COMPUTATIONAL LIGHTING DESIGN AND IMAGE FILTERING FOR MATERIAL ENHANCEMENT

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Ivaylo Ivanov Boyadzhiev

August 2015

COMPUTATIONAL LIGHTING DESIGN AND IMAGE FILTERING FOR MATERIAL ENHANCEMENT

Ivaylo Ivanov Boyadzhiev, Ph.D.

Cornell University 2015

Photography provides a powerful tool for depicting the world around us by capturing the intricate relationship between light and materials. Indeed, much of the art and craft of photography is based on understanding how light interacts with surface properties, and how this interaction gets captured by the camera.

The recent digitization of photography, and in particular the point and shoot paradigm, has made photography accessible to millions of people by automating a large part of the decision making process. For example, modern cameras provide algorithms to automatically set many of the tunable parameters, such as white balance, exposure time and aperture size that are appropriate to a given scene.

However, there is little or no support for the end-user when it comes to helping them to light or manipulate surface appearance in a scene. A good photographer is never passive towards the lighting in the scene and often manipulates the subject of interest in order to emphasize certain properties. For example, in portrait photography lights can be positioned to control the appearance of wrinkles, or makeup can be used to hide variation in skin color.

One key challenge is that lighting design and manipulation of surface properties are considered highly labor intensive, manual tasks that require specialized equipment and lots of preparation, which is beyond the skills of casual photographers. In this thesis, we develop a family of new computational tools to make

such advanced photographic tasks more accessible to novice users. We call our broad approach **computational lighting and material design**. Our approach is based on entirely image-based, post-process techniques combined with carefully designed user interactions that allow novice users to explore a non-trivial space of solutions, based on common goals in photography. As we show in this thesis, the results are often much more effective and easy to produce than what could be achieved using traditional techniques.

We provide practical methods for three photographic tasks in particular.

1. **Lighting design.** For lighting design, we propose a computational approach that partially automates the process, by allowing novices to uniformly walk around a static scene with a single light source. Then, we describe a set of optimizations to assemble the input lights to create a few *basis lights* that correspond to common goals pursued by photographers, e.g., accentuating edges and curved regions. We also introduce *modifiers* that capture standard photographic tasks, e.g., to alter the lights to soften highlights and shadows, akin to umbrellas and soft boxes. This approach to lighting allows the photographer to achieve sophisticated lighting without a complicated manual setup.

2. **Multi-lights white balance.** When a scene has a mixture of lights with different colors, the common white balance problem becomes much more challenging. In this thesis, we propose a solution to the ill-posed mixed light white balance problem, based on user guidance. We allow users to scribble on a few regions that should have the same color, indicate one or more regions of neutral color, and select regions where the current color looks correct. Then, we reformulate the spatially varying white balance problem as a sparse data interpolation problem in which the user scribbles form constraints. We

demonstrate that our approach can produce satisfying results on a variety of scenes with intuitive scribbles and without any knowledge about the lights.

3. **Material editing.** Our third work addresses the problem of surface manipulation, which is a common pre-processing step in the professional photography pipeline, that often requires a non-trivial physical interaction with the object. For example, in food photography glycerine may be used to give the food a more fresh and appealing look. In this thesis, we reformulate the problem as a post-process step, where the goal is to manipulate material properties after the image has been taken. For example, to increase shininess or to decrease aging cues, such as wrinkles and blemishes. We design and study a set of 2D image operations, based on multi-scale image analysis, that are easy and straightforward, and that can consistently modify perceived material properties. Through user studies, we identify a set of operators that yield consistent subjective effects for a variety of materials and scenes.

The computational methods presented in this dissertation have made a step towards automating advanced photographic tasks by simplifying the required user preparation and interaction. Remaining challenges include developing more general basis lights to support a wider range of photographic objectives (such as portrait photography) as well as finding more powerful methods for acquisition of both static and dynamic scenes. We believe that our work in image-based material editing would spur others to explore this important area. A data-driven approach for both lighting design and material editing is a promising future direction of leveraging state-of-the art machine learning algorithms for developing fully automatic solutions. We also believe the techniques introduced in this dissertation can provide valuable insights for developing computational methods for lighting design and material alternation for both real and CG scenes.

BIOGRAPHICAL SKETCH

Ivaylo (Ivo) Boyadzhiev was born on February 13th, 1986 in Kyustendil, Bulgaria. After graduating from Math and Science Gymnasium, Kyustendil, he began computer science course of studies at Sofia University in 2005. Following completion of the Bachelor of Science program in 2010, he joined Cornell University's computer science department as a Ph.D. student.

To my grandparents Todor and Ganka.

ACKNOWLEDGEMENTS

I would like to first express my deep gratitude to my advisor Kavita Bala for her continuous source of inspiration and guidance. Her incredible instinct for exploring new research paths, solving concrete research problems and always keeping a positive attitude are qualities I shall aspire to for the rest of my life. Without her continuous support and managerial skills to glue together a team of researchers with diverse backgrounds, this thesis would not have been possible.

I am deeply grateful to Sylvain Paris, who was like a second advisor to me. His rich expertise and insightful advise have been invaluable part for many pieces of this dissertation. The three summers I spent working closely with him taught me not only techniques for solving specific research problems, but also how to think as a computer scientist.

I am grateful to Charles Van Loan for his inspiring lectures and unique style of presenting the world of matrix computations, to David Bindel for introducing me to the parallel version of those algorithms, to Noah Snavely for showing me the amazing world of 3D reconstruction, to Thorsten Joachims and Lillian Lee for presenting me the theoretical foundations and practical algorithms for data science and machine learning.

I am very fortunate to have the chance working with many excellent collaborators: Kavita Bala, Sylvain Paris, Edward Adelson, Frédo Durand and Jiawen Chen. Thank you for your hard work during the SIGGRAPH deadlines.

I also thank my lab mates, including my summer internships office mates at Adobe, who have created a stimulating environment and enriched my experience: Michael Gharbi, Yichang Shih, Olga Diamanti, Daniel Hauagge, Kevin Matzen, Nicolas Savva, Pramook Khungurn, Sean Bell, Paul Upchurch, Scott Wehrwein and Kyle Wilson.

There are several friends who made my stay at Cornell pass in the blink of an eye. Thank you Milen, Rado, Maria and Sherry. I want to thank Matey, Aleksandrina and Yered for making my summers in Boston a wonderful experience.

Finally, I would like to thank my parents Ivan and Anastasia, and my sister Teodora for their constant love, support and energy. I thank my wonderful wife Kristine for the great deal of love, support and patience during all those years, and for keeping the Christmas spirit alive throughout the SIGGRAPH deadlines. I thank my grandparents Todor and Ganka for passing on to me their thirst of knowledge and showing me the path of education from an early age. I dedicate this thesis to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

A photograph is a depiction of the 3D reality on a 2D medium. We create photographic depictions in many forms, e.g., still images or videos, and for many reasons. For example, to create a snapshot of reality that will remind us of past locations and events, or to produce compelling visual messages in the advertisement of products, real estate or tourist destinations. In the former case, one may desire that the photograph depicts reality as close as possible, i.e., capture the true colors and dynamic range of the real world. However, in the latter case, when closely studying the techniques used by product or real estate photographers, one realizes that they often have to modify the conditions that reality naturally offers. For these kind of applications, where the goal is to capture certain properties or convey certain information or emotions, the photographer is never passive towards the subject of interest, but actively manipulates both the physical properties of the subject as well as the lighting in the scene.

Photo-graphy means ”writing with light”. Indeed, understanding the intricate interplay between light and physical materials is a large part of the art and craft of photography. The photographer is always mindful towards lighting, both when using natural or artificial light sources. For example, outdoor shots may require waiting for the ideal time of day or in some cases the ideal time of year for the desired lighting conditions. Further, it is a common practice to add artificial lighting when the properties of the natural light are not good enough. For instance, a flash light is often welcome to “fill-in” too sharp shadows for portraits under direct natural illumination, and many photographers use artificial projectors and reflectors to improve natural lighting. In studio photography,

(a) Realistic look (© Flickr user blair_25) (b) Hyper-realistic look (© Harold Ross)

Figure 1.1: *Studio photography: realistic vs hyper-realistic look.*

characterized by its complex installation of artificial light sources, the photographer has full control over the lighting. This gives a lot of expressive power and a trained photographer can set up lighting that depicts the subject under a variety of looks, ranging from realistic to hyper-realistic, Figure 1.1. However, this level of understanding, planning and usage of specialized equipment is beyond what casual photographers have access to. Thus, making the difference between professionally planned versus point and shoot photographs quite obvious most of the time, Figure 1.2.

In addition to lighting, photographers care a great deal about the surface appearance of objects they photograph; indeed, much of the craft of studio photography involves controlling material appearance using physical techniques. A portrait photographer may control the appearance of skin wrinkles by the use of lighting. Makeup can be used to hide variation in skin color (e.g., blemishes or mottling). Powder can be applied to make skin appear less shiny. In product photography, a dulling spray may be used to reduce specular highlights. In food photography, specularity may be desirable, and it can be enhanced with a glycerine spray, making the food look fresher or juicier. In movie shots, objects

(a) Naive lighting (© Martha Stewart) (b) Professional lighting (© Tango Mango)

Figure 1.2: *Naive vs professional lighting*. Salad with Russian dressing.

might be given a more worn out or weathered look to provoke a certain emotion of the scene. All this requires special planning and physical pre-processing which is beyond the time budget and effort that most casual photographers are willing to spend on a single shot.

