervised finetuning is shown to significantly improve highlight separation over the use of supervised learning on synthetic images alone. This method is validated through experimental comparisons to previous techniques for highlight extraction and our method is shown to produce results that more closely match the ground truth acquired by cross-polarization.

5.2 Pretraining with synthetic data

For Highlight-Net, we adopt a network structure used previously for intrinsic image decomposition [71], a related image separation task. To pretrain this network, we render synthetic data using generic face models [80] and real indoor and outdoor HDR environment maps collected from the Internet. Details on data preparation are presented in Section 5.4.1. With synthetic ground truth specular images, we minimize the L2 loss between the predicted and ground truth highlights for pretraining.

5.3 Unsupervised finetuning on real images

With only pretraining on synthetic data, Highlight-Net performs inadequately on real images. This may be attributed to the limited variation of face shapes, textures, and environment maps in the synthetic data, as well as the gap in appearance between synthetic and real face images. Since producing a large-scale collection of real ground-truth highlight separation data is impractical, we present an unsupervised strategy for finetuning Highlight-Net that only requires real images of faces under varying illumination environments.

Figure 5.1: Examples of selected aligned photos for four celebrities.

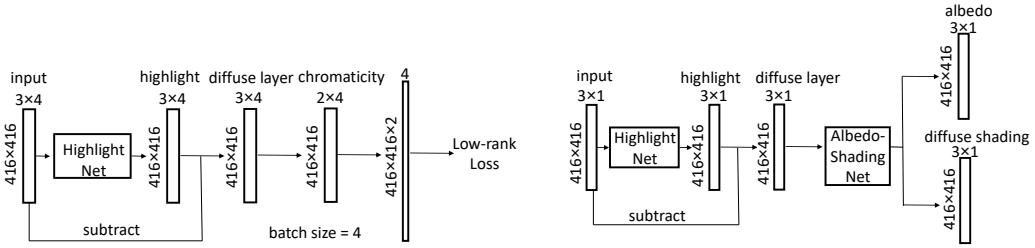


Figure 5.2: (a) Network structure for finetuning Highlight-Net; (b) Testing network structure for separating an input face image into three layers: highlight, diffuse shading, and albedo.

This strategy is based on the observation that the diffuse chromaticity over a given person's face should be consistent in different images, regardless of illumination changes, because a person's facial surface features should remain the same. Among images of the same face, the diffuse chromaticity map should differ only by global scaling factors determined by illumination color and sensor attributes, which we correct in a preprocessing step. Thus, a matrix constructed by stacking the aligned diffuse chromaticity maps of a person should be of low rank. In place of ground-truth highlight layers of real face images, we use this low-rank property of ground-truth diffuse layers to finetune our Highlight-Net.

This finetuning is implemented using the network structure shown in Figure 5.2 (a), where Highlight-Net is augmented with a low-rank loss. The images for training are taken from the MS-celeb-1M database [27], which contains 100 images for each of 100,000 celebrities. After some preprocessing described in Section 5.4.1, we have a set of aligned frontal face images under a consistent illumination color for each celebrity as the examples shown in Figure 5.1.

In the training, four face images of the same celebrity are randomly selected for each batch from the dataset. A batch is fed into Highlight-Net to produce the estimated highlight layers for the four images. These highlight layers are subtracted from the original images to obtain the corresponding diffuse layers. For a diffuse layer I_d, its diffuse chromaticity map is computed per-pixel as in Equation 4.1.

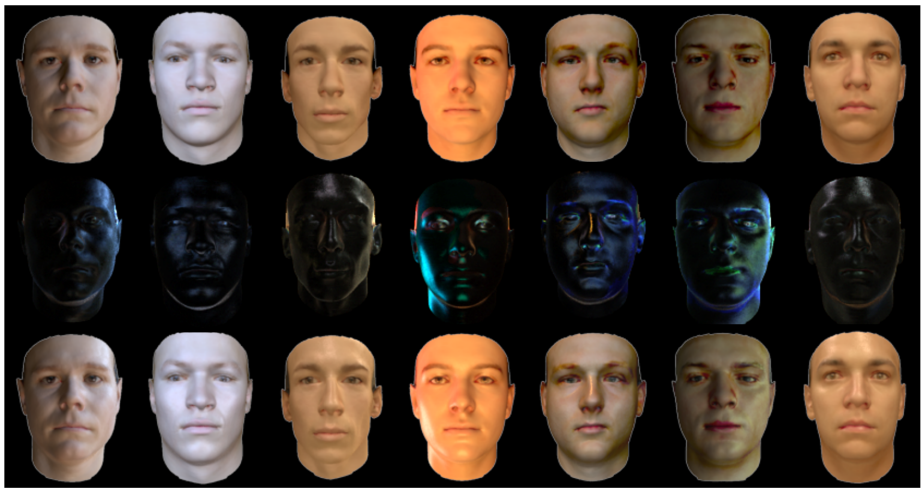


Figure 5.3: Examples of rendered synthetic faces. The top row shows rendered diffuse components; the middle row displays rendered specular components; and the bottom row are composite renderings that combine the diffuse and specular layers.

Each diffuse chromaticity map is then reshaped into a vector I^{dc}, and the vectors of the four images are stacked into a matrix $D = \left[I_1^{dc}, I_2^{dc}, I_3^{dc}, I_4^{dc}\right]^T$. With a low-rank loss enforced on D, Highlight-Net is finetuned through backpropagation.

Since the diffuse chromaticity of a face should be consistent among images, the rank of matrix D should ideally be one. So we define the low-rank loss as in Equation 4.2, during backpropagation the partial derivative of σ_2 with respect to each matrix element is evaluated according to [77].

5.4 Experiments

5.4.1 Training data

For the pretraining of Highlight-Net, we use the Basel Face Model [80] to randomly generate 50 3D faces. For each face shape, we adjust the texture map to simulate three different skin tones. These 150 faces are then rendered under 200 different HDR environment maps, including 100 from indoor scenes and 100 from outdoor scenes. The diffuse and specular components are rendered separately, where a spatially uniform specular albedo is randomly generated between $[0, 1]$. For training, we preprocessed each rendering by subtracting the mean image value and then normalizing to the range [0,1]. Examples of rendered synthetic faces are shown in Figure 5.3.

In finetuning Highlight-Net, the image set for each celebrity undergoes a series of commonly-used preprocessing steps so that the faces are aligned, frontal, radiometrically calibrated, and under a consistent illumination color. For face frontalization, we apply the method in [28]. We then identify facial landmarks [124] to crop and align these frontal

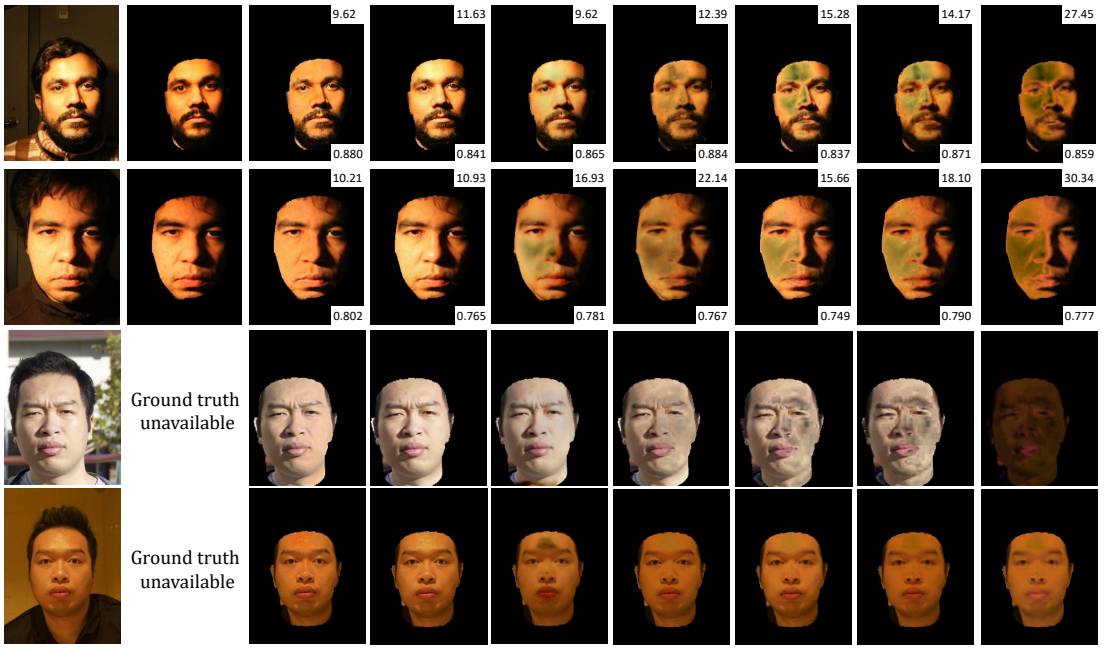


Figure 5.4: Highlight removal comparisons on laboratory images with ground truth and on natural images. Face regions are cropped out automatically by landmark detection [124]. (a) Input photo. (b) Ground truth captured by cross-polarization for lab data. (c-h) Highlight removal results by (c) our finetuned Highlight-Net, (d) Highlight-Net without finetuning, (e) [99], (f) [53], (g) [95], (h) [117], and (i) [106]. For the lab images, RMSE values are given at the top-right, and SSIM [110] (larger is better) at the bottom-right.

faces. The cropped images are radiometrically calibrated by the method in [52], and their color histograms are matched by the built-in histogram transfer function in MATLAB [67] to reduce illumination color differences. We note that in each celebrity's set, images were manually removed if the face exhibits a strong expression or multiple lighting colors, since these cases often lead to inaccurate spatial alignment or poor illumination color matching. Some examples of these preprocessed images are presented in Figure 5.1.

5.4.2 Evaluation of highlight removal

To examine highlight extraction performance, we compare our highlight removal results to those of several previous techniques [53, 95, 99, 106, 117] in Figure 5.4. The first two rows show results on faces with known ground truth captured by cross-polarization under an indoor directional light. In order to show fair comparisons for both absolute intensity errors and structural similarities, we use both RMSE and SSIM [110] as error/similarity metrics. The last two rows are qualitative comparisons on natural outdoor and indoor illuminations, where ground truth is unavailable due to the difficulty of cross-polarization in general settings. In all of these examples, our method outperforms the previous techniques,

	Synthetic data						Real data					
	Ours	[99]	[53]	[95]	[117]	[106]	Ours	[99]	[53]	[95]	[117]	[106]
Mean RMSE	**3.37**	4.15	5.35	6.75	8.08	28.00	**7.61**	8.93	10.34	10.51	11.74	19.60
Median RMSE	**3.41**	3.54	4.68	6.41	7.82	29.50	**6.75**	8.71	10.54	9.76	11.53	22.96
Mean SSIM	**0.94**	**0.94**	0.92	0.91	0.91	0.87	0.89	0.89	**0.90**	0.86	0.88	0.88
Median SSIM	**0.95**	0.94	0.92	0.91	0.91	0.87	0.90	0.90	**0.91**	0.88	0.90	0.89

Table 5.1: Quantitative highlight removal evaluation.

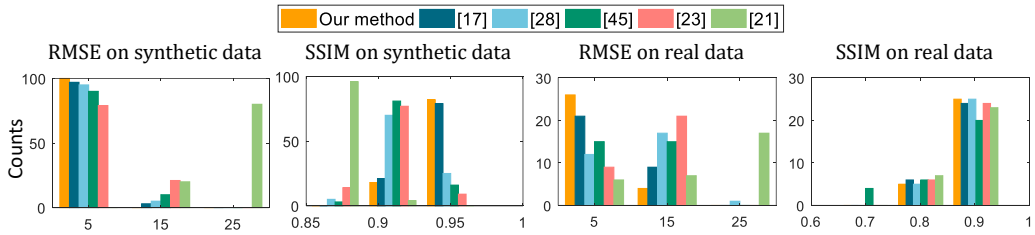


Figure 5.5: Quantitative comparisons on highlight removal for 100 synthetic faces and 30 real faces in terms of RMSE and SSIM histograms (larger SSIM is better).

which generally have difficulty in dealing with the saturated pixels that commonly appear in highlight regions. We note that since most previous techniques are based on color analysis and the dichromatic reflection model [94], they cannot process grayscale images, unlike our CNN-based method. While testing on grayscale images, we duplicate the one channel in grayscale images to three channels, as input for Highlight-Net. The figure also illustrates the importance of training on real image data. Comparing our finetuning-based method in (c) to our method without finetuning in (d) and a CNN-based method trained on synthetic data [99] in (e) shows that training only on synthetic data is insufficient, and that our unsupervised approach for finetuning on real images substantially elevates the quality of highlight separation.

Quantitative comparisons over 100 synthetic faces and 30 real faces are presented in Table 5.1. Error histograms and image results are shown in Figure 5.5. Visual comparisons of synthetic data are presented in Figure 6.5.

To show the robustness of Highlight-Net, we tested hard examples like non-neutral expressions, with occluders like glasses or beard, and various ages or skin tones, we provide additional results in Figure 5.7, which indicate reasonable performance.

5.5 Conclusion

We propose a network to remove highlight reflections from faces. Our network is able to make use of unlabeled real facial images in MS-Celeb-1M database [27], and perform an unsupervised finetuning. The Highlight-Net finetuned on real images significantly outperforms the one only trained on synthetic images. The proposed unsupervised training scheme and the low-rank loss can be adopted for other tasks such as intrinsic image decomposition.

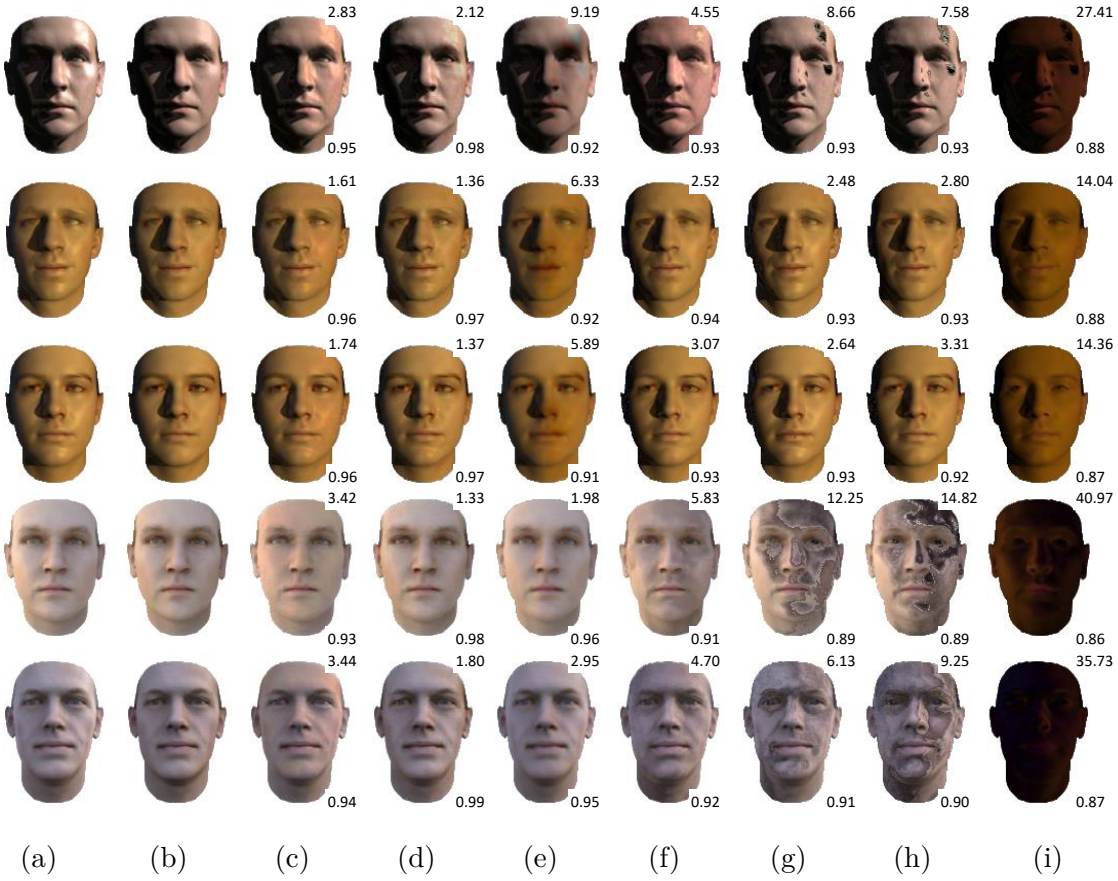


Figure 5.6: Highlight removal comparisons on a subset of the synthetic images. (a) Input photo. (b) Diffuse rendering under the same illumination. (c-h) Highlight removal results by (c) our method, (d) our pretrained net, (e) [99], (f) [53], (g) [95], (h) [117], and (i) [106]. RMSE values are given at the top-right, and SSIM at the bottom-right. RMSE and SSIM are computed on highlight layers.