The science of photography provides a powerful illusion of reality, and as the famous quote by the photographer Arnold Newman says "Photography, as we all know, is not real at all. It is an illusion of reality with which we create our own private world." While lighting and surface properties are two of the most important axes along which the photographer can manipulate the reality, there are many more degrees of freedom in the construction of photographs. For example, camera orientation, focus distance, exposure time, aperture size and white balance settings can all have a dramatic impact on the final captured image. Thus, for every single shot, the photographer has to optimize his goal over a

complicated multi-dimensional function where lighting, surface properties and the various camera settings, including position and orientation, are mapped into a final 2D image, representing some version of the 3D reality. This freedom of choice is what attracts so many creative people into photography, since it allows them to depict their "own private world". Unfortunately, this can also be discouraging to casual photographers, since often times their pictures look nothing like the professional photographs they are used to see, or even the reality they perceive with their own eyes.

Modern digital cameras have been designed with simplicity in mind: they harness a variety of sophisticated computational photography algorithms in order to provide an automatic selection for many of the camera settings like focus distance, exposure time, aperture size and white balance. Thus, reducing the degrees of freedom for novice photographers. This is usually done based on various presets, [46; 151], or more advanced algorithms based on scene analysis, [41; 99; 13]. For example, in landscape mode a small aperture will be chosen, such as f/16 in order to keep everything in focus, [151]. In portrait mode, a larger aperture will be selected in order to produce a shallow depth of field to make the main subject stand out, [151]. Furthermore, colors can be remapped based on the preset, e.g., more vivid blue and green colors for landscape and more natural skin colors for portrait mode, [34; 96]. Advanced algorithms would also analyze the scene to determine the best settings applicable for the concrete situation. For example, faces can be detected to set the focus region that renders them sharp, [41], or scene colors can be analyzed to set the proper white balance settings, [88; 38; 148]

However, despite the advancement in digital camera technologies, pho-

tographic tasks such as lighting design and manipulation of surface properties are still considered advanced, highly labor intensive and requiring specialized knowledge and equipment. Even the common white balance correction becomes a much more challenging task once the scene has two or more light sources with different colors. For example, this happens often in lighting design when combining natural light with artificial light sources like tungsten or fluorescent lamps. Fixing the unpleasant color casts is a required step towards producing believable colors, since the human visual system has a certain level of invariance towards colored light sources, but this invariance does not hold when observing pictures of the same scene, [76; 89].

In this thesis, we explore one possible approach to address common photographic challenges. We devise computational mechanisms to make advanced photographic tasks, such as (1) lighting design, (2) white balance under mixed lighting and (3) image-based material editing, more accessible to novice users. Our general approach is based on combining state-of-the-art image analysis techniques with carefully designed user interactions that allow both novice and professional photographers to quickly explore a non-trivial space of potential solutions (for white balance), designs (for lighting) and modifications (for material editing). As we show in this thesis, for casual photographers the results are often much more powerful and easier to obtain than what could be achieved using traditional techniques.

1.1 Summary of contributions

Computational lighting design. We propose a practical approach to lighting design, based on taking multiple pictures of a static scene by moving a single light source, uniformly around the scene. We develop computational mechanisms to extract a set of basis lights based on common goals in photography, and we introduce modifiers that affect the basis lights and achieve effects similar to standard lighting equipment, such as soft boxes to soften highlights and shadows, and snoots to restrict the light extent. We design these lights and modifiers by reasoning entirely in image space, thereby avoiding a computationally expensive, and potentially brittle 3D reconstruction of the scene. Our approach removes the need for manually intensive pre-planned lighting design.

Computational lighting design for product videography. To assist novice users with the production of product videos we propose a set of design principles for product videography. We support these principles with empirical observations of professionally made videos and existing literature on lighting design. We introduce a simple acquisition procedure to generate short video snippets covering a wide variety of light configurations. We describe robust analysis tools to rank these snippets according to the criteria that we expressed in our design principles. We demonstrate a user interface based on faceted search to browse the snippets and assemble a few of them to produce a quality video.

White balance under mixed lighting. We propose a practical algorithm, based on simple user scribbles, to white balance scenes illuminated by a mixture of light sources, with no assumption on their number, color or configuration. Our user-

assisted scribbles only deal with what humans identify most easily: reflectance properties. We re-formulate the white balance problem as an interpolation problem, where the user scribbles form a sparse set of constraints, and we show that the null space of the Matting Laplacian spans the white-balance solution for canonical local configurations.

Image-based material editing. We introduce a new approach to image-based material editing based on multi-scale image decomposition and sifting of sub-band coefficients. We propose and study a space of band-sifting operators that act along several criteria at the same time, based on scale, amplitude, and sign of the signal. We study the perceptual effects of our band-sifting operators; we validate their perceptual consistency through user studies; and we demonstrate their usefulness for both image and video post-process material editing.

1.2 Organization of the dissertation

The dissertation is divided into a total of 7 chapters and supplementary appendices. It is organized as follows: Chapter 2 presents an overview of relevant prior work on computational lighting design, white balance under mixed-lighting and image-based material editing. In Chapter 3, we describe our computational mechanism for lighting design of large indoor and outdoor spaces using multi-lights image collections. In Chapter 4 we introduce our follow-up project, which extends our previous lighting design work for product photography and videography, while further simplifying the required equipment. In Chapter 5, we develop a user-guided white balance algorithm for mixed lighting conditions.

In Chapter 6, we propose an approach for image-based material editing based on a novel look of the well known multi-scale image decomposition techniques. Chapter 7 presents our conclusions and suggestions for future research directions. In Appendix A we show a detailed derivation of the Matting Laplacian interpolation for the case of local duochromatic reflectance model under multi-lights conditions. In Appendix B we provide a detailed pseudo-code implementation of our image-based material editing operators.

CHAPTER 2

RELATED WORK

In computer graphics, there has been a large body of work related to capturing and editing various subsets of the plenoptic function, [93; 68; 94; 58; 3]. The plenoptic function, introduced by Adelson and Bergen [1], is an idealized function that expresses the image of a scene from any possible viewing position at any viewing angle at any point in time. Thus, the plenoptic function provides a precise notion of the visual world surrounding us that we wish to depict. From that perspective, our lighting design work in Chapter 3 is similar to Agarwala's thesis [2], where the goal is to capture a few slices of the plenoptic function and then fuse them into a final composition that captures the best of all individual pictures. On the other hand, in our mixed-lights white balance and image-based material editing projects we work with a single slice of the plenoptic function, and the goal is to modify the image as if the conditions, at the time when the slice of the plenoptic function was taken, were different. For example, in Chapter 5 our goal is to render the scene as if the color of all light sources was white, which is a required step towards rendering plausible looking material colors [76]. In Chapter 6 our goal is to render the scene as if material properties, such as shininess, smoothness or weathering, were different, e.g, modify the plenoptic slice as if the captured object was more or less shiny.

While we always start with a slice or a few slices of the plenoptic function, the modifications that we introduce produce physically plausible, but not necessarily physically correct results. Similar to the depiction principles described by Durand in [47], many of our user-assisted modifications are inspired by aesthetic principles in photography. Furthermore, many of our objectives can be

related to established principles in the emerging field of neuroesthetics, [52; 158; 110], where the goal is to find why the human brain finds some artistic works more alluring than others. For example, our white balance work is related to the *constancy* property of the human brain, which has the unique ability to retain knowledge of constant and essential properties of an object and discard irrelevant dynamic properties such as changes of the light colors [157]. In our lighting design project, we propose an optimization criterion that aims to enhance essential edges by finding a proper mixture of the input lights. This is related to the *contrast principle* observed by [130], which involves eliminating redundant information and focusing attention. Our image-based material editing work can be related to the *peak shift principle*, [121], where we emphasize certain properties of the input signal to make the observer respond strongly on stimuli that are already presented in the image.

In the rest of this chapter we conduct a specific literature review on each of the three directions: (1) computational lighting design, (2) white balance for mixed lighting conditions and (3) image-based material editing.

2.1 Lighting design for multi-lights image collections

We discuss related work in terms of computational lighting design, single-image lighting editing, and 3D lighting design.

Computational Lighting Design. Debevec et al. [42], Akers et al. [5] and Agarwala et al. [3] provide user interfaces to combine several images taken from the same viewpoint but under different lighting, thereby introducing the idea of computational lighting design. In Debevec et al. a scene can be realistically relit under

novel illumination conditions, by treating the input images as a basis and fitting a lighting model per image pixel. However, their method needs to know the position of the light source in each image, which requires specialized acquisition equipment. For comparison, our technique is based on simple, widely available equipment, such as a single flash, and we do not need to know the light positions. Further, their custom device has been designed for medium scale scenes, such as human faces, whereas we are interested in large scale architectural scenes. In Akers et al. and Agarwala et al. users mark the regions of interest in the input images and the algorithm is in charge of producing a satisfying composite. While this is a reasonable approach when there are only a few input photos, it becomes intractable when there are a hundred of them. With such datasets, deciding which images to use, and which parts in them to combine, is a major challenge that is as difficult as producing the actual combination. In our work, we introduce *basis light sources* to organize the many input photos into a smaller, more manageable set of images that correspond to standard photography goals.

Raskar et al. [122], Cohen et al. [40], Fattal et al. [55] and Mertens et al. [108] combine several photos taken under different lighting to generate a better picture. Raskar et al. generate non-photorealistic results, whereas we seek to retain a photorealistic look. Cohen et al. and Fattal et al. focus on a single object and aim at revealing the object's details and removing shadows, while ignoring other effects such as shadows projected onto other objects. As we shall see, this becomes an issue in larger scenes, that are of interest in our work. Further, their approaches are mostly automatic, with a few presets offered to users, whereas we give more control so that users can make artistic choices. Mertens et al. target high-dynamic range scenes as their main applications, and they show that their technique can also apply to simple multi-light configurations. But this technique

does not handle the more diverse lighting configurations of the computational lighting workflow well.

Single-Image Lighting Editing. Several techniques exist to manipulate the lighting in a single image. For instance, Carroll et al. [32] describe how to alter the interreflections in a picture after the user makes some annotations to describe the lighting configuration. Bousseau et al. [16] use similar annotations to perform white balance. Mallick et al. [101] control the intensity of specularities. Tone-mapping operators remap intensities to fit the dynamic range to a given display [123]. Photo editing software follows a similar approach to define adjustments that brighten shadows and decrease highlights. From our perspective, all these methods have in common that they are limited to a specific effect such as shadow brightening, but keep the spatial configuration unchanged, e.g., shadows cannot be altered. In comparison, we seek to produce a wider range of effects to enable more control over the achieved lighting configuration.

3D Lighting Design. Several techniques exist to edit lighting environments in the context of 3D rendering, e.g., [129; 118; 14]. Compared to our approach, these methods tackle the problem from a fundamentally different direction since they have access to a full geometric description of the scene and have total control over the light sources. In comparison, we have no a priori knowledge about the photographed scene and have access to only a limited number of observations of it. One could try to solve an inverse problem to infer a 3D description of the scene and its materials, but current 3D reconstruction techniques are often fragile, and cannot handle such large problems, especially with a fixed viewpoint.

Lighting Design for Videography. A few techniques exist to relight video content. For instance, Shih et al. [135] adjust the low-frequency illumination of video portraits as part of their style transfer technique. In comparison, we seek a more fine-grained and more generic technique to control the lighting in product videos. Wenger et al. [150] offer such control but requires a dedicated light stage, which limits its use to professional productions. Further, while it enables relighting, it does not assist users in this process unlike our approach that automatically extracts and sorts snippets to help users follow our design principles.

Finally, our approach is also related to techniques that select video snippets either to generate a summary, e.g., [4], or an infinitely looping video, e.g., [128; 95]. However, these techniques are concerned with the duration of the video and are not about lighting design.

2.2 White balance for mixed lighting conditions

White balance is the process of removing unrealistic color casts, so that materials appear as if the color of all lights in the scene was neutral. This is a necessary step to produce realistic looking material colors in photography [76].

The single-light case, where all the light sources have the same color, is well addressed with automatic methods, e.g. [57; 64], and photo editing tools. When used in cases where the lighting is mixed, they have to make compromises and residual color casts remain. For example, Photoshop has a tool that allows users to click on a neutral-color object. However, these single-color techniques all fail on scenes lit by lights of different colors. At best, one can neutralize the effects of

one of the sources (Fig. 5.1) but the unsightly color cast due to the other sources remain.

A few techniques deal with light mixtures. Ebner [50], Riess et al. [124], Bleier et al. [12], and Gijsenij et al. [65] assume that, locally, a single light dominates. This might approximate human perception, but from a photography perspective, the results have faded colors and retain local color casts Ebner [50].

Hsu et al. [75] focus on the two-light scenario and further assume that the light colors are known. In contrast, we do not seek a fully automatic technique and strive for an approach that handles an as-wide-as-possible range of scenes. Furthermore, our experience with their system shows that their voting stage is sensitive to the initial choice of the lights' RGB values, which is not always trivial to specify for scenes taken outside a controlled environment. Finally, as their paper mentions in Section 8, "scenes that exhibit a strong foreground-background separation may also cause problems." This is because they need to observe a given reflectance under a number of different mixtures.

Lischinski et al. [97] describe a scribble interface to perform local edits. This approach can be used to correct a color cast that is well localized in the scene, but it can be tedious in scenes where the light mixture occurs everywhere. Further, this tool requires the user to specify the absolute correction to be applied, which is nontrivial in our case. Determining the local color of the illumination in a complex colorful scene is challenging even for a human observer. In comparison, we make sure that our scribbles only deal with relative characteristics of the scene reflectance, which is significantly easier than determining the absolute color of the local illumination.

Carroll et al. [33] use a scribble interface to edit the color of inter-reflections. While related, our approach deals with effects that are more global and affect large portions of the image, whereas inter-reflections have a limited spatial extent. Further, their technique relies on scribbles that describe properties of the illumination, such as the fact that a light source does not affect a designated area. Since in our context light sources have a global impact, such information would be particularly challenging.

Bousseau et al. [17] and Shen et al. [133] compute intrinsic images, i.e., they separate an object's reflectance from the illumination reaching it. While related to white balance, intrinsic images are also significantly different because they often assume a monochromatic or nearly monochromatic illumination; the main challenge being the estimation of the light intensity at each point. In comparison, we focus on colored illumination and, as we shall see, seek to leave lighting intensity untouched, without estimating it. Further, the technique of Bousseau et al. [17] requires scribbles about absolute and relative properties of the illumination such as "this point is fully lit" or "the illumination in this region is smooth." Since we are dealing with complex multi-source illumination, we cannot expect that users will be able to specify this, and argue that such information is hard to specify for novices and experts alike.

2.3 Material editing from a single image

The appearance of materials depends on various physical parameters, such as: (1) the underlying material properties (2) the underlying geometry and (3) the lighting in the scene. In order to modify the appearance of material properties,

in a physically correct way, one needs to model all those physical processes. However, acquiring both material and geometry properties, from a single image of an arbitrary scene, is a very hard and ambiguous problem [127]. In this project we are interested in studying what material properties can be altered consistently, in a physically plausible way, through entirely image-based operations. Our goal is to produce a visually satisfying illusion of material transformation, but not a physically accurate result, similar to [83].

Image Decomposition. Splitting an image into components is a standard strategy to manipulate some properties independently of others. For instance, one can convert RGB colors into YIQ or CIE-Lab to edit luminance and chrominance independently. The coring operation used for denoising drives the low-amplitude coefficients of a multi-scale decomposition towards zero without changing the high-amplitude coefficients [45; 136]. The classic Retinex algorithm by Land and McCann [89] use a similar amplitude threshold in the gradient domain to separate the illumination from the reflectance of a scene. Mallic et al. [102] describe a technique for separating specular and diffuse reflection components in images and videos. Durand and Dorsey [48] separate large-scale variations from the small-scale ones for the purpose of HDR tone mapping, Bae et al. [9] rely on a similar split for style transfer, and Farbman et al. [53] for a variety of photo edits such as detail enhancement and local sharpening. Heeger and Bergen [74] and Simoncelli and Portilla [120] also use a multi-scale decomposition for their texture synthesis techniques. Motoyoshi et al. [109] showed that manipulating the skewness of the coefficient distribution of the high-frequency bands affects the perceived gloss of materials.

Our work is related to this body of work since it splits images into components

that are later modified separately. However, our purpose here is to look, more systematically, at the range of material-related manipulations that can be attained by doing modifications within the subband domain. Furthermore, material editing is different from other editing tasks because the visual information is distributed across space and subbands, and cannot be easily untangled.

Photo Editing. Many image operators exist to manipulate the level of texture in photographs, e.g., [142; 53; 54; 72; 116; 62; 63; 155; 156; 80]. These works focus on the signal processing challenges, e.g., they improve the output or accelerate the computation. While these methods change the perceived properties of the materials, this aspect is not discussed in these papers. Fattal et al. [56] implicitly use this effect to reveal details that would be hard to see otherwise. However, the perceptual aspects themselves are not studied. Our work is complementary to these articles and focuses on how image operators alter the observer's material perception.

In parallel, the perceptual effect of some editing tools have been quantified. For instance, Mantiuk et al. studied contrast changes [104] and whether image changes are visible [103]. Trentacoste et al. studied the interaction between blur size and image resolution [143] and showed that boosting the high frequencies of an image can be perceived as sharpening, halos, or countershading depending on the selected cutoff [144]. In comparison, our work focuses on material perception.

Hybrid 2D/3D Material Editing. Khan et al. [83] and Vergne et al. [146] edit images to alter materials and their properties. The main difference with our work is their use of 3D data provided by users or inferred from the images themselves — this allows Khan et al. to render new materials using standard

3D rendering, and Vergne et al. to warp images to convey shape and material properties. In comparison, our approach relies solely on the content of the input images and our operators are purely two-dimensional, thereby avoiding any sort of 3D reconstruction, that can be brittle on scenes with complex materials like those in which we are interested.

Weathering 3D Models. Several techniques exist to modify 3D models and simulate aging and weathering, e.g., [67; 107; 66]. One of the effects we demonstrate makes people look older and objects more worn out, with the major difference being that we work purely in 2D. We also study several other effects beside aging, e.g., shininess and wetness.

Material Perception. A few techniques recognize materials depicted in photos, for instance, to differentiate plastic from wood, e.g., [98; 131; 10]. Fleming et al. [61] conduct user studies to explore the interactions between material classification and judgments of material qualities, such as glossiness, roughness and hardness in the visual and semantic domains. In comparison, we are interested in altering the properties of a given material like its shininess or roughness. Researchers have also studied the interplay between physical sources, such as 3D geometry, surface reflectance and the light field in the perception of surface properties, such as gloss [84; 106; 105]. In our work we are interested in changing perceived surface properties, such as gloss, based on entirely image-based operations. More related to our approach, a few studies have shown a correlation between image statistics and the perception of properties such as translucency [59] and lightness [109; 132], and have proposed image filters that manipulate these statistics to alter

the specific property that they study. Our work is inspired by these techniques and we build upon some of their findings. However, whereas these papers focus on a single effect, we explore a larger spectrum of effects, and systematically characterize how they affect material perception.

CHAPTER 3

COMPUTATIONAL LIGHTING DESIGN

In this chapter, we present the first component of our suite for assisting advanced photographic tasks: a new computational approach to lighting design.

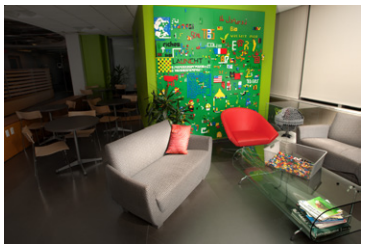
The appearance of materials depends heavily on the arrangement and the quality of lights, but traditional photography equipment is hard to setup and control, and thus difficult for casual photographers. We simplify the equipment requirements by allowing novice users to walk around the scene with a single light source. We then describe algorithms that combine the captured data into a set of basis lights and modifiers that correspond to common goals in photography, e.g., emphasize edges, reveal colors or produce soft lighting. We allow users to apply those settings globally or by selecting local regions in the scene. The proposed approach allows novice users to achieve sophisticated lighting designs with minimal training and equipment. This work originally appeared at ACM SIGGRAPH 2013 [27].