Figure 5.7: Evaluation of highlight removal on testing data with non-neutral expressions, occluders and various ages/skin tones. Input images are shown on the top row, and corresponding highlight removal results are shown on the bottom row.

The face highlight removal method can also sparkle a set of following applications, such as estimating the environment illumination for realistic augmented reality applications.

Chapter 6

Unsupervised Non-Lambertian Intrinsic Image Decomposition

In this chapter, we present a non-Lambertian intrinsic image decomposition (highlight separation and intrinsic image decomposition) method for general objects, based on the proposed unsupervised training scheme presented in Chapter 4.

6.1 Introduction

Separating reflectance layers in an image is an essential step for various image editing and scene understanding tasks. One such layer is composed of highlights, which are mirror-like reflections off the surface of objects. Extracting highlights from an image can be useful for problems such as estimating scene illumination [62,120] and reducing the oily appearance of faces [55]. The other two layers represent shading and albedo. Their separation is commonly known as intrinsic image decomposition, which has been utilized in applications such as shading-based scene reconstruction [75,121] and texture replacement in images [34,111].

Factorizing an image into the three reflectance layers is an ill-posed problem that is best solved at present through machine learning. However, obtaining large-scale ground-truth data for training deep neural networks remains a challenge, and this has motivated recent work on developing unsupervised schemes for the reflectance separation problem. The unsupervised techniques that have been presented thus far all take the same approach of training a network on image sequences of a fixed scene under changing illumination [58,65]. With images from such a sequence, these methods guide network training by exploiting the albedo consistency that exists for each scene point throughout the sequence.

A benefit of using image sequences of fixed scenes is that the images are perfectly aligned, allowing scene point consistency to be easily utilized. However, there exists an untapped wealth of image data captured of objects from different viewpoints. A prominent example of such data is customer product photos uploaded by consumers to show items they bought. Some example customer photos are shown in Figure 6.1. This source of imagery is valuable

not just because of its vast quantity online, but also because it provides object-centric data (different from the scene data compiled in [58] from webcams) and can promote robustness of factorizations to different object orientations. These images also exhibit a larger variation in illumination conditions and camera settings, which can potentially benefit the trained network. An issue with using such images though is that they are difficult to align accurately, as they vary in viewpoint, lighting and imaging device. Misalignment among the images of an object would lead to violations of scene point consistency on which the existing unsupervised methods are based.

In this chapter, we present an unsupervised method for reflectance layer separation using multi-view image sets such as customer product photos. To effectively learn from such data, our system is designed so that its training is relatively insensitive to misalignments. After approximately aligning images with state-of-the-art correspondence estimation techniques [32, 88], the network transforms the images into a proposed representation based on local color distributions. An important property of this representation is its ability to model detailed local content over an object in a manner that discards fine-scale positional information. With this color distribution based descriptor, unsupervised training becomes possible using consistency constraints between multi-view images of an object.

An additional contribution of this work is a method for further guiding the unsupervised training via a relationship between highlight separation and intrinsic decomposition of shading and albedo. We observe that shading separation becomes less reliable when highlights are present in its input images, due to color distortions caused by different highlight saturation and possibly different illumination color among the images. Our system takes advantage of this through a novel contrastive loss that is defined between shading separation results computed with and without the inclusion of our highlight extraction sub-network. We show that by maximizing this contrastive loss, the shading separation sub-network provides supervision that improves the performance of the highlight extraction sub-network.

The main contributions of this work can be summarized as follows:

1. A proposed color distribution loss that is robust to spatial misalignment, a major issue for networks that assume exact pixel-to-pixel correspondence of images.

2. The large-scale Customer Product Photos Dataset, which can also be used for tasks other than reflectance separation, such as shape-from-shading and multi-view stereo.

3. A network to separate highlights, albedo and shading through unsupervised training on multiview images.

With the presented approach, our system produces state-of-the-art results for highlight separation and intrinsic image decomposition on real-world objects.

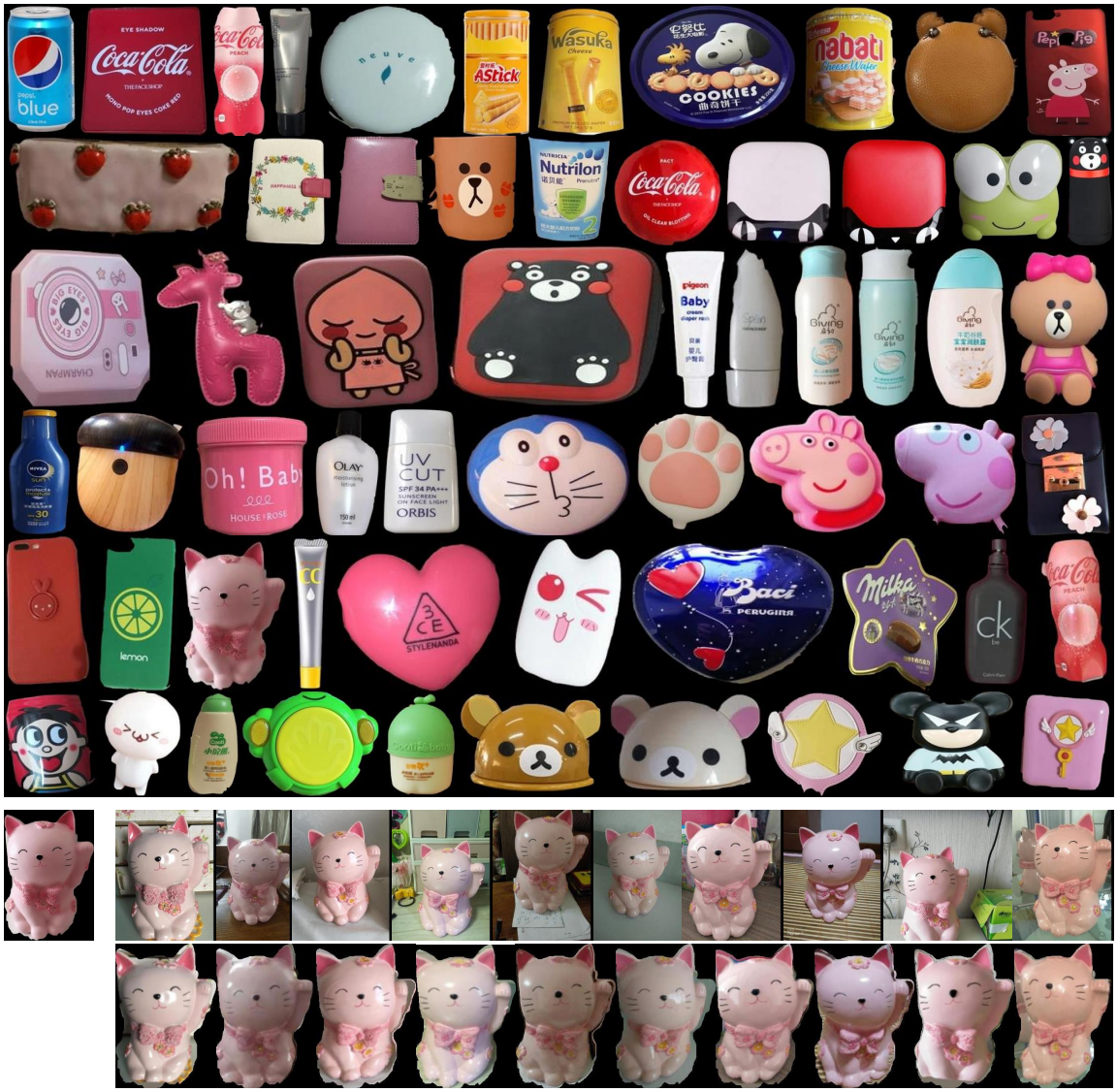


Figure 6.1: Selected product photos from the Customer Product Photos Dataset. The products exhibit a wide range of textures, shapes, shadings, and highlight patterns. The second last row shows selected multiview images of the same object, where the leftmost one is the segmented reference image. The last row shows the roughly aligned images.

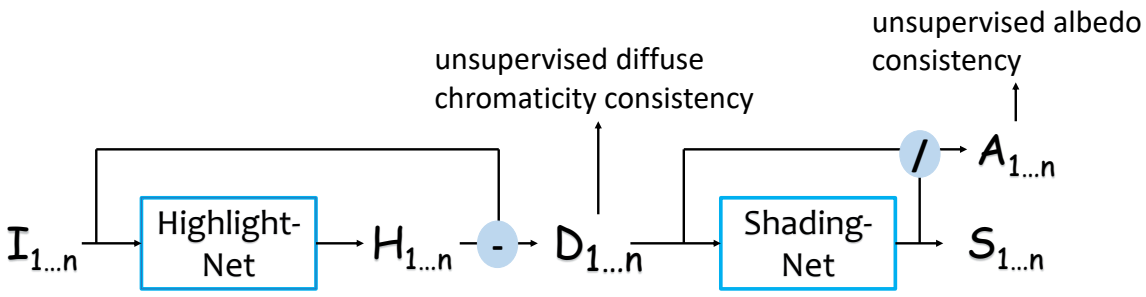


Figure 6.2: Network structure.

6.2 Overview

We train an end-to-end deep neural network to separate a single image into highlight, albedo/reflectance, and shading layers using the Customer Product Photos Dataset. Compiled from online shopping websites, the dataset contains numerous product photos provided in customer reviews. The photos for a given product are captured under various viewpoints, illumination conditions, and backgrounds. We introduce this dataset in Section 6.3.

As illustrated in Figure 7.1, our network consists of two subnets: Highlight-Net for decomposing an image into diffuse and highlight layers, and Shading-Net for additionally decomposing the diffuse layer into albedo and shading layers. Training consists of three phases. First, both Highlight-Net and Shading-Net are pretrained using a small set of synthetic data from [99]. Each subnet is then finetuned in an unsupervised manner on the Customer Product Photos Dataset using the proposed color distribution loss (Section 6.4.3), which is robust to misalignments. In the last phase, a novel contrastive loss is used to finetune the whole network end-to-end. The training phases are presented in Section 6.4.

6.3 Customer product photos dataset

Almost every popular online shopping website includes customer reviews, where customers are often encouraged to upload product photos. For a given product, the customer photos capture it under a various viewpoints, illuminations, and backgrounds. At the same time, the different products cover a large variety of materials and shapes. Collectively, these customer photos capture the complex interaction between different 3D shapes, materials, and illumination, and form a dataset that can be useful for computer vision tasks such as intrinsic image decomposition and multi-view stereo.

Construction of the dataset involved the following steps:

1. Product selection: We manually select product pages containing many customer photos and for which the product does not have multiple versions (e.g., different colors,

textures or shapes), so that the product is the same in each photo. We also favor products with an apparent front side, which facilitates alignment.

2. Photo downloading: We then download customer photos of selected products with batch downloading tools.

3. Rough image alignment: For each product, we select one image as the reference and manually segment the object to remove the background. The unconstrained viewpoints and illumination differences among the images makes alignment challenging. We first use WeakAlign [88] to align each of the other images to the segmented reference by an affine transformation. After this global parametric warping, we use FlowNet2.0 [32] to further align the warped images to the reference. After the transformations of these two steps, the objects in each image will roughly but imperfectly align to the reference. The foreground mask of the reference is used to segment the objects after this alignment. An example of this alignment is shown in the last two rows of Figure 6.1.

4. Data filtering: Customer photos exhibit large differences in illumination color as well. To simplify our task, we select photos whose illumination color is similar to that of the reference. This similarity is measured by the difference in median chromaticity. We keep only the top 20% of images by this metric. No white balancing is applied, and a gamma 2.2 is assumed for radiometric calibration. Then we manually check all the images and remove those with unsuitable content or poor alignment.

The final Customer Product Photos Dataset consists of 228 products (some shown in Figure 6.1) with 10–520 photos for each product. In total, the dataset consists of 9,472 photos. For each product, there is one mask provided for the reference image.

6.4 Our network

6.4.1 Problem formulation

An input image I comprises an additive combination of a highlight layer H and a diffuse layer I_d, where the diffuse layer I_d is a pixelwise product of an albedo/reflectance layer A and a shading layer S, i.e.,

$$I = H + I_d = H + A \cdot S. \tag{6.1}$$

Our problem is to estimate H, I_d, A, S from the input image I. We note that this image model differs from the conventional intrinsic image model, $I = A \cdot S$, which omits the additive effects of highlights and thus implicitly assumes object surfaces to be matte [99].

6.4.2 Unsupervised training with low-rank loss

Most CNN-based methods [4, 33, 70, 99] for intrinsic image separation require ground truth separation results for supervised training. As it is difficult to obtain reference ground truth for highlight separation or intrinsic image decomposition, we propose to train our network

by unsupervised learning after an initial pretraining step with synthetic data from objects in the ShapeNet dataset [99]. This pretraining uses 28,000 out of the 2,443,336 images in the dataset, or about 1.1% of the total, and is intended to provide the network with a good initialization.

We first assume perfect image alignment in deriving the low-rank loss for unsupervised training. This requirement on alignment will be relaxed in the next subsection.

Unsupervised training of Highlight-Net For training of highlight separation, our network utilizes input consisting of multiple aligned images $I_1, I_2, I_3, \cdots$ of the same object under different lighting. According to the image formation model, these images each have a diffuse layer, denoted as $I_{d1}, I_{d2}, I_{d3}, \cdots$. These diffuse layers can differ from each other due to changes in shading that arise from different illumination conditions. To discount this shading variation, we compute the chromaticity maps of these diffuse layers. A chromaticity map (Ch_r, Ch_g) is an intensity-normalized image, where

$$Ch_r(p) = \frac{R(p)}{R(p) + G(p) + B(p)},$$

$$Ch_g(p) = \frac{G(p)}{R(p) + G(p) + B(p)},$$

at each pixel p, with $R(p), G(p), B(p)$ denoting the color values at p.

According to the dichromatic reflectance model [94], the chromaticity of diffuse layers is the chromaticity of the surface albedo multiplied with that of the illumination. Assuming a constant illumination color across each image, we discount the effect of illumination chromaticity by matching the median chromaticity of each diffuse image to that of the reference image in each batch. After these normalizations, the set of chromaticity maps should be of low rank if the images are accurately aligned.

The structure of Highlight-Net is adopted from the encoder-decoder network in [71] with an added batch normalization layer after each convolution layer to aid in network convergence. We also examined adding skip connections between the encoder and decoder as done in [99], but we found them not to be helpful in our network.

Unsupervised training of Shading-Net Our Shading-Net for predicting the shading layer S uses the same network structure as Highlight-Net. The albedo layer A is computed from S at each pixel p according to the image formation model, as

$$A(p) = I_d(p)/S(p), \tag{6.2}$$

once the shading layer is fixed.

For multiple aligned diffuse images I_{d1}, I_{d2}, I_{d3}, of the same object, their albedo layers $A_1, A_2, A_3, \cdots$ should be the same. Therefore, we can enforce a consistency loss on these different albedo layers for unsupervised training of Shading-Net.

Low-rank loss Our unsupervised training enforces consistency among diffuse chromaticity layers and albedo layers via a low-rank loss. For the case of albedo layers, the low-rank loss can be defined as the second singular value of the matrix M formed by reshaping each albedo image into a vector and stacking the vectors of multiple images [120]. Although consistency could alternatively be enforced through minimizing L1 or L2 differences, e.g. minimizing $|A_1 - A_2|_{1,2}$, the lack of scale invariance of the L1 and L2 losses can lead to degenerate results where A_1 and A_2 approach zero. To avoid this problem, the loss function should satisfy the following constraint,

$$loss(A_1, A_2) = loss(\alpha A_1, \alpha A_2),$$

where α is a global scale factor for the whole albedo image.

In order to make the low-rank loss scale-invariant, we use the first singular value to approximate the scale and define a scale-invariant low-rank loss (SILR) as

$$\begin{aligned} &Loss_{SILR} = \sigma_2/\sigma_1, \\ &\frac{\partial Loss_{SILR}}{\partial M_{i,j}} = \frac{\sigma_1 * (U_{i,2} \times V_{2,j}) - \sigma_2 * (U_{i,1} \times V_{1,j})}{\sigma_1^2}. \end{aligned} \tag{6.3}$$

where σ_1 and σ_2 are the first two singular value of M computed by SVD decomposition. We apply this scale-invariant low-rank loss (SILR) to train both Highlight-Net and Shading-Net.