3.1 Introduction

Lighting is a key component of photography, on an equal footing with other aspects such as composition and content. In many cases, photographers actively illuminate their subject with a variety of lights to obtain a desired look. Lighting a scene is a challenging task that is the topic of many courses and books, e.g., [77]. Not only the notion of "good" lighting is elusive and heavily relies on one's subjectivity, but the traditional way to set up the lights itself is complex. Positioning and setting the power of each flash is a nontrivial and tedious task; further, most

(a) Input images of a large indoor scene under different lighting (showing 16 of 112 images).

(b) Average image



(c) Lighting design by a novice user, exploring our system for 5 minutes

Figure 3.1: *Light compositing by a novice user using our system*. Cafe: (a) A large indoor scene lit with a light in various positions. Photographers usually spend hours selecting and blending desirable parts from different images to produce a final image. Possible solutions, like an average image (b) produce unappealing results. We propose a set of basis lights and modifiers based on common photography goals that allow users to produce final results in a few minutes (c).

lights are accompanied by modifiers that also need to be adjusted, e.g., a snoot to restrict the lit area, or a diffuser to soften the shadows.

While post-processing the result in photo editing software is common, this step has almost no effect on the lighting which essentially remains the same as what was captured at exposure time. Recently, a few photographers have introduced a new workflow to control lighting that relies a lot more on the editing stage. Instead of a single photo with many light sources, they take many photos with a single light located at different locations each time. Then, they load all the images as layers in photo editing software and carefully composite the images to produce the final image. There are several advantages to this workflow accounting for its increasing popularity.

- First, capture sessions are shorter, easier to set up, and require less equip-

ment.

- Second, this workflow permits considerable control by enabling arbitrary layering and post-exposure adjustment over all the lights, allowing room for experimentation. For instance, one can easily control the region affected by a light source with a mask and set its intensity with a brightness adjustment.

This new process is fundamentally different from the traditional one because the capture session is not concerned with directly producing a visually pleasing image, it only seeks to record useful data for the later editing step. From this perspective, it is related to recent work in computational photography such as coded apertures, e.g., [92; 11], and lightfield cameras, e.g., [112; 1], in which computation is an integral part of the image formation process. This motivates us to name this modern approach *computational lighting design*. See [81; 69] for examples of this workflow from professional photographers that inspired us (unaffiliated with the project). These examples demonstrate the use of this workflow in architectural photography, and product photography.

However, one major disadvantage of this workflow is that it is quite cumbersome even for experienced photographers. When the number of images grows to several tens, or even above a hundred, navigating the corresponding layer stack becomes impractical. Further, with large scenes, most images show the main subject mostly in the dark with only a small part lit (see Figure 3.1a). Finding the useful images in the stack, setting their relative intensities, and blending them together to get visually pleasing results are highly challenging tasks that require advanced photography and image editing skills.

3.2 Motivation and Approach

Our objective is to assist photographers with creating a compelling lighting environment for a scene. In this section, we first describe how photographers work with lighting. This motivates our approach to designing lighting in this workflow.

3.2.1 Computational Lighting Design

With the traditional approach to lighting, photographers set up lights so that they fire simultaneously to produce the desired lighting. Then, illumination is essentially left untouched during post-processing. In comparison, for the computational workflow in which we are interested, photographers capture many images under different illuminations. The setup is often as simple as a single flash light moved to a different location between each shot. This has numerous advantages in terms of cost and mobility compared to the many lights used for studio lighting. Most importantly, the goal of the capture session is different. Whereas the traditional approach is concerned with the final result, the new computational workflow is about capturing useful "building blocks." Each photo aims to illuminate a portion of the scene in an interesting way that may be used later in combination with other images. The main objective is good coverage of the scene.

After the capture session, all the images are loaded as layers into image editing software. For each region, photographers select the desired appearance by editing the image alpha channels and ordering the layers appropriately. In

parallel, each layer can be edited, for instance, to increase its brightness, which is equivalent to having a more powerful light source, but with all the advantages of using editing software, such as instant feedback and unlimited undo. Only when this process is done, that is, after all the adjustments and after combining the layers according to their alpha channels, is the final image produced. Further editing may occur, for instance to remove unwanted elements, but this is not in the scope of our work. See [81; 69] for examples.

3.2.2 Objectives of Photographic Lighting

There are many ways to illuminate a scene in photography. We found a few recurring trends based on interviews with professionals [82] and field reports [81; 69].

Discrete Reasoning. Photographers think of lighting as the discrete combination of a few standard configurations. For instance, the *key light* illuminates the main subject, and the *fill light* is aimed at shadows to control how dark they are. While the exact setups depend on each photographer and scene, decomposing illumination into a small number of objectives is standard practice.

Curves and Lines. Photographers identify a few important geometric features of the scene that they seek to accentuate in the final result. These features are typically occluding contours that separate two objects at different depth, surface discontinuities such as creases, and curved regions that are characteristic of the object's shape. To emphasize these features, photographers set the illumination up so that a highlight falls on one side and a shadow on the other.

Light Quality. Photographers seek a "good light". While the concept is elusive, a few properties stand out. Harsh highlights and hard shadow boundaries are undesirable because they tend to distract from the scene. Overly dark shadows are also to be avoided because they hide the scene content.

We propose a few options to mimic photographers' solutions to these issues. We introduce *basis lights* that address a few well-defined goals. For example, to help users accentuate scene edges and curved regions, we first analyze the input images to identify these features and then propose an energy function that favors high contrast around them. This defines what we call the *edge light*. We simulate area light sources with several point sources to soften the highlight and shadows. We offer a *fill light* to control the darkness of the shadows, and let users restrict the extent of a light, which can be useful to prevent undesirable highlights and shadows.

3.2.3 Our Approach

We propose an approach inspired by the photographers' workflow described in the previous two sections. First, we describe the input data. Then, building upon our observations about the types of lights used, and lighting practices, we propose a set of basis light sources and controls that assist users in achieving powerful effects.

Input Photos. We use input data similar to what photographers capture, that is, a few tens or more photos taken from a fixed viewpoint and with a different

lighting configuration each time, typically using a single flash light. We assume that the light sources are approximately uniformly distributed in the scene and that their power is approximately constant. Further, we assume that the input set of images is white balanced with respect to the color of the flash light, i.e., the light source appears white in the input photos. For the datasets that we acquired ourselves, we used a remotely triggered flash unit and moved it at a different position after each shot, covering the entire scene in about 100 pictures. This is a rather mechanical process, where the main goal is to get a good coverage of the whole scene, with no particular planning. For a single data set, we spent around 20 minutes photographing it. We put a camera on a tripod and walked around with a remotely triggered flash. Exposure was fixed so that the flash dominated the other lights. For the Library scene, (Fig. 3.9), we also took a few images with longer exposure so that the outside is visible. We tried to keep ourselves out of the shots, but in the occasional pictures where the equipment/photographer was visible, we manually masked it out, so that those regions are not considered later. This pre-processing step has to be done once.

Basis Lights. We propose an *edge light* that emphasizes edges and curved regions in the scene, a *diffuse color light* that emphasizes the underlying colors, and a *fill light* that provides more even illumination. For each of these lights, we formulate an energy function that models the objective, e.g., large gradients that align with the main scene features for the *edge light*. Minimizing the energy gives us a set of coefficients that we use to combine the input images.

Modifiers. We also introduce controls to mimic the effects of standard light modifiers. For instance, we embed the input images into a weighted graph based

on their similarity and apply a diffusion process on this graph to modify a given set of coefficients to approximate the shadow-softening effect of an umbrella or a soft box. Other modifiers include *per-object* and *regional modifiers* that let us control the lighting for objects, like using a snoot, or to change the relative lighting of foreground vs. background.

The User Process. The user starts from a base image, for example the average of the image stack or a single image from the stack, and then they edit the stack using basis lights and modifiers. They first try to arrive at some globally reasonable solution, and then further refine the scene to identify either objects or parts of the scene that need more attention through edge enhancement, more light, or other lighting effects. See [20] for example sessions.

In allowing the user to pick individual objects, and applying an optimization of lighting for that particular object, we introduce inconsistent lighting in a scene. However, this is acceptable based on perceptual research about human insensitivity to lighting inconsistencies [114].

We evaluate our approach on several test cases demonstrating that it enables the design of sophisticated lighting environments with a small set of meaningful degrees of freedom, instead of a complex physical setup or the tedious manipulation of tens of layers. We further demonstrate the ease of lighting for novices and professionals using our basis lights in this new workflow.

3.3 Basis Lights

We propose a set of *basis* lights, which we relate to standard photography practices. Some of those lights correspond to actual lighting scenarios, commonly used in photography. The *fill light*, directly corresponds to lights used by photographers. Basis lights, like the *edge light* and the *diffuse color light*, address standard objectives such as emphasizing edges and curved areas, and revealing the diffuse color of objects, respectively.

We find the basis lights through an optimization scheme, that looks for the best linear combination of the input images, such that a certain objective is minimized. We first introduce some notation, and then describe the objective functions for our three basis lights: *fill*, *edge*, and *diffuse color*.