6.4.3 Misalignment-robust color distribution loss

We present a way to relax the requirement of pixel-to-pixel correspondence in the low-rank loss, so that customer photos can be effectively utilized for training. Our observation is that, though precise pixelwise alignment is generally difficult, the state-of-the-art alignment algorithms, e.g. WeakAlign [88] and FlowNet [17, 32], are mature enough to establish a reasonable approximate alignment. Thus, though some pixels may be misaligned, their correct correspondences are still within a small neighborhood of their estimated locations. This motivates us to develop a local distribution based representation for the low-rank loss.

Suppose we have a predicted albedo layer A. We partition it into a grid of N cells. Within each cell, we reorder the pixels by increasing intensity. This is done for each color channel individually, and all the cells for all the color channels are reshaped and concatenated to form a new vector representation for the image. The color distribution loss is then computed as the SILR of these image vectors. In our implementation, we divided 320×320 images into 256 grid cells for all training phases.

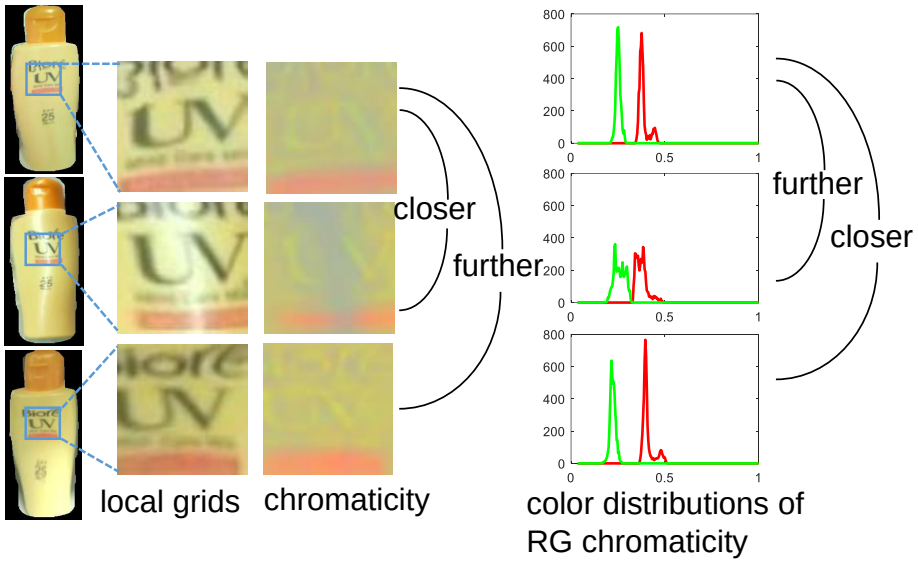


Figure 6.3: Distances between color distributions are more sensitive to the presence of highlights than to pixel-to-pixel distance between misaligned images. The grid cells in the top two images are spatially closer to each other, but have greater difference in color distribution due to highlights.

This vector representation of locally re-ordered pixel values is robust to slight misalignment for the following reasons: (1) Since the dimensions of grid cells are much larger than typical misalignment distances, the corresponding grid cells of different images will largely overlap the same object regions; (2) Products tend to have a sparse set of surface colors, and the pixel reordering will help to align these colors between the corresponding grid cells of different images, which is sufficient for measuring color-based consistency; (3) With this representation, the SILR loss is empirically found to be more sensitive to the presence of highlights or albedo distortions than to slight misalignment, as illustrated in Figure 6.3 for diffuse chromaticity.

6.4.4 Joint finetuning by contrastive loss

After training Highlight-Net and Shading-Net individually, we adopt a novel contrastive loss to finetune the entire network in an end-to-end manner. Our approach is based on the observation that intrinsic image decomposition can be better performed after highlights have been separated from input images. Related observations have been made in other recent works. For example, Ma et al. [65] mention that their method cannot handle specularity well, and this limitation will be addressed in future work. Also, Shi et al. [99] discuss that the multiplicative intrinsic image decomposition model, $I_d = A \cdot S$, cannot adequately account for additive highlight components.

Based on this observation, we define a contrastive loss. As indicated in Figure 6.4, our low-rank loss on the albedo layers of multiple images is $Loss_1$ if highlights are removed

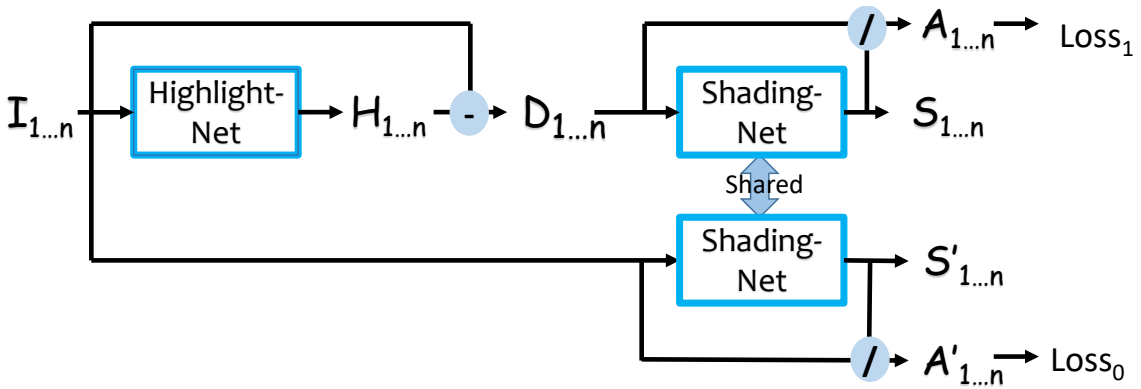


Figure 6.4: Network structure for joint finetuning by contrastive loss.

from the input images following the image formation model $I = A \cdot S + H$. In another branch, we compute the low-rank loss on albedo layers as $Loss_0$, where the input images are decomposed by Shading-Net directly following the image formation model $I = A \cdot S$. The contrastive loss is defined as:

$$Loss_{ct} = Loss_1 - Loss_0. \tag{6.4}$$

Intuitively, the contrastive loss is designed to maximize the distance between $Loss_1$ and $Loss_0$ (where $Loss_{ct}$ is negative), so as to force Highlight-Net to improve its highlight separation and thus decrease $Loss_1$ relative to $Loss_0$. Both subnets can be finetuned by this loss. In our experiments, we found that using $Loss_{ct}$ alone will lead to increases of both $Loss_1$ and $Loss_0$, as this increases their difference as well. To avoid this degenerate case, we add $\omega Loss_1$ as a regularization, such that the joint finetuning loss becomes $Loss = Loss_{ct} + \omega Loss_1$, where ω is set to 1.0 in our implementation.

After these three training phases, our network shown in Figure 7.1 is able to separate the highlight, diffuse, albedo, and shading layers of a test image.

6.5 Evaluations

Since previous works generally address highlight separation or intrinsic image estimation but not both, we evaluate our method on these two tasks separately. Comparisons to several techniques are presented quantitatively and qualitatively on both synthetic and real data. Ablations are also presented to show the robustness of our color distribution loss to misaligned images.

6.5.1 Evaluations of highlight separation

We compare with [106], [117], [95], [26] and [99] on highlight separation using synthetic data from the ShapeNet Intrinsic Dataset [99]. Since no standard real-image dataset ex-

Method	Synthetic		Real	
	MSE	DSSIM	MSE	DSSIM
Tan [106]	0.0155	0.0616	0.0173	0.0368
Yang [117]	0.0053	0.0336	0.0043	0.0162
Shen [95]	0.0059	0.0338	0.0047	0.0163
Shi [99]	0.0063	0.0526	0.0063	0.0237
Guo [26]	0.0028*	0.0208*	0.0045	0.0145
Ours	0.0016	0.0159	0.0036	0.0139

Table 6.1: Quantitative highlight separation comparison on the synthetic ShapeNet Intrinsic Dataset and on a real-image dataset. The lowest errors are highlighted in red, and the second lowest are in blue. Guo [26] is tested on only 50 of the 500 synthetic data in total, with the results marked by *, since we needed the authors to process our images.

ists for evaluating highlight separation, we captured a dataset consisting of 20 ordinary objects/scenes with ground truth by cross polarization, and also test on this.

Evaluation on synthetic dataset

On the ShapeNet Intrinsic Dataset, we randomly select 500 images covering a wide range of objects and materials to form the test set. Table 6.1 summarizes the MSE and DSSIM scores of different methods, which measure pixelwise difference and structural dissimilarities, respectively.

Examples for visual comparison are shown in Figure 6.5. Earlier methods [95, 106, 117] often assume white illumination and can estimate only a grayscale highlight layer, even when the lighting is not white. Moreover, they cannot deal with saturated regions well. A recent method [26] handles saturated highlight regions better with a low-rank and sparse decomposition. However, it still cannot recover correct diffuse color at saturated regions where its assumed dichromatic model is violated, leading to artifacts in diffuse layers. The CNN-based method of [99] can learn from various training data composed of different surface materials, but it still does not handle saturation well. By comparison, our method succeeds in predicting highlight colors and generates reasonable diffuse layers even for saturated regions.

Evaluation on real data

To evaluate performance on real images, we captured a dataset with ground truth by cross-polarization in a lab environment.

Table 6.1 shows the MSE and DSSIM of different methods on this dataset. Figure 6.6 shows qualitative comparisons on example images. Our method is found to generate highlight and diffuse layers closest to the ground truth. Our recovered highlights are of correct color even in saturated regions. Our method successfully recovers the surface colors in the diffuse layers, while the other methods tend to leave black artifacts at saturated regions.

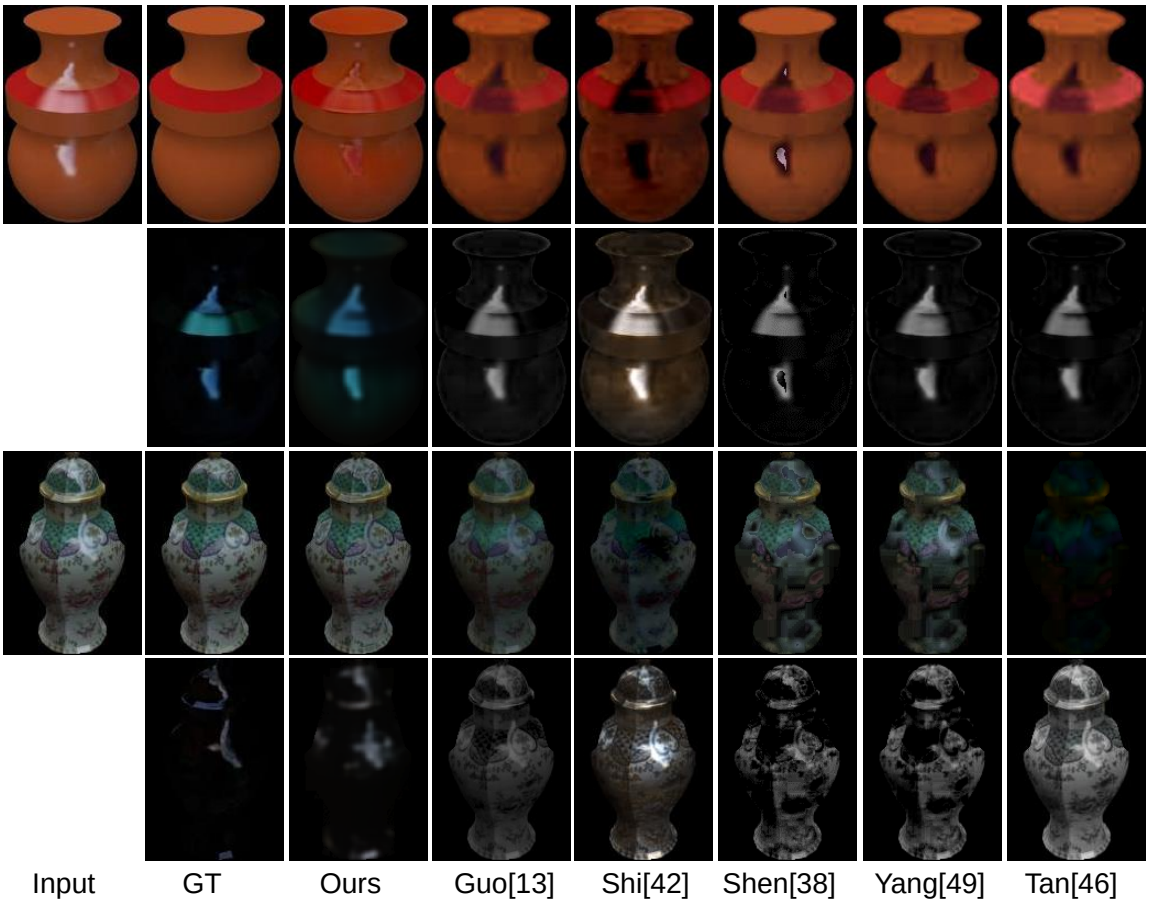


Figure 6.5: Visual comparisons of highlight separation on the ShapeNet Intrinsic Dataset. For each example, the top row shows the input image and separated diffuse layers, and the bottom row exhibits the separated highlight layers. GT denotes ground truth.

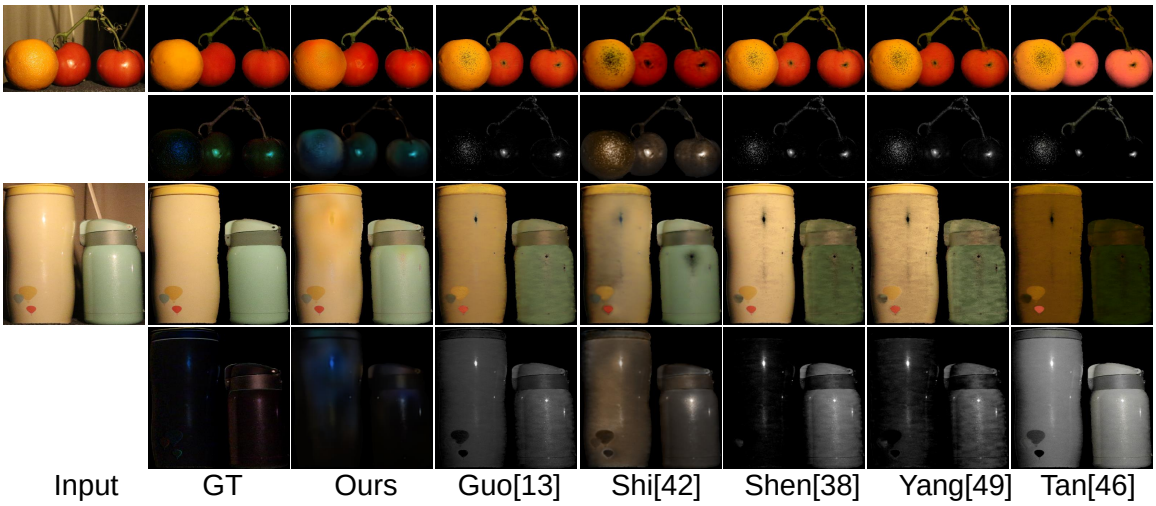


Figure 6.6: Visual comparisons of highlight extraction on real images. For each example, the top row shows the input image and separated diffuse layers, and the bottom row exhibits the separated highlight layers.

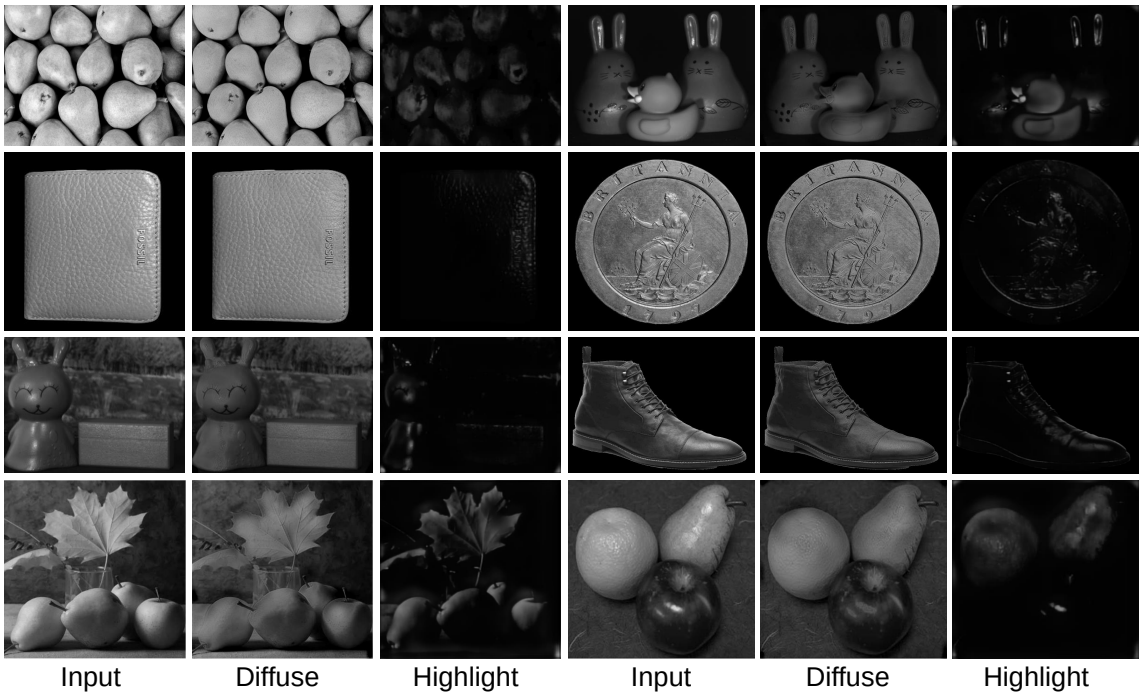


Figure 6.7: Qualitative results of highlight separation on grayscale images.