Standard Definitions and Notation. Throughout the paper, we use $\mathbf{I}_i(p)$ to denote the RGB components of a pixel p in the i^{th} input image. We work with sRGB values that are not gamma compressed, i.e. we apply inverse gamma correction by assuming 2.2 gamma. We refer to the intensity of a pixel as $\bar{I}_i(p) = \text{dot}(\mathbf{I}_i(p), (0.2990, 0.5870, 0.1140))$, defined as a weighted average of its RGB channels. We name N the number of input images. We use $\mathbf{W} = (1, 1, 1)^\mathsf{T}$ for the white color. We rely on angles between RGB vectors to reason about color saturation. We use the notation $\angle(\mathbf{C}_1, \mathbf{C}_2) = \arccos(\text{dot}(\mathbf{C}_1/\|\mathbf{C}_1\|, \mathbf{C}_2/\|\mathbf{C}_2\|))$ for the angle between the two colors $\mathbf{C}_1$ and $\mathbf{C}_2$. In several instances, we use a weighting function $w_i(p) = \bar{I}_i(p)/(\bar{I}_i(p) + \epsilon)$ that varies between 0 for low values of $\bar{I}_i(p)$ and 1 for high values. This function is useful to reduce the influence of dark pixels that are more noisy. In all our experiments, we use $\epsilon = 0.01$, assuming that the RGB channels range between 0 and 1.

(a) Average image (b) Weighted average

Figure 3.2: *Our* fill light*: average vs weighted-average*. Compared to the average (a), the weighted average (b) produces more even illumination , which we use as a *fill light*.

3.3.1 Fill Light

The role of the fill light is to provide ambient lighting that gives approximately even illumination everywhere. This is the light that illuminates the shadows, i.e., it controls how dark they are. Since we assume that the input lights are roughly uniformly distributed, we could use the average of all the input images, $\frac{1}{N}\sum_i \mathbf{I}_i$. However, since the light distribution is not perfectly uniform, the average may exhibit some large intensity variations. We improve over this simple average by giving more importance to bright pixels using the w_i weights, which reduces the influence of dark noisy pixels:

$$\mathbf{I}_{\text{fill}}(p) = \frac{\sum_i w_i(p)\,\mathbf{I}_i(p)}{\sum_i w_i(p)} \tag{3.1}$$

where i is the index over all images. Figure 3.2 compares the simple average image to our actual fill light $\mathbf{I}_{\text{fill}}$ using the weighted average.

3.3.2 Edge Light

As discussed in Section 3.2.2, photographers often seek to emphasize the main edges and curved areas in the scene. A common approach is to position the lighting such that it creates a tonal variation around those regions of interest. In particular, highlights and shadows are among the main building blocks through which photographers achieve this [77]. We define the *edge light* to assist them with this task. We proceed in two steps; first, we analyze the input images to identify the features to accentuate, and then we linearly combine the input images to emphasize the detected features, the mixture coefficients being a solution to an optimization problem that we define.

We define the features that we want to emphasize as edges in the input images that look persistent under the changing lighting conditions. Those can be due to geometric discontinuities in the scene, or more persistent highlights and shadows, which generate discontinuities in the observed images. However, occasional hard shadows and sharp highlights also produce image discontinuities but we are not interested in them, since creating highlights and shadows that compete with the main features of a scene is something photographers try to avoid [77].

The key observation of our approach is that main edges of the scene are always located at the same place in the image, whereas discontinuities due to occasional highlights and shadows move depending on where the light source is. By observing the statistics at a given location, we can differentiate between a persistent scene feature and ephemeral edges due to occasional illumination effects. The former appears consistently in all images while the latter is only present once or a few times, i.e., it is an outlier. Our approach builds upon this observation and uses robust statistics to extract a map of the main scene

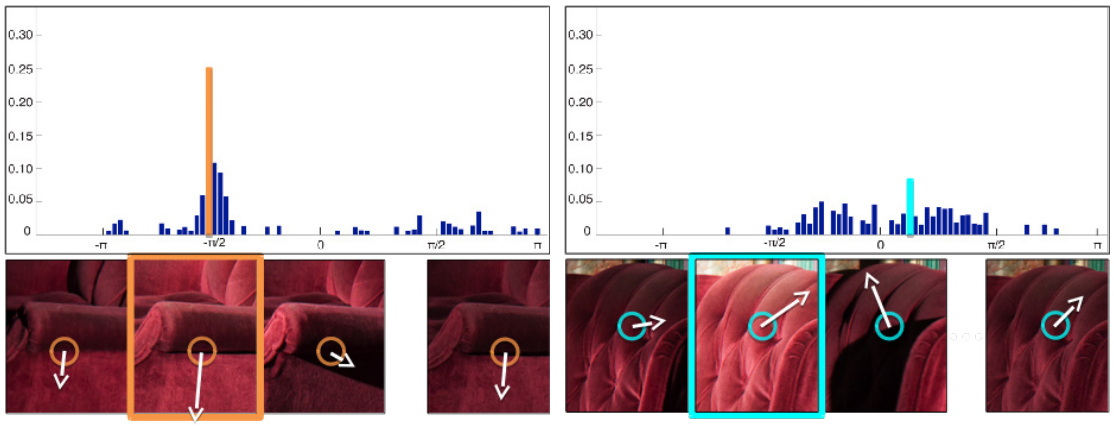
features. We tested a few options such as computing the robust max or the median gradient at each pixel. However, we found that the solution that we present below performs better for our goal of emphasizing persistent edges. We compare against the robust max and median gradients in Figure 3.4.

Our approach uses the fact that edges due to occasional highlights and shadows have an inconsistent orientation. To exploit this phenomenon, at each pixel, we build the histogram of the gradient orientations. In practice, we use bins that span 5°. To prevent dark noisy pixels from perturbing the process, we weight the contribution of each pixel using its w_i weight. Also, to differentiate between flat regions and vertical edges, small gradients of magnitudes less than 0.001 are handled separately. Then, within the largest bin, we pick the gradient of maximum amplitude. Intuitively, this process selects the strongest gradient that aligns with the most persistent orientation at every pixel. This gives us a target gradient map $\mathbf{G}$. We seek the *edge light* as a linear combination of the input images: $\mathbf{I}_{\text{edge}} = \sum_i \lambda_i \mathbf{I}_i$. To find the mixture coefficients λ_i, we minimize the following weighted least-squares energy function:

$$\arg\min_{\{\lambda_i\}} \sum_p h(p) \left\| \nabla\Big(\sum_i \lambda_i \mathbf{I}_i(p)\Big) - \mathbf{G}(p) \right\|^2 \tag{3.2}$$

where, the per-pixel weights $h(p)$ give more influence to pixels that have a peaked orientation histogram, that is, pixels that have a well-defined orientation. We define h by normalizing the histograms to 1 so that we can compare them across pixels, and picking the value of the largest bin at each pixel. Figure 3.3 illustrates this process.

Discussion. The effect of our *edge light* is not to avoid all shadows and highlights, which would be undesirable from a photographic point of view. By



(a) Strong geometric edge (b) Flat texture edge

Figure 3.3: *Our* edge light*: histogram of gradients orientations*. We compute a histogram of gradients orientations for every pixel across all different lighting conditions. In (a) we show an example of a pixel that corresponds to a strong geometric edge, where we have a well defined orientation across different lighting conditions. In (b) we show an example of a flat, texture-like region where the orientation is less well defined. We use the height of the largest bin as a confidence value of the measurement, i.e. the $h(p)$ quantity defined above. For each pixel, we pick as representative the gradients from one of the input images, where the orientation matches the direction of the largest bin. For the example pixels above, we mark those representative images with orange and blue colors.

optimizing for lighting that maximizes gradients that align with the main scene features, it favors highlights and shadows that align with them (Fig. 3.6a. 3.7). This behavior is reminiscent of the line drawing technique of Judd et al. [79] who motivate their approach by characterizing the lines worth drawing as the ones that appear across multiple lighting configurations. From this perspective, our *edge light* seeks to produce an image in which discontinuities would be a good line drawing of the scene.

(a) Robust max (b) Median (c) Ours

Figure 3.4: *Comparison of our gradient map against other alternatives*. We compare our gradient map against two other alternatives for emphasizing edges. Using the robust max gradients can produce results that emphasize distracting elements, like the shadows on the seat of the chair (a). The median gradients are more robust to occasional shadow boundaries, but other edges are also de-emphasized (b). In comparison, our proposed gradients better capture the main scene features and their orientations.

3.3.3 Diffuse Color Light

The objective of the *diffuse color light* is to emphasize the base color of objects. To reason about scene colors, we use a simple diffuse+specular image formation model in which the diffuse color can be arbitrary, and the specular color is the same as the light color.

First, we consider the case of a colorful object. We seek to design an energy function that favors images in which the diffuse component is strong compared to the specular reflection. Similar to Tan et al. [138] we observe that because the specular component is white, the stronger it is, the less saturated the observed color is. Formally, we consider a pixel $\mathbf{I} = d\mathbf{D} + s\mathbf{W}$ where $\mathbf{D}$ is the diffuse color and d its intensity, $\mathbf{W} = (1, 1, 1)^\mathsf{T}$ the white color, and s the specular intensity. We characterize the saturation by the angle $\angle(\mathbf{I}, \mathbf{W})$ between the observed color $\mathbf{I}$ and the white color $\mathbf{W}$. For a fixed d value, this angle decreases when s increases.

This motivates the following energy term:

$$\arg\max_{\{\lambda_i\}} \sum_p \hat{w}(p) \angle\Big(\sum_i \lambda_i \mathbf{I}_i(p), \mathbf{W}\Big) \tag{3.3}$$

where, $\hat{w}(p) = \sum_i \lambda_i \mathbf{I}_i(p) / (\sum_i \lambda_i \mathbf{I}_i(p) + \epsilon)$ is a term that prevents selection of linear combination of lights that produce dark pixels, that tend to be noisier. With Equation 3.3, we seek a linear combination of input images that maximizes the angle with the white color, while preventing the selection of dark pixels.

This approach works well for colorful objects, that is, when $\angle(\mathbf{D}, \mathbf{W}) \gg 0$. However, this term is less effective for objects of neutral color, i.e., when $\mathbf{D} \approx \mathbf{W}$. For neutral colored objects, changes in specular intensity create only small angle variations. And most importantly, the optimization becomes sensitive to colored inter-reflections. For such neutral objects, even the small change of saturation generated by light reflecting off nearby colored objects has a significant impact on the energy value. In our experiments, using the previous energy term alone produced images in which gray objects have strong colored inter-reflections, which looked unpleasant. We address this issue with a second energy term based on the observation that the average image lowers the contribution of rare illumination effects, such as strong inter-reflections and highlights, which are undesirable features based on our *diffuse color light* definition. We design an energy term that encourages similarity between the average image and our result:

$$\arg\min_{\{\lambda_i\}} \sum_p \angle(\sum_i \lambda_i \mathbf{I}_i, \frac{1}{N} \sum_i \mathbf{I}_i) \tag{3.4}$$

Since we seek to use this term only for neutral colors, else the solution will tend towards the average, we use a balancing term that equals 1 only for neutral

colors and has lower values otherwise:

$$\alpha(p) = \exp(-\angle(\frac{1}{N}\sum_i \mathbf{I}_i(p), \mathbf{W})^2/2\sigma^2)) \tag{3.5}$$

with $\sigma = 0.5$. Figure 3.5 shows the significance of this term.

Putting the two terms together, and realizing that the goal is to maximize Equation 3.3, but minimize Equation 3.4, we obtain the coefficients of the *diffuse color light* $\mathbf{I}_{\text{diffuse}}$ by minimizing:

$$\arg\min_{\{\lambda_i\}} \sum_p \Big[\alpha(p)\ \angle(\sum_i \lambda_i \mathbf{I}_i(p), \frac{1}{N}\sum_i \mathbf{I}_i(p)) \\ -(1-\alpha(p))\ \hat{w}(p)\ \angle(\sum_i \lambda_i \mathbf{I}_i(p), \mathbf{W})\Big] \tag{3.6}$$

We minimize this function using an interior point method with finite differences to approximate the gradients (see Figures 3.5 and 3.6b).

(a) Saturation term only (b) Both terms

Figure 3.5: *Evaluation of the diffuse energy terms*. The color of a neutral object, like the white ceiling, can be dominated by interreflections from nearby objects. We propose per-pixel weights that encourage more similarity between the average image and our result for pixels that look neutral on average.

Summary. In summary, we have designed energy terms for each of the three basis lights: *fill light*, *edge light* and *diffuse color lights*. We solve for the linear combination of images, and their corresponding weights, that minimize the energy terms.

(a) Edge light (b) Diffuse color light (c) Fill light

Figure 3.6: *Our basis lights applied on the red chair*. Our *edge light*, optimized for the red chair puts more emphasis on the main edges (a), whereas the *diffuse color light* reveals more of the deep red colors (b). We use the weighted average image as a *fill light* that provides more even illumination (c).

3.4 Modifiers

In addition to the basis lights described in the previous section, we also introduce *modifiers* that alter the properties of these lights in ways that mimic standard practices in photography. The *per-object lighting modifier* restricts the light's extent, the *regional lighting modifier* balances the illumination intensity between different regions of the scene, and the *soft lighting modifier* modifies the lights so that they produce softer shadows and highlights.

3.4.1 Per-Object Lighting Modifier

This is the simplest of our modifiers. It is inspired by equipment like snoots that photographers use to control the spread of lights. We let users select objects in the image. Then, we compute the *fill*, *edge*, and *diffuse color lights* as described in the previous section. The only difference is that we only consider the pixels within the selected object. Users can then locally mix these three lights. To ensure smooth blending with the rest of the image, we apply a cross bilateral filter [51; 119] to the binary selection, using the intensities of the current result as the guiding image. We use the fast cross bilateral filtering by Paris et al. [115] to transform the binary selection into weighting masks that respect the edges for each of the basis lights. Then, in our interactive interface we approximate the weighting mask of the current combination of basis lights by linearly blending their corresponding masks. This produces a continuous mask that "snaps" at the main scene edges, which yields satisfying results. We also experimented with simple Gaussian blur but this generated severe halos, and also with multiscale blending [30] but color artifacts appeared.

3.4.2 Soft Lighting Modifier

This modifier aims for an effect akin to umbrellas and soft boxes, that is, simulating area light sources that produce soft shadows and highlights. Our strategy is to approximate an area light source by a set of nearby point sources. However, in our context, the position of the light sources is unknown a priori.

We address this problem with an approach inspired by Winnemöller et al. [152] who showed that for two images taken from the same viewpoint with

two different point lights, the spatial distance between the lights is correlated to the difference between the observed images: close light sources generate similar looking images and distant sources create different images. They demonstrate that this can be used to recover the positions of lights on a sphere. However, they mention that more general configurations are challenging.

For our *soft lighting modifier*, we build upon the same correlation between light position and image appearance, and sidestep the difficulties stemming from general light configurations by directly modifying the light mixture coefficients without explicitly recovering the light positions. We implicitly embed the input images into a weighted graph based on their similarity and apply a diffusion process on this graph to modify a given set of mixture coefficients $\{\lambda_i\}$ to approximate the effect of soft box lighting. We define a $N \times N$ matrix $\mathbf{S}$ with coefficients:

$$S_{ij} = \exp(-\|\mathbf{I}_i - \mathbf{I}_j\|^2 / 2{\sigma_s}^2) \tag{3.7}$$

and a vector $\Lambda = (\lambda_1, \ldots, \lambda_N)^\mathsf{T}$. Intuitively, multiplying Λ by $\mathbf{S}$ spreads the contribution of each light to the nearby sources using the image similarity $\|\mathbf{I}_i - \mathbf{I}_j\|$ as a proxy for the spatial distance. The σ_s parameter controls how far the intensity of each light is diffused. As is, this approach does not preserve the overall illumination intensity. We experimented with a few options and found that a simple global rescaling works well and is computationally inexpensive. To summarize, our *soft lighting modifier* is defined as:

$$soft_{\sigma_s}(\Lambda) = \frac{\|\Lambda\|}{\|\mathbf{S}\Lambda\|}\mathbf{S}\Lambda \tag{3.8}$$

To gain intuition, we observe two extreme σ_s settings. For $\sigma_s \to 0$, the modifier does nothing as one would expect, that is, point light sources remain as is. And for $\sigma_s \to \infty$, the output coefficients are all equal, i.e., the output is the average

of the input images, which is in some sense the largest area light that we can simulate with our data. Other values of σ_s provide approximations to area light sources of intermediate sizes as shown in Figure 3.7c.

(a) Input images (4 out of 129)

(b) Edge light

(c) With the soft lighting modifier

(d) Adding fill light

Figure 3.7: *Our* edge light *and* soft lighting modifier *applied on the basket scene*. The *edge light* emphasizes the main edges of the scene, which tends to produce sharp highlights and deep hard shadows (b). Applying the *soft lighting modifier* softens the highlights and the shadow boundaries and keeps the shadows dark (c). The fill light has a complementary effect; it brightens the shadows and keeps their boundaries and the highlights sharp (d).

3.4.3 Regional Lighting Modifier

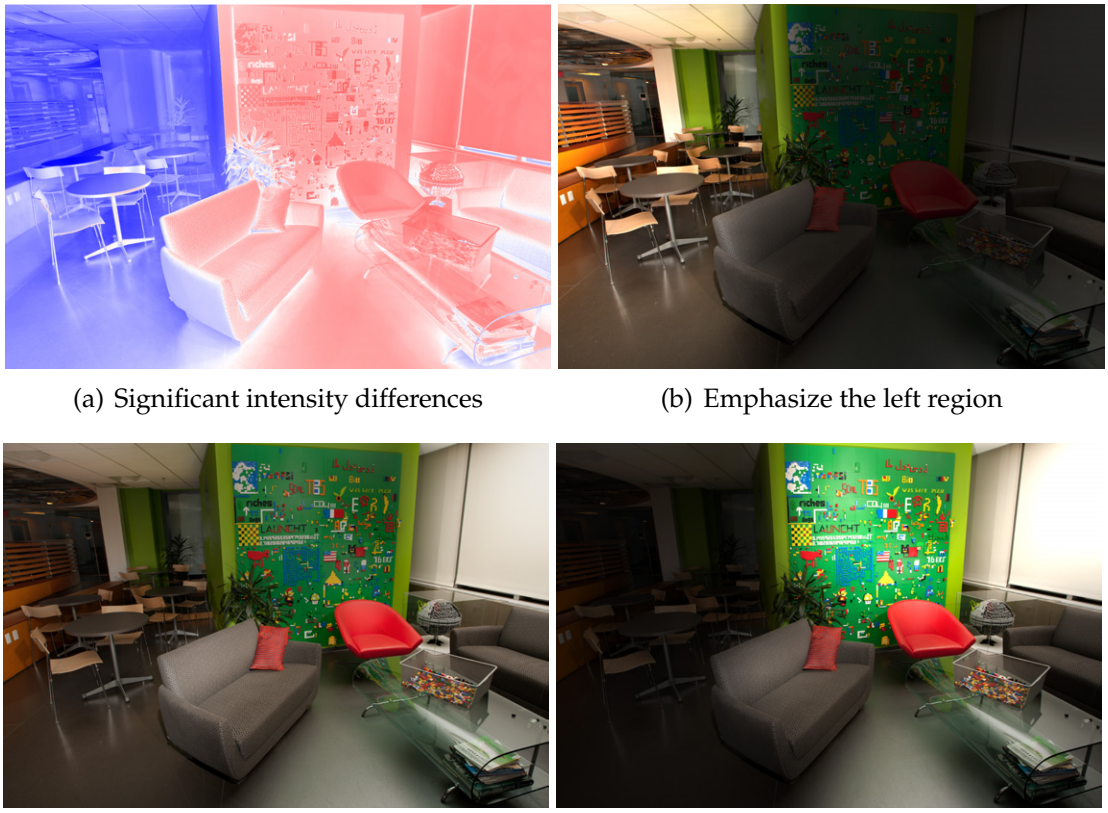
Photographers carefully balance light intensities in a scene to either emphasize a specific region or to do the opposite. We propose our *regional lighting modifier* to assist this process. We seek to provide a simple way to balance the lighting across the scene at a coarse level. Fine-grain adjustments can be made with our *per-object modifier*. Our observation is that the PCA decomposition of the input images extracts the main modes of variation of the illumination. In particular, for scenes that can be decomposed into "regions" illuminated independently of each other, e.g., foreground versus background, or left versus right, the first PCA component captures this structure well.