Highlight separation for grayscale images

Other than highlight extraction of color images, one advantage of CNN-based methods is that the CNNs trained from color images can also be used on grayscale images, in contrast to conventional methods which rely on color analysis based on the dichromatic model and/or piecewise diffuse colors.

For tests on grayscale images, we obtain the predicted highlight in grayscale by averaging its values over the three channels. Subtracting the grayscale highlight layer from the input image gives the diffuse layer. Qualitative results on real images are shown in Figure 6.7.

6.5.2 Evaluations on intrinsic image decomposition

In this subsection, we compare our network to different intrinsic image decomposition methods including SIRFS [4], DI [71], Shi et al. [99], and Li et al. [58].

Evaluation on the ShapeNet Intrinsic Dataset

Similar to the evaluation of highlight separation, we use MSE and DSSIM to measure the results from different methods. Note that while DSSIM is insensitive to scale changes, MSE does depend on scale. So in computing MSE, we first solve for a global rescaling factor that would most closely match the estimated albedo to the ground truth in order to resolve the scale ambiguity between albedo and shading. After that, we compute the MSE of the

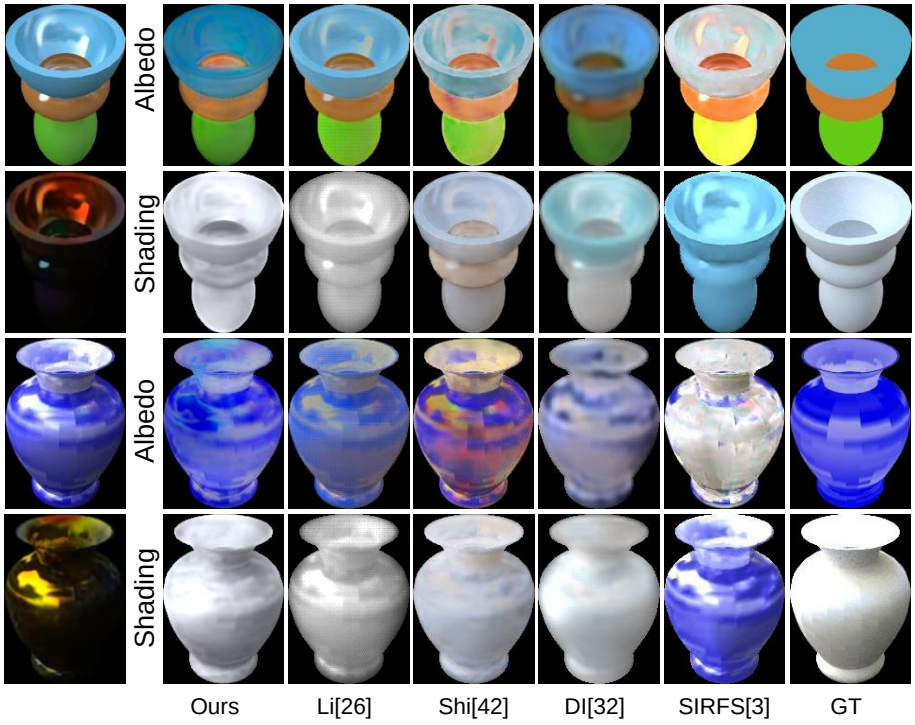


Figure 6.8: Visual comparisons of intrinsic image decomposition on testing data from the ShapeNet Intrinsic Dataset. For the first column, odd rows show input image and even rows show our separated highlights.

rescaled albedo. Table 6.2 summarizes the quantitative results and shows the relatively strong performance of our method.

Some qualitative comparisons are shown in Figure 6.8. SIRFS [4], which is based on scene priors, fails on non-Lambertian objects. The learning-based method DI [71] trained on synthetic diffuse scenes exhibits similar problems. The method by Shi et al. [99] performs better than previous methods on non-Lambertian objects. One reason is that, like our method, it explicitly models highlights, in contrast to other methods [4, 58, 71] which consequently have artifacts in the albedo layer on highlight regions. Another reason is because it is trained on the ShapeNet Intrinsic training split with 80% of the whole dataset. In comparison, our method is pretrained on a very small amount (1.1%) of the ShapeNet dataset to obtain a good network initialization, and is finetuned on a large amount of real data. Despite this, it still performs well on synthetic ShapeNet images. Since our Shading-Net solves for shading and then computes albedo using the image formation model $I_d = A \cdot S$, it generates high resolution albedo maps with texture details, whereas many networks that directly solve for albedo will obtain blurred results due to feature map downsampling in the network.

	SIRFS [4]	DI [71]	Shi [99]	Li [58]	Ours
MSE(A)	0.0081	0.0086	0.0068	0.0066	0.0054
DSSIM(A)	0.0636	0.0590	0.0565	0.0541	0.0436
MSE(S)	0.0066	0.0047	0.0023	0.0063	0.0045
DSSIM(S)	0.0785	0.0765	0.0691	0.0812	0.0686

Table 6.2: Quantitative intrinsic image comparison on synthetic data from ShapeNet Intrinsic Dataset. The lowest errors are highlighted in red and the second lowest are in blue.

Evaluation on the MIT intrinsics dataset

We also test our method on the MIT intrinsic image dataset [25], which contains real images under white illumination with mostly Lambertian objects. For this evaluation, we use Shading-Net alone, because highlights are merged into the shading in the ground truth decomposition, modeled as $I = A \cdot S$. Since highlights are not correctly represented in this model, the resulting shading contains distortions due to highlight, which we aim to approximate by using Shading-Net instead of our full system to recover shading. Despite this less-than-ideal scenario for our method, it still produces reasonable results.

Table 6.3 summarizes the results of different methods. Previous learning based methods, e.g. [99], generally have problems on this dataset due to the domain shift from synthetic image training to real image testing. Compared to such methods, our Shading-Net has the advantage of being trainable on multiview sets of real images. SIRFS obtains the best results on this dataset. As noted in previous work [99], SIRFS is built on priors that match the MIT dataset well (e.g. mostly Lambertian surfaces, white lighting). However, such priors cause SIRFS to be less effective on non-Lambertian objects, as seen in the ShapeNet Intrinsic Dataset experiments.

In the table, we also show results of our Shading-Net and those of Shi et al. [99] with finetuning on the MIT training split used by DI [70]. Due to our network structure, we only use ground truth albedo in training and do not take advantage of ground truth shading. Our shading is computed directly from the additional hard constraint $I = A \cdot S$ once albedo is fixed.

Qualitative comparison examples are shown in Figure 6.9. The recovered albedo maps from our method have the highest resolution and most texture detail, while other learning-based methods tend to obtain blurred results.

Evaluation on IIW dataset

As shown in Figure 6.10, we also test our method on the IIW dataset, which contains many daily scene photos. Although our method is trained on object-centric images, it also shows reasonable results on scene photos and generates results comparable to Li et al. [58], which is the most recent method trained on scene images. Our albedo results maintain the same resolution as the input images, and preserve detailed textures like the carpet pattern in the

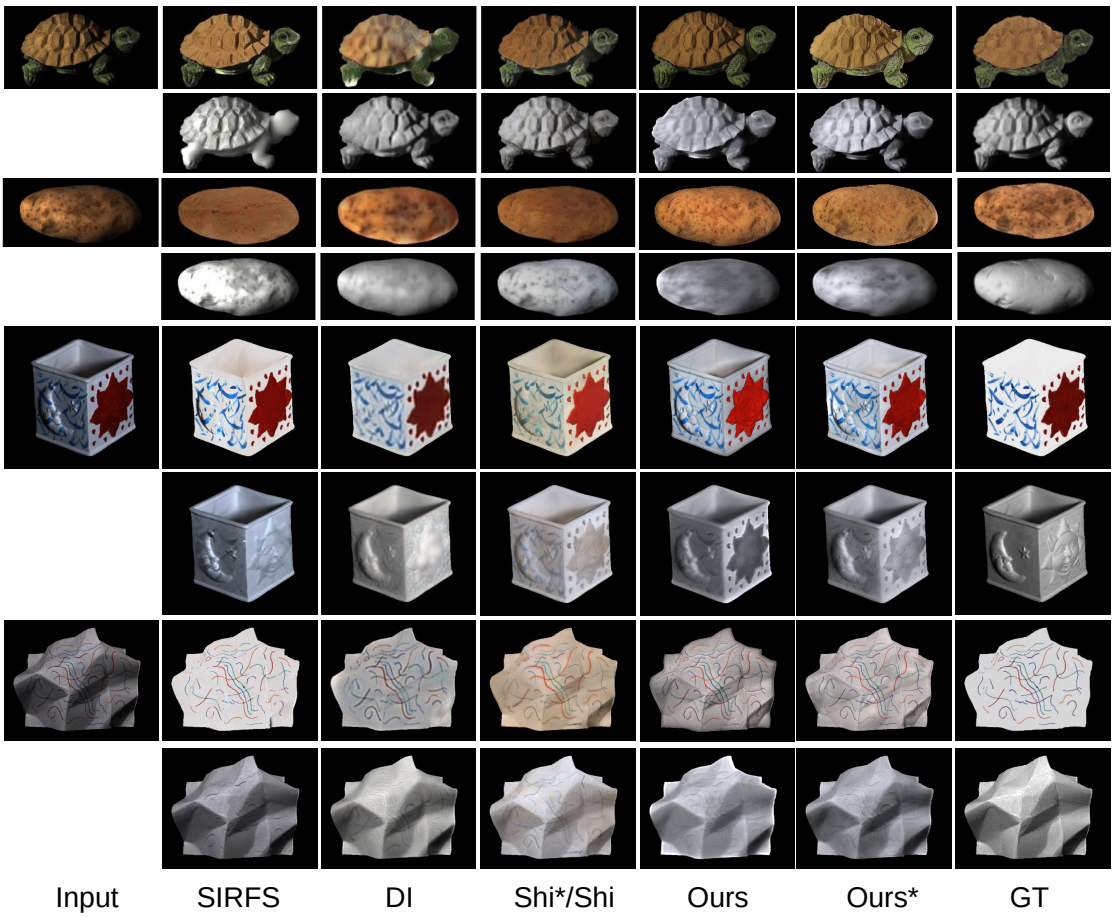


Figure 6.9: Visual comparisons of intrinsic image results on the MIT intrinsics dataset. Ours denotes our Shading-Net without finetuning on MIT, and ours* denotes our Shading-Net after finetuning on MIT. Since the model of Shi* is not available, the top two examples shown the results of Shi* given in their paper, and the bottom two examples show their results before finetuning on MIT. SIRFS denotes [4] and DI denotes [71].

Method	Training set	LMSE		MSE	
		albedo	shading	albedo	shading
SIRFS [3]	MIT	0.0416	0.0168	0.0147	0.0083
DI [71]	MIT+ST	0.0585	0.0295	0.0277	0.0154
Shi [99]	SN	0.0752	0.0318	0.0468	0.0194
Shi* [99]	SN+MIT	0.0503	0.0240	0.0278	0.0126
Ours	SN+CP	0.0520	0.0416	0.0365	0.0272
Ours*	SN+CP+MIT	0.0476	0.0284	0.0274	0.0145

Table 6.3: Quantitative intrinsic image decomposition evaluation on MIT intrinsic dataset. For the training set, ST denotes ResynthSintel dataset [71], SN denotes ShapeNet intrinsics dataset, and CP denotes our Customer Photos Dataset. * indicates finetuning on the MIT split used in DI.

top example. Compared to Shi et al. [99], which is also trained on object-centric data, our method is better able to handle real images, thanks to its ability to train on (multi-view) real image sets.

6.5.3 Robustness to misalignment

To examine the importance of our color distribution loss in dealing with misalignment, we compare to the results of our network when using a pixel-to-pixel low-rank loss instead, while training on the Customer Photos Dataset. For highlight extraction, the corresponding MSE and DSSIM by this network is 0.0020 and 0.0166 for the ShapeNet Intrinsics Dataset and 0.0041 and 0.0149 for real images, respectively, which are larger than the errors when using the color distribution loss, shown in Table 6.1. For intrinsic image decomposition, the performance difference is even greater, with MSE and DSSIM on the ShapeNet Intrinsic Dataset of 0.0067 and 0.0460 for albedo, and 0.0087 and 0.0774 for shading, compared to the values in Table 6.2. This illustrates the sensitivity to image misalignment of pixel-to-pixel loss functions, as used in [58, 65]. Qualitative results are shown in Figure 6.11.

6.5.4 Without pretraining on synthetic data

Our model is pretrained on a small amount of synthetic data to bootstrap the unsupervised phases. Here, we examine training the network from scratch with only the unsupervised finetuning. As shown in Figure 6.12, reasonable highlight extraction and intrinsic image decomposition can be achieved even without pretraining on synthetic data. We evaluated the fully unsupervised network on ShapeNet Intrinsics Dataset and obtained an MSE and DSSIM for highlight extraction of 0.0041 and 0.0227, compared to the leftmost two columns of Table 6.1. The MSE and DSSIM on real images are 0.0057 and 0.0199, compared to the rightmost two columns of Table 6.1, which are comparable to previous methods. For intrinsic image decomposition, the MSE and DSSIM are 0.0067 and 0.0527 for albedo, and 0.0059 and 0.0808 for shading, compared to the corresponding values 0.0054 and 0.0436 for albedo, and 0.0045 and 0.0686 for shading in Table 6.2. This indicates that there is

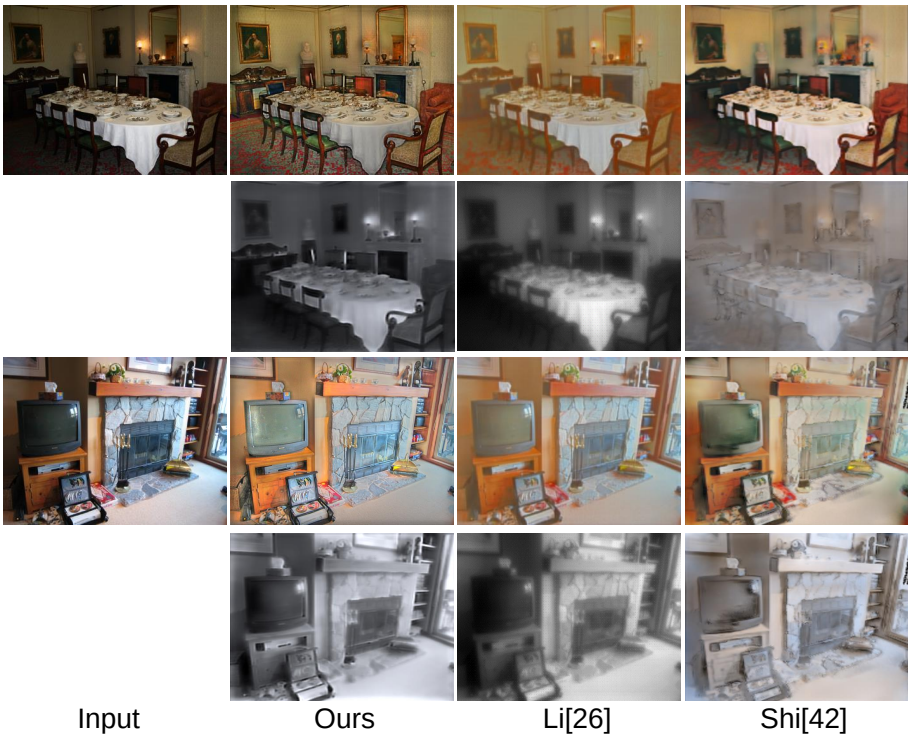


Figure 6.10: Qualitative comparisons on scene images from the IIW dataset. The albedo layers are shown on odd rows, and shadings at even rows.