We build our modifier upon this observation. Since PCA assumes an additive mode of variation, and light interaction with materials is multiplicative, we work in the log domain. Further, because we seek to only modulate pixel intensities without altering their color, we work with the intensity images $\{\bar{I}_i\}$. First, we estimate the first PCA component P of the log intensities $\{\ln(\bar{I}_i)\}$. To avoid perturbing the overall image intensity, we enforce a zero mean onto P by defining:

$$\hat{P} = P - \frac{1}{N}\sum_{p} P(p) \tag{3.9}$$

As is, $\hat{P}$ often exhibits undesirable shadow boundaries. We remove them by applying a cross bilateral filter [115] to $\hat{P}$ with the current global result (a.k.a. the smooth blending of all locally and globally optimized basis lights) as the guiding image. Finally, we create a map $M = \exp(\beta\hat{P})$ where β is a user parameter controlling the magnitude of the effect: $\beta = 0$ does not alter the result, $\beta > 0$ emphasizes the regions where $\hat{P} > 0$ by making them brighter and the rest darker, and $\beta < 0$ has the opposite effect, i.e., it emphasizes the $\hat{P} < 0$ regions. The M map is multiplied pixel-wise to the current result to obtain the final result. The

effect of this modifier is shown in Figure 3.8.



(a) Significant intensity differences (b) Emphasize the left region

(c) Original lighting ($\beta = 0$) (d) Emphasize the right region

Figure 3.8: *Our* regional lighting modifier *applied on the cafe scene*. Our *regional lighting modifier* can be used to move the emphasis between the two regions that show significant intensities differences in the input images. We detect these regions automatically, by looking at the first PCA vector of the input intensities (a).

3.5 Results

We now describe our implementation and the user interface of our prototype. Then, we demonstrate our approach on a variety of scenes, and show comparisons with related work.

Implementation. We use a combination of C++ and Matlab in our prototype system. The part that optimizes the basis lights is an offline process, which we implemented in Matlab, since it was less time critical. We use Matlab to solve the constrained linear and non-linear optimizations problems that correspond to our basis-light objectives. The timing for this step depends on the size of the regions, but it can take from a few seconds up to 10 minutes, when the region is the whole image. However, this offline process can be done in parallel for all pre-segmented objects and basis lights. In our user interface, in order to achieve interactive speeds, we do the image blending on the GPU.

In Table 3.1 we describe each scene, the resolution of the input images, the number of objects that we segmented, and the time for optimization, both per object and per image. We pre-compute the basis lights for the set of pre-segmented objects and the whole image so that users do not have to wait. Last two columns are max-per-object and full image timings.

User Interface. We now briefly describe our prototype interactive interface where users can explore a variety of lighting designs. For every object, the pre-computed basis lights (*edge*, *diffuse color*, *fill*) can be mixed by changing their relative contributions with sliders. Depending on the user's preferences, this can be done in two ways: (1) by preserving the overall intensity using normalized weights that add to one, or (2) by not normalizing. This is controlled by the checkbox "Keep intensities constant", visible in [20]. In addition to that, we also let users control a simple exposure slider.

For every local object (selected by clicking on the object, and using the checkbox "Show local lighting"), the sliders control the object-based lighting. Segment-

ing the image into regions is beyond the scope of this paper, and we assume it is done by the user. When a region is first selected and the local lighting enabled, the initial exposure is set to match that of the global lights.

Users can also interactively change the strength of the *soft lighting modifier* to control the softness of the shadows and highlights (Fig. 3.7c). To ensure a consistent behavior across scenes, we normalize the image differences in Equation 3.7 so that the largest one is 1. To enable interactive editing, for each light, we precompute the effect of the modifier for 100 regularly spaced values of σ_s between 0.01 and 10. At run-time, we linearly interpolate the values of the two samples that are closest to the requested parameter.

Our last slider controls the regional modifier. This allows users to interactively change the emphasis in the scene, by smoothly modifying the per-pixel exposure, through the parameter β (Fig. 3.8).

3.5.1 Image Results

We now demonstrate our results on a range of scenes.

Cafe. In Figure 3.1, we show results for a larger interior scene that has a variety of objects with different shapes and materials. For example, the red chair has strong glossy components, and at the same time, a deep red color. Our *edge light* better reveals the shape of the chair, by emphasizing highlights (Fig. 3.6a). Our *diffuse color light* shows more of the deep red color of the chair (Fig. 3.6b). Our system allows novice users to easily explore different lighting designs, by mixing the basis lights in various proportions, globally or per pre-segmented objects

(Fig. 3.1c). In [19] we show 6 more results, which demonstrates that even novice users can produce non-trivial variations using our system.

Library. The Library in Figure 3.9 shows an example of an indoor room, where an outside view is also visible through a window. We gave our data set to a professional photographer, with instructions to create a result that looks good to him. Figure 3.9b shows his result, achieved in 20 minutes, using the full power of Photoshop. We gave the same scene to novice users, who were able to explore various lighting designs, using our system. For example, in Figure 3.9c our user used more *diffuse color light* and locally decreased the exposure to better show the black color of the sofa. *Diffuse color light* was also used to emphasize the deep red of the armchair. Other results available in [19] show that even novice users are able to produce nontrivial variations.

House. Figure 3.10, third row, shows an example of a big outdoor scene. Lighting this scene by walking around with a single light is particularly useful and one of the few options at this time of the day, when the ambient lighting is low. In [19] we show the two regions found by our *regional lighting modifier*. For this scene, the regions roughly correspond to objects closer to the ground, and objects closer to the sky, which exhibit significant intensity differences in the input data set. Users of our system can use this for an artistic control to add more contrast between the lighting in those two regions. In [20] we show all other steps used to generate this result.

Basket. Figure 3.7b shows the result of our *edge light*, applied to the entire scene. One of the main edges in this scene separate the foreground objects from

the background table, and our *edge light* emphasized those further (Fig. 3.7b). However, in this cluttered scene, occasional shadows can arise for many lighting positions, producing distracting shadows in the *edge light* solution. In our system, users can interactively apply the *soft lighting modifier* to simulate a larger area light source. This can be used to soften the hard shadows on the table, cast by the fruits. The *soft lighting modifier* can also be used to soften the highlights on the tomato (Fig. 3.7c). A *fill light* can be used to add even illumination to the whole scene, which provides more details in dark regions, including shadows (Fig. 3.7d). We want to stress the difference between the *soft lighting modifier* and the *fill light* Although at their extreme values they both produce some version of the average image, their intermediate effect is different. The *soft lighting modifier* simulates a gradual increase of the current light source, by combining nearby lights and weighting them appropriately, whereas the fill light cross fades between the current illumination and the average image.

Los Feliz. In Figure 3.11, we show results for a stack of images that we received from a professional photographer (naive to our research). This is a set that was not captured or processed by us. The photographer also gave us his preferred final result for that scene. We demonstrate in [20] that we were able to achieve a similar effect in a few seconds, rather than half an hour. In particular, our *edge light* applied to the whole scene has captured the gist of the result produced by the professional.

Dependency on the Input Images. In Figures 3.12 and 3.13 , we evaluate the dependency of our system on the number and quality of the input images. Figure 3.12, row 1 shows the results of our *edge light* optimized for 11 different

segments, such as the red chair, the central sofa, the green wall, etc. In [19] we show all pre-segmented regions. Figure 3.12, row 2 shows the results of our *diffuse color light*, optimized for the same set of segments. We conducted two tests based on the image selection. In the first test, we randomly select 5 and then 15 images from the original data set, (Fig. 3.12b,d). In the second test, 5 and then 15 images were carefully selected, so that they contain useful features such as lighting that emphasize edges, or lighting that reveals the underlying material, (Fig. 3.12c,e). First, our evaluation suggests that a small number of random shots are insufficient to produce good features across all parts of this large scale scene, (Fig. 3.12b,d). Second, even if carefully chosen, if the number is too small (5) it is not enough for a scene of this size, (Fig. 3.12c). Finally, post-hoc it was possible for us to find 15 images that would produce reasonable basis lights, (Fig. 3.12e). So, it might be possible for a person with a lot of experience to make use of our basis lights with a smaller number of carefully planned shots. However, even experienced photographers use the earlier work flow (capturing many images) because they are worried they could miss something and do not want to take chances. Further, in Figure 3.13 we show that the quality of our *soft lighting modifier* is more closely related to the number of input images. The reason is that a more uniform sampling of the lighting in the scene produces more close-by lights. These are needed for the gradual simulation of large area lights, computed by our *soft lighting modifier*.

Comparison with Related Work. In Figure 3.10, we compare with two other systems that share some common goals with ours. Exposure Fusion [108] expects a sequence of images with different exposures, and produces a single, well-exposed image. Their system was primarily designed for scenes with constant

lighting and high dynamic range. In comparison, our input image sequences with dynamic lighting moving around the scene introduce new challenges. When applied to our datasets, Exposure Fusion produces unsatisfactory results (Fig. 3.10, second column). Further, their results appear flat since they seek to expose the entire scene, including the shadow areas, equally well.