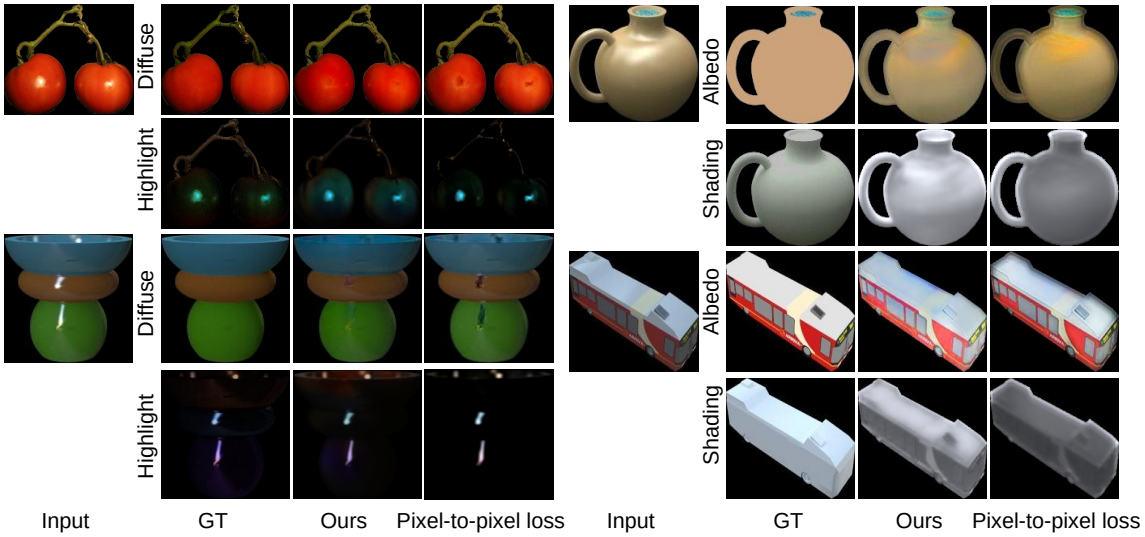


Figure 6.11: Visual comparisons between our color distribution loss and the pixel-to-pixel low-rank loss in handling misalignment of training images. The top two examples show comparisons on highlight separation, and the bottom two show comparisons on intrinsic image decomposition.

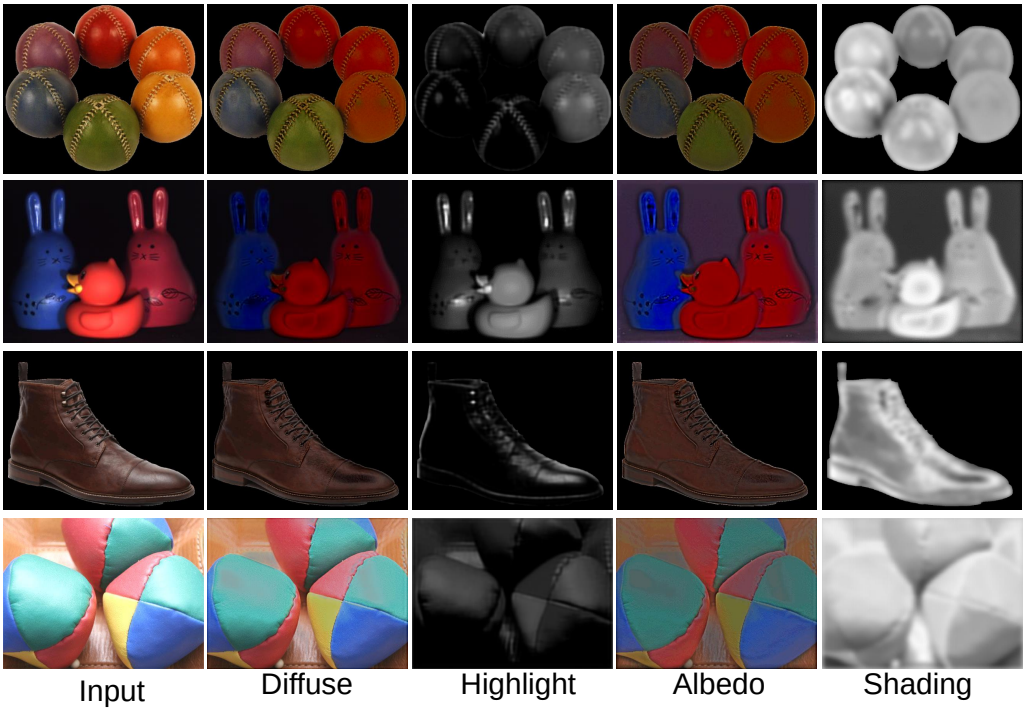


Figure 6.12: Qualitative results on real images for a fully unsupervised version of our network, without pretraining on synthetic data.

some moderate dropoff without the pretraining on synthetic data, but the performance nevertheless compares well to previous techniques.

6.5.5 Evaluation of end-to-end separations

To evaluate the performance of our end-to-end network, we separate real images into highlight, diffuse, albedo, and shading layers all at once, assuming the image formation model $I = H + A \cdot S$. For comparison, we combine the methods by Yang et al. [117] for highlight separation and Shi et al. [99] for intrinsic image decomposition, which have state-of-the-art performance for these tasks. The highlight in the input image is first computed by the method by Yang et al. [117] and separated from the input image. The remaining diffuse image is then decomposed into albedo and shading by the method of Shi et al. [99]. As shown in Figure 6.13, our method shows better performance than the combination of Yang et al. [117] and Shi et al. [99], and performs well even on scenes with strong highlights and complicated textures.

6.6 Conclusion

We proposed an end-to-end network to solve highlight separation and intrinsic image decomposition together. Our network is able to leverage multi-view object-centric image sets,

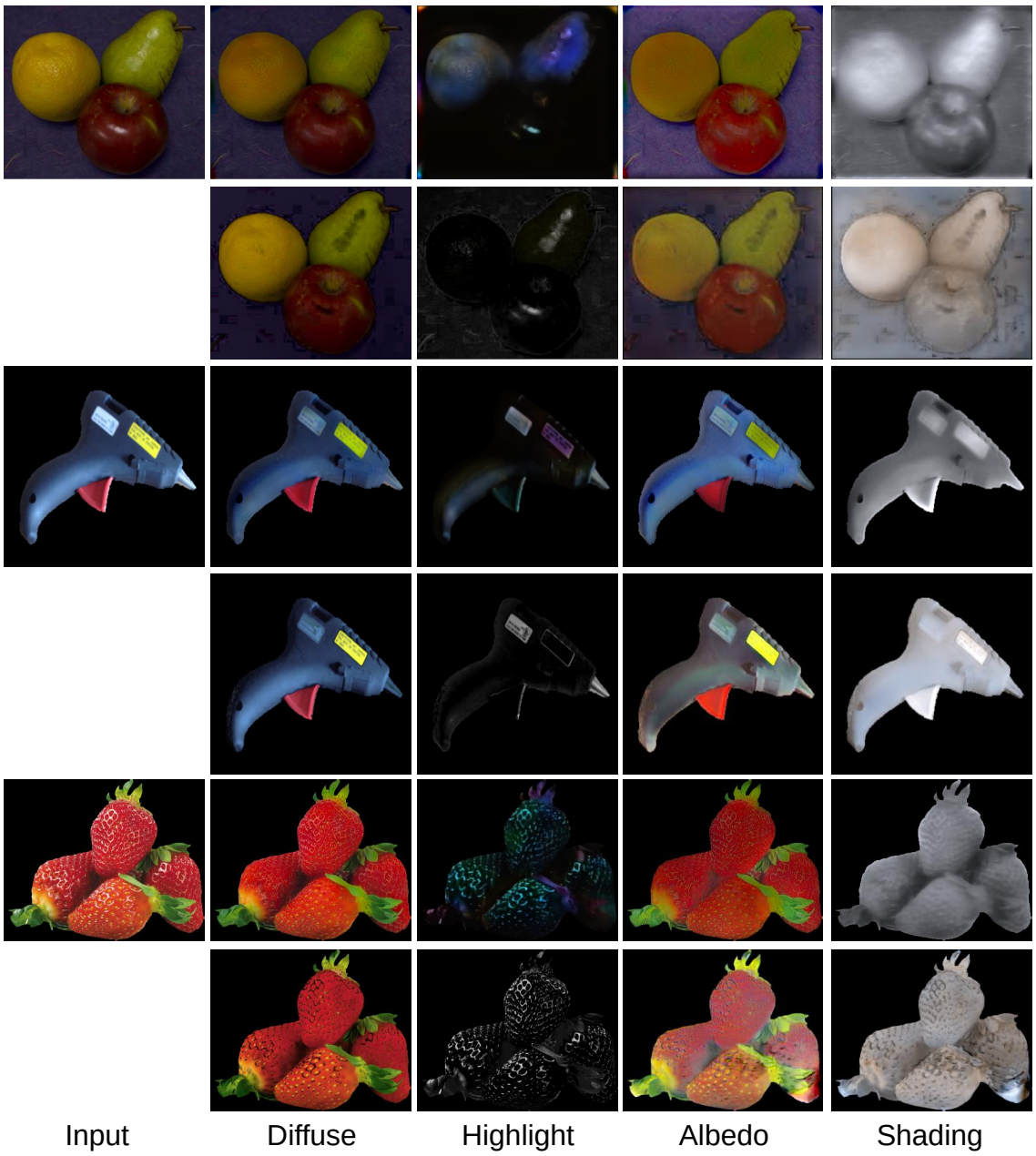


Figure 6.13: Qualitative comparisons on real images. We compare our end-to-end separation of highlight, diffuse, albedo and shading layers to the combination of Yang et al. [117] for highlight separation and Shi et al. [99] for intrinsic image decomposition, which have the second best performance in quantitative evaluations. The odd rows are our results, and even rows are results of Yang et al. [117] and Shi et al. [99].

such as our Customer Product Photos Dataset, for unsupervised training via a proposed color distribution loss that is robust to misaligned data. This loss can readily be adapted for other tasks that are sensitive to misalignment.

Chapter 7

Application of Illumination Estimation from Separated Image Layers

In this chapter, we demonstrate the application of illumination estimation from separated face layers. For face images, we take advantages of existed face shape estimation methods, and face BRDF database. This is a follow-up work of Chapter 5. For general objects in Section 6, the problem would be more complex due to unknown BRDF of general objects, so we will leave it as our future work.

7.1 Introduction

Spicing up selfies by inserting virtual hats, sunglasses or toys has become easy to do with mobile augmented reality (AR) apps like *Snapchat* [101]. But while the entertainment value of mobile AR is evident, it is just as clear to see that the generated results are usually far from realistic. A major reason is that virtual objects are typically not rendered under the same illumination conditions as in the imaged scene, which leads to inconsistency in appearance between the object and its background. For high photorealism in AR, it is thus necessary to estimate the illumination in the image, and then use this estimate to render the inserted object compatibly with its surroundings.

Illumination estimation from a single image is a challenging problem because lighting is intertwined with geometry and reflectance in the appearance of a scene. To make this problem more manageable, most methods assume the geometry and/or reflectance to be known [57, 62, 76, 81, 84, 91, 92, 109]. Such knowledge is generally unavailable in practice; however, there exist priors about the geometry and reflectance properties of human faces that have been exploited for illumination estimation [36, 42, 54, 86]. Faces are a common occurrence in photographs and are the focus of many mobile AR applications. The previous works on face-based illumination estimation consider reflections to be diffuse and estimate only the low-frequency component of the environment lighting, as diffuse reflectance acts

as a low-pass filter on the reflected illumination [84]. However, a low-frequency lighting estimate often does not provide the level of detail needed to accurately depict virtual objects, especially those with shiny surfaces.

In addressing this problem, we consider the parallels between human faces and mirrored spheres, which are conventionally used as lighting probes for acquiring ground truth illumination. What makes a mirrored sphere ideal for illumination recovery is its perfectly sharp specular reflections over a full range of known surface normals. Rays can be traced from the camera's sensor to the sphere and then to the surrounding environment to obtain a complete environment map that includes lighting from all directions and over all frequencies, subject to camera resolution. We observe that faces share these favorable properties to a large degree. They produce fairly sharp specular reflections (highlights) over its surface because of the oil content in skin. Moreover, faces cover a broad range of surface normals, and there exist various methods for recovering face geometry from a single image [8, 36, 86, 93, 116]. Unlike mirrored spheres, the specular reflections of faces are not perfectly sharp and are mixed with diffuse reflection. In this chapter, we propose a method for dealing with these differences to facilitate the use of faces as light probes.

As described in Chapter 5, we first present a deep neural network for separating specular highlights from diffuse reflections in face images by unsupervised training on a large-scale real-image database. With the extracted specular highlights, we then recover the environment illumination. This recovery is inspired by the frequency domain analysis of reflectance in [84], which concludes that reflected light is a convolved version of the environment map. Thus, we estimate illumination through a deconvolution of the specular reflection, in which the deconvolution kernel is determined from prior knowledge of face material properties. This approach enables recovery of higher-frequency details in the environment lighting.

This method is validated through experimental comparisons to previous techniques for illumination estimation. Greater precision is obtained over a variety of both indoor and outdoor scenes. We additionally show that the 3D positions of local point lights can be estimated using this method, by triangulating the light source positions from the environment maps of multiple faces in an image. With this 3D lighting information, the spatially variant illumination throughout a scene can be obtained. Recovering the detailed illumination in a scene not only benefits AR applications but also can promote scene understanding in general.

7.2 Overview

For a single image input, the network described in Chapter 5 takes an input image and estimates its highlight layer. Together with reconstructed facial geometry estimated by a previous method, the extracted highlights are used to obtain an initial environment map, by tracing the highlight reflections back towards the scene (Section 7.3). This initial map

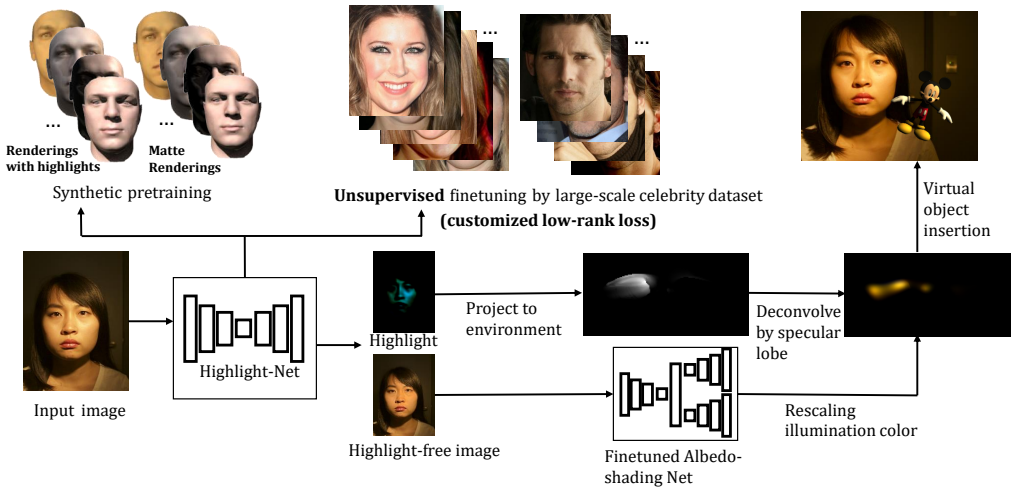


Figure 7.1: Overview of our method. An input image is first separated into its highlight and diffuse layers. We trace the highlight reflections back to the scene according to facial geometry to recover a non-parametric environment map. A diffuse layer obtained through intrinsic component separation [71] is used to determine illumination color. With the estimated environment map, virtual objects can be inserted into the input image with consistent lighting.

is blurred due to the band-limiting effects of surface reflectance [84]. To mitigate this blur, our method performs deconvolution on the environment map using kernels determined from facial reflectance statistics (Section 7.4). Details about rescaling illumination color to deal with the color saturation in highlight layers are described in Section 7.5. Furthermore, we also demonstrated the estimation of spatially variant illumination from multiple faces in the a single photo (Section 7.6). Comparisons and evaluations to previous techniques show the state-of-the-art performance of this approach on a variety of indoor and outdoor scenes (Section 7.7).

7.3 Environment map initialization

The specular reflections of a mirror are ideal for illumination estimation, because the observed highlights can be exactly traced back to the environment map when surface normals are known. This exact tracing is possible because a highlight reflection is directed along a single reflection direction R that mirrors the incident lighting direction L about the surface normal N, as shown on the left side of Figure 7.2. This raytracing approach is widely used to capture environment maps with mirrored spheres in computer graphics applications.

For the specular reflections of a rough surface like human skin, the light energy is instead tightly distributed around the mirror reflection direction, as illustrated on the right side of Figure 7.2. This specular lobe can be approximated by the specular term of the Phong

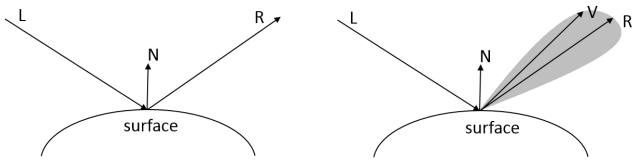
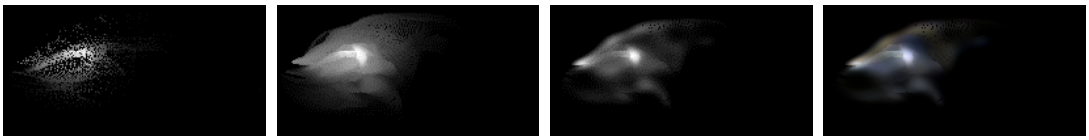


Figure 7.2: Left: Mirror reflection. Right: Specular reflection of a rough surface.



(a) (b) (c) (d)

Figure 7.3: Intermediate results of illumination estimation. (a) Traced environment map by forward warping; (b) Traced environment map by inverse warping; (c) Map after deconvolution; (d) Final environment map after illumination color rescaling.

model [82] as

$$I_s = k_s(R \cdot V)^\alpha, \qquad R = 2(L \cdot N)N - L \tag{7.1}$$

where k_s denotes the specular albedo, V is the viewing direction, and α represents the surface roughness. We specifically choose to use the Phong model to take advantage of statistics that have been compiled for it, as described later.

As rigorously derived in [84], reflection can be expressed as the environment map convolved with the surface BRDF (bidirectional reflectance distribution function), e.g., the model in Equation 7.1. Therefore, if we trace the highlight component of a face back toward the scene, we obtain a convolved version of the environment map, where the convolution kernel is determined by the specular reflectance lobe. With surface normals computed using a single-image face reconstruction algorithm [116], our method performs this tracing to recover an initial environment map, such as that exhibited in Figure 7.3 (a).

Due to limited image resolution, the surface normals on a face are sparsely sampled, and an environment map obtained by directly tracing the highlight component would be sparse as well, as shown in Figure 7.3 (a). To avoid this problem, we employ inverse image warping where for each pixel p in the environment map, trace back to the face to get its corresponding normal N_p and use the available face normals nearest to N_p to interpolate a highlight value of N_p. In this way, we avoid the holes and overlaps caused by directly tracing (i.e., forward warping) highlights to the environment map. The result of this inverse warping is illustrated in Figure 7.3 (b).

7.4 Deconvolution by the specular lobe

Next, we use the specular lobe to deconvolve the filtered environment map. This deconvolution is applied in the spherical domain, rather than in the spatial domain parameterized by latitude and longitude which would introduce geometric distortions.

Consider the deconvolution kernel K_x centered at a point $\mathbf{x} = (\theta_x, \phi_y)$ on the environment map. At a nearby point $\mathbf{y} = (\theta_y, \phi_y)$, the value of K_x is

$$K_x(\mathbf{y}) = k_s^x (L_y \cdot L_x)^{\alpha_x} \tag{7.2}$$

where L_x and L_y are 3D unit vectors that point from the sphere center toward $\mathbf{x}$ and $\mathbf{y}$, respectively. The terms α_x and k_s^x denote the surface roughness and specular albedo at $\mathbf{x}$.

To determine α_x and k_s^x for each pixel in the environment map, we use statistics from the MERL/ETH Skin Reflectance Database [113]. In these statistics, faces are categorized by skin type, and every face is divided into ten regions, each with its own mean specular albedo and roughness because of differences in skin properties, e.g., the forehead and nose being relatively more oily. Using the mean albedo and roughness value of each face region for the face's skin type[1], our method performs deconvolution by the Richardson-Lucy algorithm [64, 87]. Figure 7.3 (c) shows an environment map after deconvolution.

7.5 Rescaling illumination color

The brightness of highlight reflections often leads to saturated pixels, which have color values clipped at the maximum image intensity. As a result, the highlight intensity in these color channels may be underestimated. This problem is illustrated in Figure 7.4, where the predicted highlight layer appears blue because the light energy in the red and green channels is not fully recorded in the input image. To address this issue, we take advantage of diffuse shading, which is generally free of saturation and indicative of illumination color.

Diffuse reflection (i.e., the diffuse layer) is the product of albedo and diffuse shading, and the diffuse shading can be extracted from the diffuse layer through intrinsic image decomposition. To accomplish this decomposition, we finetune the intrinsic image network from [71] using synthetic face images to improve the network's effectiveness on faces. Specifically, 10,000 face images were synthesized from 50 face shapes randomly generated using the Basel Face Model [80], three different skin tones, diffuse reflectance, and environment maps randomly selected from 100 indoor and 100 outdoor real HDR environment maps. Adding this Albedo-Shading Net to our system as shown in Figure 5.2 (b) yields a highlight layer, albedo layer, and diffuse shading layer from an input face.

[1]Skin type is determined by the closest mean albedo to the mean value of the face's albedo layer. Extraction of the face's albedo layer is described in Section 7.5.

(a) (b) (c) (d)

Figure 7.4: (a) Input photo; (b) Automatically cropped face region by landmarks [124] (network input); (c) predicted highlight layer (scaled by 2); (d) highlight removal result.

With the diffuse shading layer, we recolor the highlight layer H extracted via Highlight-Net by rescaling its channels. When the blue channel is not saturated, its value is correct and the other channels are rescaled relative to it as

$$[H'(r),\ H'(g),\ H'(b)] = [H(b) * c_d(r)/c_d(b),\ H(b) * c_d(g)/c_d(b),\ H(b)] \tag{7.3}$$

where c_d is the diffuse shading chromaticity. Rescaling can similarly be solved from the red or green channels if they are unsaturated. If all channels are saturated, we use the blue channel as it is likely to be the least underestimated based on common colors of illumination and skin. After recoloring the highlight layer, we compute its corresponding environment map following the procedure in Sections 7.3-7.4 to produce the final result, such as shown in Figure 7.3 (d).

7.6 Triangulating lights from multiple faces

In a scene where the light sources are nearby, the incoming light distribution can vary significantly at different locations. An advantage of our non-parametric illumination model is that when there are multiple faces in an image, we can recover this spatially variant illumination by inferring the environment map at each face and using them to triangulate the 3D light source positions.

As a simple scheme to demonstrate this idea, we first use a generic 3D face model (e.g., the Basel Face Model [80]) to solve for the 3D positions of each face in the camera's coordinate system, by matching 3D landmarks on the face model to 2D landmarks in the image using the method of [124]. Highlight-Net is then utilized to acquire the environment map at each of the faces. In the environment maps, strong light sources are detected as local maxima found through non-maximum suppression. To build correspondences among the lights detected from different faces, we first match them according to their colors. When there are multiple lights of the same color, their correspondence is determined by

	Diffuse Bunny					Glossy Bunny				
Relighting RMSE	Ours	[31]	[23]	[62]	[42]	Ours	[31]	[23]	[62]	[42]
Mean (outdoor)	**10.78**	18.13	\	21.20	17.77	**11.02**	18.28	\	21.63	18.28
Median (outdoor)	**9.38**	17.03	\	19.95	15.91	**9.74**	17.67	\	20.49	16.30
Mean (indoor)	**13.18**	\	29.25	25.40	20.52	**13.69**	\	29.71	25.92	21.01
Median (indoor)	**11.68**	\	25.99	25.38	19.22	**11.98**	\	26.53	25.91	19.75

Table 7.1: Illumination estimation on synthetic data.

(a) (b) (c) (d) (e)

Figure 7.5: Virtual object insertion results for indoor (first row) and outdoor (second row) scenes. (a) Photos with real object. Object insertion by (b) our method, (c) [23] for the first row and [31] for the second row, (d) [62], (e) [42].

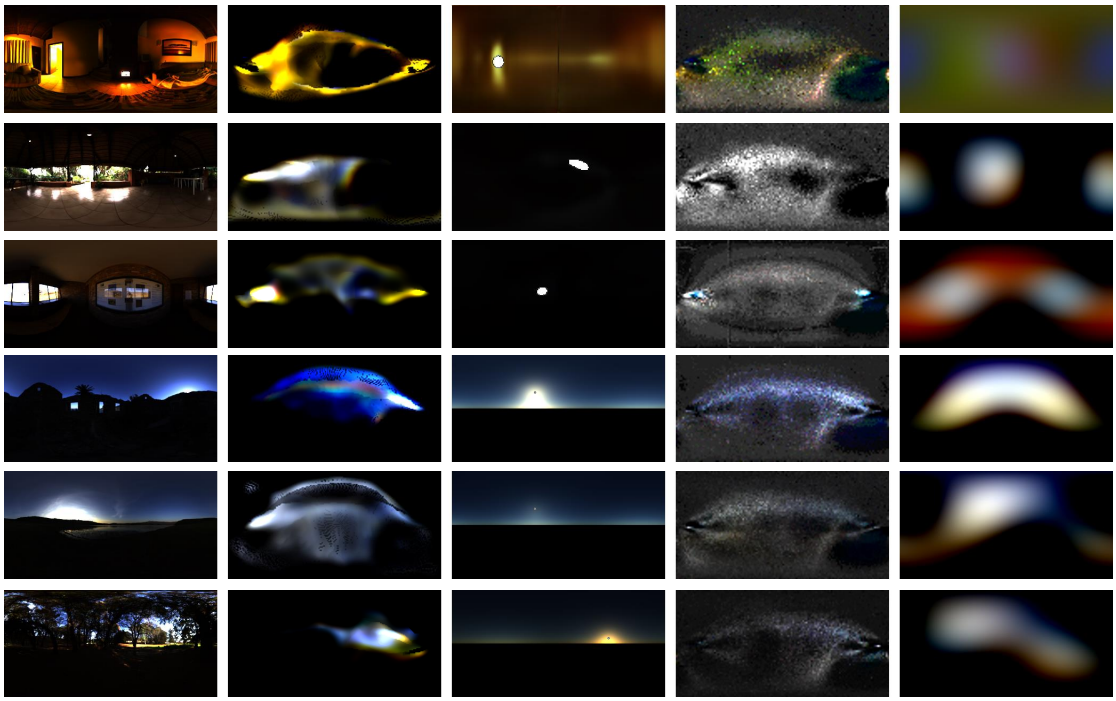
triangulating different combinations between two faces, with verification using a third face. In this way, the 3D light source positions can be recovered.

7.7 Experiments

7.7.1 Evaluation of illumination estimation

Following [31], we evaluate illumination estimation by examining the relighting errors of a Stanford bunny under predicted environment maps and the ground truth. The lighting estimation is performed on synthetic faces rendered into captured outdoor and indoor scenes and their recorded HDR environment maps. Results are computed for both a diffuse and a glossy Stanford bunny. The comparison methods include the following: our implementation

Figure 7.6: Object insertion results by our method.



(a) (b) (c) (d) (e)

Figure 7.7: Comparisons of selected indoor (top three rows) and outdoor (bottom three rows) data used in quantitative evaluation of illumination estimation. (a) Ground truth indoor environment maps, (b-e) indoor environment maps estimated by (b) our method, (c) [23], (d) [62] and (e) [42]. Total intensities of all environment maps are normalized to be the same.

of [42] which uses a face to recover spherical harmonics (SH) lighting up to second order under the assumption that the face is diffuse; downloaded code for [62] which estimates illumination and reflectance given known surface normals that we estimate using [116]; online demo code for [31] which is designed for outdoor images; and author-provided results for [23] which is intended for indoor images.

Visual comparisons on estimated environment maps are shown in Figure 7.7. The relighting errors are presented in Table 7.1 and Figure 7.9, selected visualizations are shown in Figure 7.10. Except for [31] and [23], the errors were computed for 500 environment maps estimated from five synthetic faces under 100 real HDR environment maps (50 indoor and 50 outdoor). Since [31] and [23] are respectively for outdoor and indoor scenes and are not trained on faces, their results are each computed from LDR crops from the center of the 50 indoor/outdoor environment maps. We found [31] and [23] to be generally less precise in estimating light source directions, especially when light sources are out-of-view in the input crops, but they still provide reasonable approximations. For [23], the estimates of high frequency lighting become less precise when the indoor environment is more complicated. The experiments indicate that [62] may be relatively sensitive to surface textures and imprecise

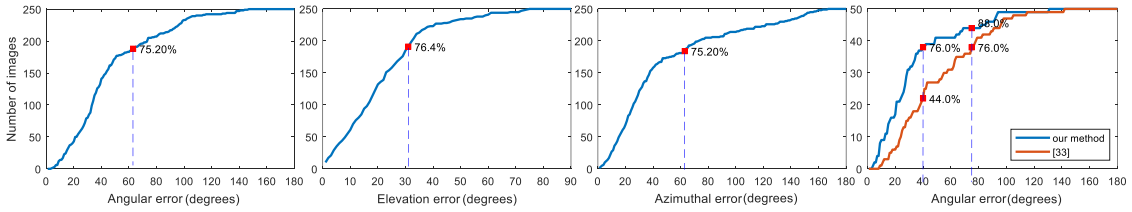


Figure 7.8: Evaluation of sun position estimation on outdoor testing data.

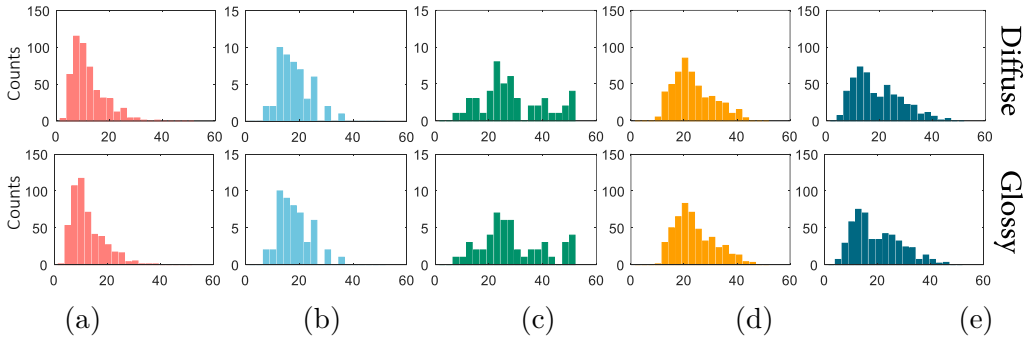


Figure 7.9: Relighting RMSE histograms of a diffuse/glossy Stanford bunny lit by illumination estimated by (a) our method, (b) [31] (for outdoor scenes), (c) [23] (for indoor scenes), (d) [62] and (e) [42] (spherical harmonics representation).

geometry in comparison to our method, which is purposely designed to deal with faces. For the Spherical Harmonics representation [42], estimates of a low-order SH model are seen to lack detail, and the estimated face albedo incorporates the illumination color, which leads to environment maps that are mostly white. Overall, the results indicate that our method provides the closest estimates to the ground truth.

To evaluate direction localization, we conducted an experiment on sun positions for outdoor scenes in Figure 7.8, we computed the centroid of the predicted environment maps as the sun position, in terms of cumulative distribution of images w.r.t. error level as done in [31], where the marked points indicate the error levels over more than 75% of the testing data.

We additionally conducted comparisons on virtual object insertion using estimated illumination, as shown in Figure 7.5. To aid in verification, we also show images that contain the actual physical object (an Android robot). In some cases such as the bottom of (c), lighting from the side is estimated as coming from farther behind, resulting in a shadowed appearance. Additional object insertion results are shown in Figure 7.6.

7.7.2 Demonstration of light source triangulation

Using the simple scheme described in Section 7.6, we demonstrate the triangulation of two local light sources from an image with three faces, shown in Figure 7.11 (a). The estimated

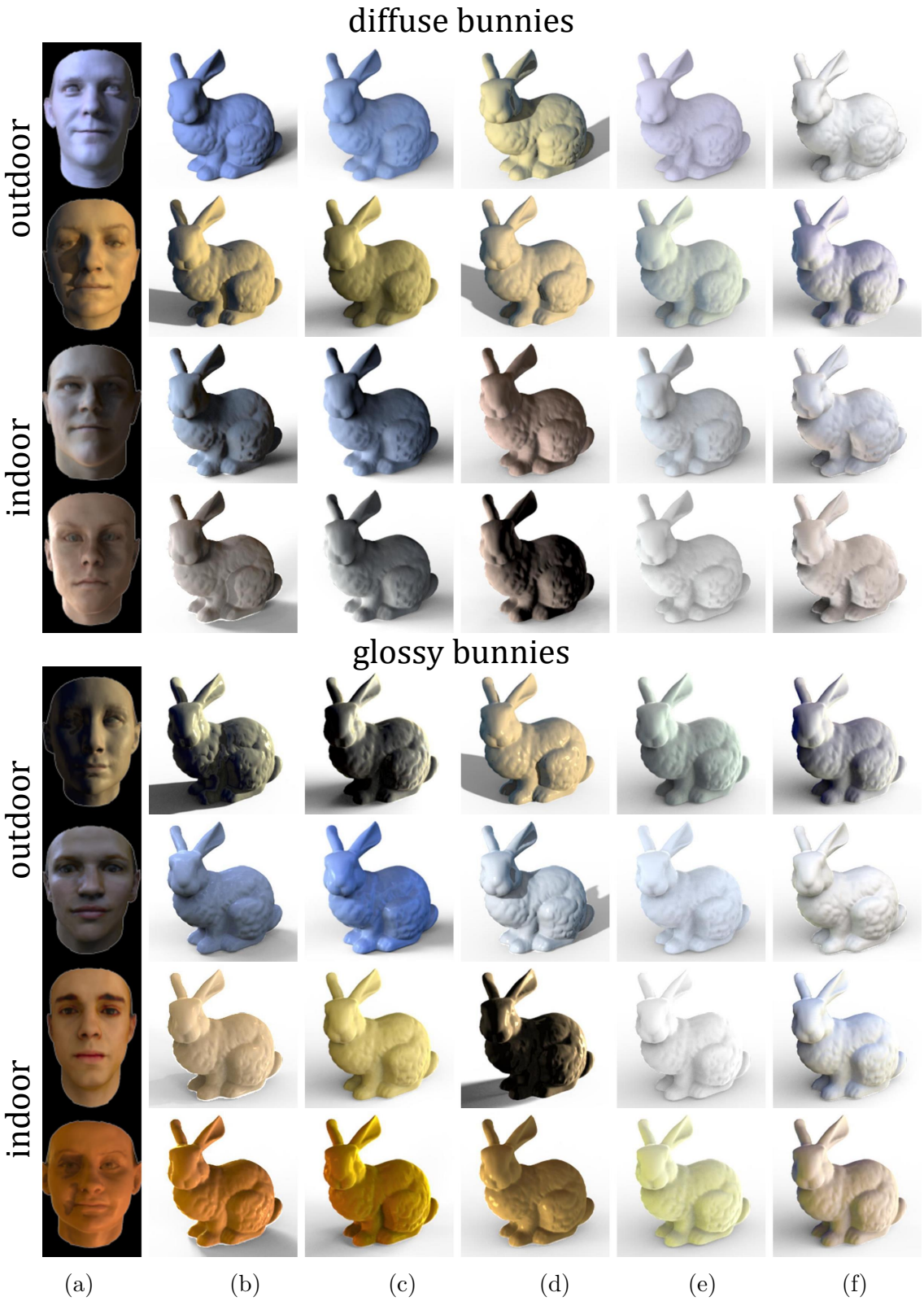
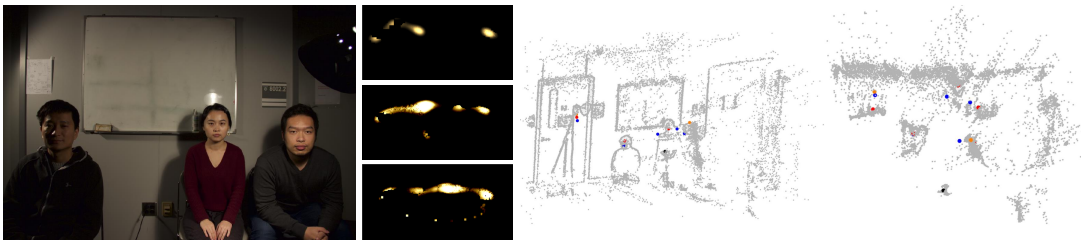


Figure 7.10: Comparisons of Stanford bunnies relit by estimated indoor and outdoor illuminations. (a) Input photo. (b) Bunnies under ground truth environment maps. (c-f) Bunnies relit by environment maps estimated by (c) our method, (d) [23], (e) [62] and (f) [42].



(a) (b) (c) (d)

Figure 7.11: (a) Input image with multiple faces; (b) their estimated environment maps (top to bottom are for faces from left to right); estimated 3D positions from (c) side view and (d) top view. Black dot: camera. Red dots: ground truth of faces and lights. Blue dots: estimated faces and lights. Orange dots: estimated lights using ground truth of face positions.

environment maps from the three faces are shown in Figure 7.11 (b). We triangulate the point lights from two of them, while using the third for validation. In order to provide a quantitative evaluation, we use the DSO SLAM system [20] to reconstruct the scene, including the faces and light sources. We manually mark the reconstructed faces and light sources in the 3D point clouds as ground truth. As shown in Figure 7.11 (c-d), the results of our method are close to this ground truth. The position errors are 0.19m, 0.44m and 0.29m for the faces from left to right, and 0.41m and 0.51m for the two lamps respectively. If the ground truth face positions are used, the position errors of the lamps are reduced to 0.20m and 0.49m, respectively.

7.8 Conclusion

We proposed a system for non-parametric illumination estimation based on an unsupervised finetuning approach for extracting highlight reflections from faces. In future work, we plan to examine more sophisticated schemes for recovering spatially variant illumination from the environment maps of multiple faces in an image. Using faces as lighting probes provides us with a better understanding of the surrounding environment not viewed by the camera, which can benefit a variety of vision applications.

Chapter 8

Conclusions

In this thesis, we propose a series of methods to solve three image layer separation problems and one application. In this chapter, we summarize the contributions, limitations and potential future works.

8.1 Contributions and limitations

Firstly, we would like to summarize most image layer separation problems into several main categories and discuss possible solutions for each category, which can provide more generalized tips for other image layer separation problems which are not solved in this thesis.

- **Opaque layers.** If an image layer is opaque, which means for each pixel, the observed intensity is equal to the intensity in one single layer, such as the fence layer in defencing problems, instead of the sum of multiple layers. These problems are easy to solve than semi-transparent layer separation, because the number of unknowns is significantly decreased. We can separate two layers by solving a binary mask, instead of an alpha mask for (partially) transparent layers. For problems belonging to this category, we can use traditional methods by handcrafted layers defining on distinctive features of each layer, as in Chapter 3.

- **Semi-transparent layers.** For semi-transparent layers, we have to solve the intensity of each layer at each pixel. These problems have a larger number of unknowns, and difficult to solve by defining a single objective function in optimization. Handcrafted priors usually cannot work for all scenarios, such as piecewise reflectance or smooth shading used in traditional intrinsic image decomposition, which does not work for objects of complex textures or complex geometries. Thus, for these problems, learning-based methods perform better, by learning priors automatically from a large amount of training data, or using priors to drive unsupervised or weakly-supervised training on unlabelled training data when ground truths are infeasible to capture, as in Chapter 4. We further divide the problems into two categories, which are semantic layers where

ground truths are easy to capture and physics-based layers where ground truths are difficult or unable to capture. For each category, we provide suggestions for possible solutions.

  - **Semantic layers.** Semantic image layers are those who have a clear semantic meaning, such as occluder layers such as fence, haze, glass reflection, or raindrops. Those occluder layers are caused by a physical object in the scene, such as fences, glasses and so on. For such layers, capturing real training data and ground truths are relatively easy. For example, the training image pairs with and without raindrops are captured by putting a glass with sprayed water in front of the lens in [83], and a similar dataset of image de-fencing is captured in [18]. Furthermore, for opaque layers such as fences, it is also possible to get a large amount of real training data with even less effort by inserting the several manually segmented fence layers into natural images/videos. By this way, the data generation is easy without causing a domain shift as synthetic data does.
  - **Physics-based layers.** Physics-based layers describe physical properties of surfaces, such as reflectance and shading layers in intrinsic image decomposition, or highlight layer in highlight separation. The ground truths of these layers are usually infeasible to capture. For such layers, it is very difficult to collect a large amount of labeled real data. Thus, weakly supervised or unsupervised training are preferred, and the loss functions can be defined by handcrafted priors or observations, such as the smoothness of reflectance or shading in intrinsic image decomposition. Furthermore, we can also use multiple unlabeled images to facilitate the training, such as the multiple aligned images used in our proposed training scheme in Chapter 4. Using handcrafted priors to define unsupervised loss functions in deep learning is a combination of traditional methods and DNN-based methods, as well as a combination of multi-image and single-image methods, where we use multiple images in training but only need a single image in testing.

We also summarize the individual contributions of each work presented in this thesis.

- **Video de-fencing.** In Chapter 3, we solve the fence segmentation in dynamic videos by a bottom-up framework, consisting of an initial segmentation step and a spatio-temporal refinement step. The main contributions of this work are that we propose to use optical flow as segmentation cues, to facilitate the separation of fence pixels and background pixels in pixel clustering. It shows that optical flow can also be used as features in many other video processing tasks. Furthermore, we design a feature to measure the orientation of fences pixels, which can be used in many other tasks like line detection, similar to [43]. At last, the spatio-temporal refinement step also

demonstrates that dense CRF can be used to improve video segmentation for keeping the temporal coherence, other than single image segmentation where it is originally applied on.

The limitation of this work is that the whole pipeline is not fully automatic yet, in the fence removal step, existing inpainting techniques are applied to each frame. If an object is occluded by fences in one frame, it is difficult to be restored in this frame. However, if a better fence inpainting method is proposed based on the information from all video frames, it is possible that occluded objects can be fully restored, based on those frames where they are visible.

- **Unsupervised training scheme for image layer separation by deep learning.** In Chapter 4, we propose an unsupervised training scheme for image layer separation by deep learning, which can be used to train networks from unlabeled real data which is numerous online. Later we apply this training scheme on face highlight separation and non-Lambertian intrinsic image decomposition. For both tasks, we make use of a large amount of unlabeled real data collected from the Internet, and after training, the performance of both tasks are better than networks trained on synthetic data only and other previous approaches. Qualitative and quantitative evaluations show the unsupervised training scheme does improve the performance of networks and the trained networks have better generalization to real-world scenes, unlike previous works trained on synthetic datasets. This training scheme can be applied to many other tasks, where such consistencies exist over images. Furthermore, in order to deal with the misalignment in the Customer Product Photos Dataset in Chapter 6, we improve the unsupervised low-rank loss to be misalignment-robust. This misalignment-robust loss can be used in many other tasks where the training data is not ideal in conditions.

  The main limitation of this training scheme is that, the low-rank properties in the proposed unsupervised training scheme is a weak supervision for the tasks we demonstrated here (for example, in highlight separation, predicting the diffuse layer as all-zeros can also achieve rank one, which means the correct separation of highlight and diffuse layers is not the only local optimal points), thus the pretraining on a small synthetic dataset is needed for providing a reasonable initialization. Although in Chapter 6, we conduct an experiment showing that without pretraining, the network would still converge to a reasonable performance, but it is not as good as the performance with pretraining. In the future, for specific tasks, if further constraints can be formulated in an unsupervised way, the pretraining step may be fully avoided.

  Furthermore, we want to add some additional discussions. One may be curious why we did not choose to use domain adaptation to solve the domain shift between synthetic and real data. In domain adaptation, when the training domain and testing domain are different, we can align the two domains either in the input space or the internal feature

space to solve the domain shift. It works when the trained network performed well for training data but poorly for testing data who is in another domain. In other words, if testing data is similar to the training data, it should perform very well, because the network is trained successfully for the training domain. In our case, we only use a small amount of synthetic data for pretraining, and the pretrained network does not perform well even for synthetic data, it only provides a reasonable initialization for the finetuning phase. Thus in our two-phase training, we only have to render a small amount of synthetic data because rendering is very expensive. If we want to train the network on synthetic data, then use domain adaptation to solve the adaptation on real images, instead of finetuning on real images, we have to generate a large amount of good quality synthetic data, to make sure the trained network works very well on synthetic data, then it is possible that domain adaptation can solve the domain shift between synthetic and real data. That is the reason why we did not choose domain adaptation.

- **Illumination estimation from separated image layers.** In Chapter 7, we propose an illumination estimation method which traces the lights from the separated highlight layer back to the environment. Together with the previous techniques of face geometry estimation, the estimated illumination achieved higher accuracy compared with the state-of-the-art methods over a series of experiments. The non-parametric representation model also has a better ability to represent both low-frequency and high-frequency of illuminations, unlike the commonly used low-order spherical harmonics representation, which can only represent low-frequency illuminations. Furthermore, by using statistics of human skins from the MERL/ETH Skin Reflectance Database [113], our method reverses the convolution by the Phong specular lobe and recover the HDR (high dynamic range) illumination.

  The limitations of this method are: firstly, the estimated illumination is only for the location of the face, since for indoor scenes, the assumption of distant lighting does not hold. Thus, currently, we cannot estimate the illumination of other location in the input photo. Although we demonstrate the estimation of spatially variant illumination from multiple faces in the input image, the scheme is very naive and only works for simple lighting environments (where we only consider the condition of multiple points lights or area lights, when illumination environment is more complicated, there are more details, such as the distortion of the environment map at different locations, should be taken into consideration). A more sophisticated and robust method should be proposed in the future. Secondly, we only use frontal faces as light probes and do not consider faces from other views. If face frontalization methods are incorporated into the pipeline, maybe we can use faces from random views as light probes in the future.

8.2 Future works

In this thesis, we propose methods for a series of image layer separation problems and one application. Based on the proposed methods, there are several interesting directions that are worthy to explore in the future.

- **Fully automatic fence removal pipeline of dynamic videos.** In Chapter 3, we propose a novel framework for separating fence layers from dynamic videos. However, a fully automatic video de-fencing pipeline should not only contain an automatic fence separation step, but also an automatic background restoration step. In our current pipeline, we apply existed in-painting techniques in the background restoration step. However, unlike image in-painting techniques where there is no information provided for the occluded regions, we can get information of occluded regions from other frames of the input videos, because the fences are thin, and the motion of fences and backgrounds are different, no regions will be occluded all the time. A better background restoration method can be proposed specifically for this problem accordingly.

- **General objects as light probes for recovering spatially variant illumination.** In Chapter 7, we use faces from a single image as light probes for following augmented reality applications. However, a frontal face may not always exist in the input images, but in every image, there would be some objects that can be used as light probes in similar ways. It is very useful for mixed reality applications like rendering virtual objects into real scenes even without the occurrence of faces. In the future, we plan to propose a method to use general objects as light probes, based on the work described in Chapter 6 where an end-to-end network is proposed to separate highlight, diffuse reflectance and shading layers.

- **Recovering spatially variant illumination from multiple face/object light probes.** In Chapter 7, we propose an application of illumination estimation based on the highlight layer separation of face images. However, the estimated illumination is the illumination at the face location. Although it is usually assumed that outdoor illumination is distant lighting, this assumption does not hold for indoor scenes, which is happened to be the application scenario of most AR applications. Although we demonstrate that with multiple frontal faces in the image, it is possible to recover the spatially variant illumination by triangulation of the light sources. However, this scheme is very simple and naive and mostly works for point lights and area lights. When the illuminations are complicated, it may cause some confusions while corresponding the light sources. In the future, we plan to improve the recovery of spatially variant illumination from multiple face/object light probes, which will benefit user-interacted mixed reality applications. For example, with recovered spatially variant

illumination at arbitrary locations in the scene, users can drag virtual objects to random locations and the objects will be rendered under the illumination of the location accordingly.

- **Material estimation from separated image layers.** Material estimation aims to estimate the physical properties (such as diffuse reflectances, specularities, or BRDFs) of the material from one or more RGB images. Material estimation from a single image is difficult, but they are highly relevant to the separated highlight, diffuse, and diffuse reflectance layers by our proposed methods. It is potential to infer the material from these image layers. Comparing to current material estimation methods inferring from RGB images directly, methods based on separated image layers will have higher accuracy since the influence of various illuminations can be avoided. It will enable a set of following applications to edit not only the synthetic objects but also real objects in the scene, such as material transfer, relighting, and so on.

Bibliography

[1] Y. Akashi and T. Okatani. Separation of reflection components by sparse nonnegative matrix factorization. *Compt. Vision and Image Underst.*, 146(C):77–85, 2016.

[2] Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. Video snapcut: robust video object cutout using localized classifiers. *ACM Trans. on Graph. (Proc. of SIGGRAPH)*, 28(3):70, 2009.

[3] J. T. Barron and J. Malik. Intrinsic scene properties from a single rgb-d image. In *Proc. CVPR*, 2013.

[4] Jonathan T. Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Trans Pattern Anal Mach Intell (PAMI)*, 37(8):1670–1687, 2015.

[5] S. Bell, K. Bala, and N. Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics*, 33(4), 2014.

[6] Marcelo Bertalmio, Andrea L Bertozzi, and Guillermo Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proc. CVPR*, volume 1, pages I–355. IEEE, 2001.

[7] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proc. ACM SIGGRAPH*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[8] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *ACM SIGGRAPH*, pages 187–194. ACM, 1999.

[9] Nicolas Bonneel, Kalyan Sunkavalli, James Tompkin, Deqing Sun, Sylvain Paris, and Hanspeter Pfister. Interactive intrinsic video editing. *ACM Transactions on Graphics*, 33(6):1–10, 2014.

[10] Adrien Bousseau, Sylvain Paris, and Frédo Durand. User-Assisted Intrinsic Images. *ACM Trans. Graph*, 28, 2009.

[11] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. PAMI*, 26(9):1124–1137, 2004.

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

[13] Dan A Calian, Jean-François Lalonde, Paulo Gotardo, Tomas Simon, Iain Matthews, and Kenny Mitchell. From faces to outdoor light probes. In *Computer Graphics Forum*, volume 37, pages 51–61. Wiley Online Library, 2018.

[14] 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.

[15] Qifeng Chen and Vladlen Koltun. A simple model for intrinsic image decomposition with depth cues. In *Proc. ICCV*, 2013.

[16] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Processing*, 13(9):1200–1212, 2004.

[17] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[18] Chen Du, Byeongkeun Kang, Zheng Xu, Ji Dai, and Truong Nguyen. Accurate and efficient video de-fencing using convolutional neural networks and temporal information. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018.

[19] Sylvain Duchêne, Clement Riant, Gaurav Chaurasia, Jorge Lopez Moreno, Pierre-Yves Laffont, Stefan Popov, Adrien Bousseau, and George Drettakis. Multiview intrinsic images of outdoors scenes with an application to relighting. *ACM Trans. Graph.*, 34(5):164:1–164:16, November 2015.

[20] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[21] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. Revisiting deep intrinsic image decompositions. In *Proc. CVPR*, 2018.

[22] Brian V. Funt, Mark S. Drew, and Michael Brockington. Recovering shading from color images. In *Proc. ECCV*, pages 124–132, 1992.

[23] 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 (SIGGRAPH Asia)*, 9(4), 2017.

[24] Pablo Garrido, Levi Valgaerts, Chenglei Wu, and Christian Theobalt. Reconstructing detailed dynamic face geometry from monocular video. *ACM Transactions on Graphics (TOG)*, 32(6):158:1–158:10, 2013.

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

[26] Jie Guo, Zuojian Zhou, and Limin Wang. Single image highlight removal with a sparse and low-rank reflection model. In *Proc. ECCV*, September 2018.

[27] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102. Springer, 2016.

[28] Tal Hassner, Shai Harel, Eran Paz, and Roee Enbar. Effective face frontalization in unconstrained images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4295–4304, 2015.

[29] James Hays, Marius Leordeanu, Alexei A Efros, and Yanxi Liu. Discovering texture regularity as a higher-order correspondence problem. In *Proc. ECCV*, pages 522–535. Springer, 2006.

[30] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2011.

[31] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2017.

[32] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks.

[33] Michael Janner, Jiajun Wu, Tejas D Kulkarni, Ilker Yildirim, and Josh Tenenbaum. Self-supervised intrinsic image decomposition. In *Advances in Neural Information Processing Systems*, pages 5936–5946, 2017.

[34] Junho Jeon, Sunghyun Cho, Xin Tong, and Seungyong Lee. Intrinsic image decomposition using structure-texture separation and surface normals. In *Proc. ECCV*, 2014.

[35] Sankaraganesh Jonna, Sukla Satapathy, and Rajiv R Sahay. Stereo image de-fencing using smartphones. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 1792–1796. IEEE, 2017.

[36] Ira Kemelmacher-Shlizerman and Ronen Basri. 3d face reconstruction from a single image using a single reference face shape. *IEEE Trans Pattern Anal Mach Intell (PAMI)*, 33(2):394–405, 2011.

[37] Vrushali S Khasare, Rajiv Ranjan Sahay, and Mohan S Kankanhalli. Seeing through the fence: Image de-fencing using a video sequence. In *Proc. ICIP*, pages 1351–1355, 2013.

[38] Hyeongwoo Kim, Hailin Jin, Sunil Hadap, and Inso Kweon. Specular reflection separation using dark channel prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1460–1467, 2013.

[39] S. Kim, K. Park, K. Sohn, and S. Lin. Unified depth prediction and intrinsic image decomposition from a single image via joint convolutional neural fields. In *Proc. ECCV*, 2016.

[40] R. Kimmel, M. Elad, D. Shaked, R. Keshet, and I. Sobel. A variational framework for retinex. *IJCV*, 52:7–23, 2003.

[41] Gudrun J Klinker, Steven A Shafer, and Takeo Kanade. The measurement of highlights in color images. *International Journal of Computer Vision*, 2(1):7–32, 1988.

[42] Sebastian B Knorr and Daniel Kurz. Real-time illumination estimation from faces for coherent rendering. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, pages 113–122. IEEE, 2014.

[43] Jana Košecká and Wei Zhang. Video compass. In *Proc. ECCV*, pages 476–490. Springer, 2002.

[44] Balazs Kovacs, Sean Bell, Noah Snavely, and Kavita Bala. Shading annotations in the wild. In *Proc. CVPR*, 2017.

[45] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, 2011.

[46] P.-Y. Laffont and J.-C. Bazin. Intrinsic decomposition of image sequences from local temporal variations. In *Proc. ICCV*, pages 433–441, 2015.

[47] Pierre-Yves Laffont, Adrien Bousseau, Sylvain Paris, Frédo Durand, and George Drettakis. Coherent intrinsic images from photo collections. *ACM Transactions on Graphics*, 31(6):1, nov 2012.

[48] Jean-François Lalonde, Srinivasa G Narasimhan, and Alexei A Efros. What does the sky tell us about the camera? In *European conference on computer vision*, pages 354–367. Springer, 2008.

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

[50] E.H. Land and J.J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 3:1684 – 1692, 1971.

[51] Kyong Joon Lee, Qi Zhao, Xin Tong, Minmin Gong, Shahram Izadi, Sang Uk Lee, Ping Tan, and Stephen Lin. Estimation of intrinsic image sequences from image+depth video. In *Proc. ECCV*, pages 327–340, 2012.

[52] Chen Li, Stephen Lin, Kun Zhou, and Katsushi Ikeuchi. Radiometric calibration from faces in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3117–3126, 2017.

[53] Chen Li, Stephen Lin, Kun Zhou, and Katsushi Ikeuchi. Specular highlight removal in facial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3107–3116, 2017.

[54] Chen Li, Kun Zhou, and Stephen Lin. Intrinsic face image decomposition with human face priors. In *Proceedings of European Conference on Computer Vision*, 2014.

[55] Chen Li, Kun Zhou, and Stephen Lin. Simulating makeup through physics-based manipulation of intrinsic image layers. In *Proc. CVPR*, 2015.

[56] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. In *ACM Trans. on Graph. (Proc. of SIGGRAPH)*, volume 23, pages 303–308. ACM, 2004.

[57] Yuanzhen Li, Stephen Lin, Hanqing Lu, and Heung-Yeung Shum. Multiple-cue illumination estimation in textured scenes. In *Proceedings of International Conference on Computer Vision*, pages 1366–1373, 2003.

[58] Zhengqi Li and Noah Snavely. Learning intrinsic image decomposition from watching the world. In *Proc. CVPR*, 2018.

[59] Ce Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, MIT, 2009.

[60] Y. Liu, Z. Yuan, N. Zheng, and Y. Wu. Saturation-preserving specular reflection separation. In *Proc. CVPR*, 2015.

[61] Yanxi Liu, Tamara Belkina, James Hays, and Roberto Lublinerman. Image de-fencing. In *Proc. CVPR*. 2006.

[62] S. Lombardi and K. Nishino. Reflectance and illumination recovery in the wild. *IEEE Trans. PAMI*, 38(1):129–141, 2016.

[63] Jorge Lopez-Moreno, Sunil Hadap, Erik Reinhard, and Diego Gutierrez. Compositing images through light source detection. *Computers & Graphics*, 34(6):698–707, 2010.

[64] Leon B Lucy. An iterative technique for the rectification of observed distributions. *The astronomical journal*, 79:745, 1974.

[65] Wei-Chiu Ma, Hang Chu, Bolei Zhou, Raquel Urtasun, and Antonio Torralba. Single image intrinsic decomposition without a single intrinsic image. In *Proc. ECCV*, pages 211–229, 2018.

[66] S. P. Mallick, T. Zickler, P. N. Belhumeur, and D. J. Kriegman. Specularity removal in images and videos: A pde approach. In *Proc. ECCV*, 2006.

[67] Mathworks. Matlab r2014b.

[68] Y. Matsushita, S. Lin, S. B. Kang, and H.-Y. Shum. Estimating intrinsic images from image sequences with biased illumination. In *Proc. ECCV*, pages 274–286, 2004.

[69] Yadong Mu, Wei Liu, and Shuicheng Yan. Video de-fencing. *arXiv preprint arXiv:1210.2388*, 2012.

[70] T. Narihira, M. Maire, and S. X. Yu. Learning lightness from human judgement on relative reflectance. In *Proc. CVPR*, 2015.

[71] Takuya Narihira, Michael Maire, and Stella X Yu. Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2992–2992, 2015.

[72] Alasdair Newson, Andrés Almansa, Matthieu Fradet, Yann Gousseau, and Patrick Pérez. Video inpainting of complex scenes. 2014.

[73] Ko Nishino and Shree K Nayar. Eyes for relighting. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 704–711. ACM, 2004.

[74] T. Okabe, I. Sato, and Y. Sato. Spherical harmonics vs. haar wavelets: Basis for recovering illumination from cast shadows. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 50–57, 2004.

[75] Roy Or-El, Guy Rosman, Aaron Wetzler, Ron Kimmel, and Alfred M. Bruckstein. Rgbd-fusion: Real-time high precision depth recovery. In *Proc. CVPR*, 2015.

[76] Alexandros Panagopoulos, Chaohui Wang, Dimitris Samaras, and Nikos Paragios. Illumination estimation and cast shadow detection through a higher-order graphical model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[77] Théodore Papadopoulo and Manolis IA Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. In *European Conference on Computer Vision*, pages 554–570. Springer, 2000.

[78] Minwoo Park, Kyle Brocklehurst, Robert T Collins, and Yanxi Liu. Image de-fencing revisited. In *Proc. ACCV*, pages 422–434. Springer, 2011.

[79] Minwoo Park, Robert T Collins, and Yanxi Liu. Deformed lattice discovery via efficient mean-shift belief propagation. In *Proc. ECCV*, pages 474–485. Springer, 2008.

[80] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 296–301. Ieee, 2009.

[81] Saulo Pessoa, Guilherme Moura, Joao Lima, Veronica Teichrieb, and Judith Kelner. Photorealistic rendering for augmented reality: A global illumination and brdf solution. In *Virtual Reality Conference (VR), 2010 IEEE*, pages 3–10. IEEE, 2010.

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

[83] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2482–2491, 2018.

[84] Ravi Ramamoorthi and Patrick Hanrahan. A signal-processing framework for inverse rendering. In *ACM SIGGRAPH*, pages 117–128. ACM, 2001.

[85] Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Efstratios Gavves, and Tinne Tuytelaars. Deep reflectance maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4508–4516, 2016.

[86] E. Richardson, M. Sela, R. Or-El, and R. Kimmel. Learning detailed face reconstruction from a single image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[87] William Hadley Richardson. Bayesian-based iterative method of image restoration. *JOSA*, 62(1):55–59, 1972.

[88] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. End-to-end weakly-supervised semantic alignment.

[89] Carsten Rother, Martin Kiefel, Lumin Zhang, Bernhard Scholkopf, and Peter V. Gehler. Recovering intrinsic images with a global sparsity prior on reflectance. In *NIPS*, pages 765–773, 2011.

[90] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graph. (Proc. of SIGGRAPH)*, 23(3):309–314, 2004.

[91] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE Trans Vis Comput Graph (TVCG)*, 5:1–12, 1999.

[92] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Illumination from shadows. *IEEE Trans Pattern Anal Mach Intell (PAMI)*, 25:290–300, 2003.

[93] M. Sela, E. Richardson, and R. Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In *Proceedings of International Conference on Computer Vision*, 2017.

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

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

[96] J. Shen, X. Yang, Y. Jia, and X. Li. Intrinsic images using optimization. In *Proc. CVPR*, 2011.

[97] L. Shen, P. Tan, and S. Lin. Intrinsic image decomposition with non-local texture cues. In *Proc. CVPR*, 2008.

[98] L. Shen and C. Yeo. Intrinsic images decomposition using a local and global sparse representation of reflectance. In *Proc. CVPR*, pages 697–704, 2011.

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

[100] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training.

[101] Snap. Inc. Snapchat.

[102] Ping Tan, Stephen Lin, and Long Quan. Separation of highlight reflections on textured surfaces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1855–1860, 2006.

[103] Ping Tan, Stephen Lin, Long Quan, and Heung-Yeung Shum. Highlight removal by illumination-constrained inpainting. In *null*, page 164. IEEE, 2003.

[104] R. Tan and K. Ikeuchi. Reflection components decomposition of textured surfaces using linear basis functions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 125–131, 2005.

[105] Robby T Tan and Katsushi Ikeuchi. Separating reflection components of textured surfaces using a single image. *IEEE Trans. PAMI*, 27(12):178–193, 2005.

[106] Robby T Tan, Ko Nishino, and Katsushi Ikeuchi. Separating reflection components based on chromaticity and noise analysis. *IEEE transactions on pattern analysis and machine intelligence*, 26(10):1373–1379, 2004.

[107] Marshall F. Tappen, Edward H. Adelson, and William T. Freeman. Estimating intrinsic component images using non-linear regression. In *Proc. CVPR*, pages 1992–1999, 2006.

[108] Theo Thonat, Abdelaziz Djelouah, Frédo Durand, and George Drettakis. Thin structures in image based rendering. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)*, 37(4):12, 2018.

[109] Yang Wang and Dimitris Samaras. Estimation of multiple illuminants from a single image of arbitrary known geometry. In *Proceedings of European Conference on Computer Vision*, pages 272–288, 2002.

[110] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[111] Yair Weiss. Deriving intrinsic images from image sequences. In *Proc. ICCV*, pages 68–75, 2001.

[112] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Trans. PAMI*, 29(3):463–476, 2007.

[113] Tim Weyrich, Wojciech Matusik, Hanspeter Pfister, Bernd Bickel, Craig Donner, Chien Tu, Janet McAndless, Jinho Lee, Addy Ngan, Henrik Wann Jensen, et al. Analysis of human faces using a measurement-based skin reflectance model. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1013–1024. ACM, 2006.

[114] Tianfan Xue, Michael Rubinstein, Ce Liu, and William T Freeman. A computational approach for obstruction-free photography. *ACM Transactions on Graphics (TOG)*, 34(4):79, 2015.

[115] Atsushi Yamashita, Akiyoshi Matsui, and Toru Kaneko. Fence removal from multifocus images. In *Proc. ICPR*, pages 4532–4535. IEEE, 2010.

[116] Fei Yang, Jue Wang, Eli Shechtman, Lubomir Bourdev, and Dimitri Metaxas. Expression flow for 3d-aware face component transfer. In *ACM Transactions on Graphics (TOG)*, volume 30, page 60. ACM, 2011.

[117] Qingxiong Yang, Shengnan Wang, and Narendra Ahuja. Real-time specular highlight removal using bilateral filtering. In *European conference on computer vision*, pages 87–100. Springer, 2010.

[118] Genzhi Ye, Elena Garces, Yebin Liu, Qionghai Dai, and Diego Gutierrez. Intrinsic video and applications. *ACM Transactions on Graphics*, 33(4):1–11, 2014.

[119] Renjiao Yi, Jue Wang, and Ping Tan. Automatic fence segmentation in videos of dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 705–713, 2016.

[120] Renjiao Yi, Chenyang Zhu, Ping Tan, and Stephen Lin. Faces as lighting probes via unsupervised deep highlight extraction. In *Proc. ECCV*, September 2018.

[121] Lap-Fai Yu, Sai-Kit Yeung, Yu-Wing Tai, and Stephen Lin. Shading-based shape refinement of rgb-d images. In *Proc. CVPR*, 2013.

[122] Qi Zhao, Ping Tan, Qiang Dai, Li Shen, Enhua Wu, and Stephen Lin. A closed-form solution to retinex with nonlocal texture constraints. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1437–1444, 2012.

[123] Tinghui Zhou, Philipp Krahenbuhl, and Alexei A Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3469–3477, 2015.

[124] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.