Figure 3.10, third column, shows results from MLIC [55], which is more closely related to our goals, as they work on similar input data: a static scene under dynamic lighting. However, they propose an automatic system, that emphasizes details over different scales. As a result, they also tend to flatten the look of the image, and decrease its realism by mixing information from different scales, under different lighting. While this works well on single objects as demonstrated in the original paper, it is less successful on large scenes. In comparison, we generate plausible pleasant lights, based on common photography practices.

3.5.2 User Validation

To validate our contribution we evaluated our idea with novice users and expert photographers.

For the expert evaluation, we asked 3 experienced photographers to comment on the merits of our basis lights. We sent them two Photoshop projects: one containing the full stack of input images, and a second project with a reduced set of images that represent our basis lights, optimized for a few pre-segmented objects. We asked each of them to spend some time working with both image stacks. The goal of this study was to see whether our basis lights could give them a good starting point, while saving them time wasted in finding and blending

features from the original images. They were all enthusiastic about the lights, and reported that working with the reduced image stack made their workflow more efficient, compared to going through tens of input images. They also reported that our basis lights would be useful in their workflow, but sometimes they had to apply additional adjustments, like color balance and light levels to achieve a desired effect. However, this type of adjustments is orthogonal to our work. See [19] for their results using the full image stack and our reduced image stack.

We also tested our system with 7 novice users who had little or no experience with photography. The goal of this study was to show that our prototype system can allow ordinary users to explore different options and achieve sophisticated lighting designs in a couple of minutes. Figure 3.9b shows the result of a professional photographer using Photoshop with our reduced image stack. Figure 3.9c shows results on the same scene generated by a novice user interacting with our system for the first time. The task was not to match the solution of the professional, as Photoshop allows them to apply many adjustments, like nonlinear curves to increase contrast, which is orthogonal to our project. However, we showed the professional's result to our users, just for a minute, as an example of a good lighting design. We then asked them to explore solutions in an open-ended manner. Our system allowed users to rapidly explore different lighting designs and produce visually pleasing results (Fig. 3.9c). We show results of 6 other users in [19]. On average, users spend 15 minutes on this data set.

In [19] we show another evaluation on the "Cafe" scene, where we gave users 5 to 10 minutes to explore different lighting designs, using our system. Then, we asked them to switch to Photoshop and spend the same amount of time, using the full, unprocessed image stack. They were instructed to look for similar

Scene	Size	Images	Objects	Max (min)	Full (min)
Cafe	1.5MP	112	11	4	9
Library	1.0MP	83	13	3	8
Basket	1.2MP	129	9	1	9
House	1.5MP	149	6	3	10
Sofas	1.5MP	32	7	0.5	2
Kitchen	1.5MP	127	7	4	10

Table 3.1: Number of regions and time for optimizing our set of basis lights on those regions and the full scene.

features as the one they produced using our system. Our experiment showed that for novice users the results were poor; searching and blending features from the full stack of input images is a nontrivial task which prevents them from producing good results.

3.5.3 Discussion and limitations

Our approach is a user-driven creation process meant to help users create compelling images. However, not all slider configurations produce such images, e.g., if one uses only the fill light to illuminate the scene, the result will look dull. That said, our experiments show that even novice users are able to generate quality results.

In general, our results do not correspond to a physical setup. For instance, our *regional lighting* and *per-object modifiers* do not alter the illumination in a physical way. However, they are close to what could be produced using blockers, and our results look plausible. In addition, the core of our approach is based on linear

combinations of the input lights, which corresponds to actually turning on the lights at the same time with the appropriate intensity. This further contributes to generating plausible images.

Finally, image editing software offers virtually infinite control over the result whereas our approach covers a smaller design space. However, we argue that for most users, unbounded editing capability is actually a hindrance more than a help since it requires advanced skills and a lot of time, which is confirmed by our experiments. For the few users with mastery of advanced editing tools, we do not offer a complete set, but we envision that they would first use our approach to quickly obtain a satisfying result and, if needed, they would later refine it with standard photo editing software.

3.6 Conclusions

Lighting is critical to good photography. There is a new workflow emerging for lighting static scenes, where photographers capture many images of the scene with a single (or small set of) light(s) in different locations to create a set of input images that serve as data to a later compositing stage. This workflow gives the photographers a lot of flexibility in post-process, and is faster in time-constrained shots, but dealing with the many images after that is difficult.

We have introduced a set of optimizations to help the photographers assemble all these images into a small set of basis lights and modifiers that are easy and fast to user in an interactive editing session. The photographer can use these lights to achieve common photography goals like accentuating highlights, filling shadows, and emphasizing object color. Our studies with both novice and professional

users shows that this approach is a significant improvement over the traditional workflow.

There are multiple areas of future work. One possibility is to explore other types of basis lights related to common photography practices, such as rim lighting that emphasizes contours, or lighting that better reveals the glossy behavior of objects. Another interesting avenue for future work would be the development of an interactive system that could assist the acquisition process, by guiding the placement of lights that achieve a desired effect. We would also like to explore better optimization techniques for the basis light objectives, like multi-grid on GPU, which could make this step interactive. Finally, we believe that our techniques can find potential uses beyond consumer photography, such as lighting for stop-motion movies or museums.

(a) Library, 4 out of 83 images

(b) Result of a professional (20min in Photoshop)

(c) Novice user, using our system for 16min

Figure 3.9: *Evaluation with novice users*. Our evaluation with novice users shows that our system allows them to explore non-trivial lighting designs (c) in a short amount of time, comparable in quality to what a professional photographer would achieve, using the full power of Photoshop in 20 minutes (b).

(a) Average (b) Exposure fusion (c) MLIC (d) Ours

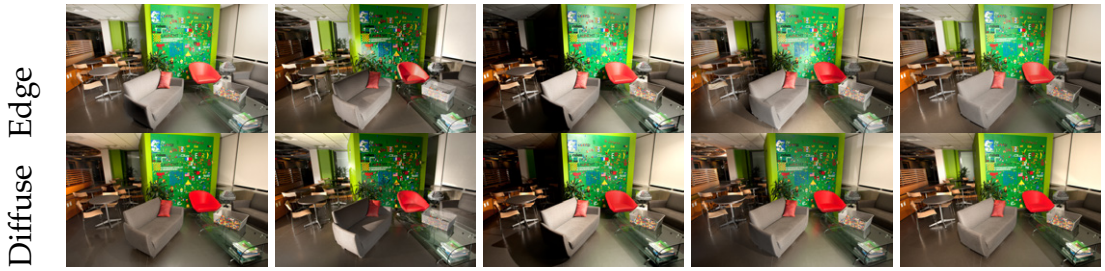
Figure 3.10: *Comparison against prior work*. Comparison of our user-driven system to two automatic systems, that share some goals with ours, shows that those systems produce less satisfying results for the type of scenes in which we are interested. Further, the goal of our system is to provide a simple set of controls, through which users can explore a variety of solutions, whereas these systems are mostly automatic.

(a) 4 out of 30 images

(b) Result by professional (30min in Photoshop)

(c) Result, using our system for 30sec

Figure 3.11: *Comparison on a prior data set (Los Feliz).* On a data set provided by a professional photographer (© Michael Kelley), our system provides a quick way to explore a reasonable solution, that captures the overall look and feel of the professional result. In [20] we show that combination of our global *edge* and *fill lights* capture many of the desired features, like background that is well lit and fill light on the table and chairs.



(a) All images (b) 5 random (c) 5 planned (d) 15 random (e) 15 planned

Figure 3.12: *Evaluation on the qualiy of results and the number of images*. Our *diffuse color* and *edge lights* optimized for 11 regions, using input data with different properties. In (a) we used the original data set (all images). In (b) and (d) we show the quality of the results that could be obtained with 5/15 randomly selected images. In (c) and (e) we demonstrate the results of our system with 5/15 carefully selected pictures. First, note that a small number of randomly selected images produces unsatisfactory results for many parts of the scene, (b) and (d). Second, even if carefully chosen, if the number is too small (5) it is not enough for a scene of this size, (c). In (e) we show that with 15 carefully selected images our basis lights can produce reasonable results.

(a) Initial lighting (b) Using 15 images (c) Using all images

Figure 3.13: *Evaluation of our* soft lighting modifier *and the number of images*. Our *soft lighting modifier*, applied to the lighting in (a) shows that a small number of images (15), even if carefully selected, do not provide enough close-by lights, needed for the gradual simulation of large area lights, (b). In (c) we show that the effect of our *soft lighting modifier* improves when used with the full image set.

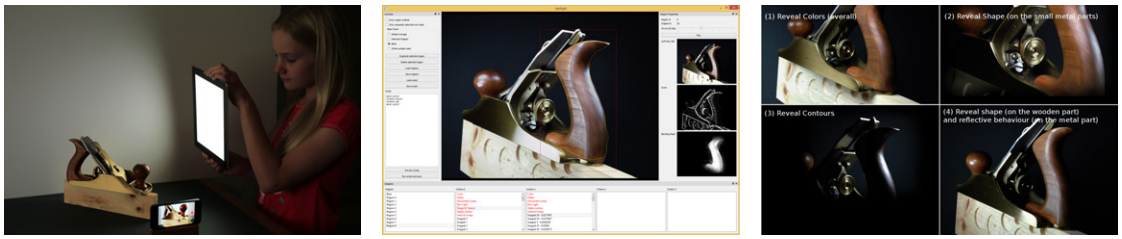
CHAPTER 4

DO-IT-YOURSELF LIGHTING DESIGN FOR PRODUCT VIDEOGRAPHY

This chapter describes our follow-up work on lighting design which extends our previous method for the case of product photography and videography, while further simplifying the acquisition equipment. We also focus on specular and glass objects which are often the target of product photography. Shiny and reflective materials provide challenging cases for our previous work, due to our sparse acquisition and discrete reasoning about the lighting in the scene. As a result of that, our basis lights produce unsatisfactory results that have distracting reflections and cross-fading of the input lights, as we show later in this chapter. In this chapter, we propose a new approach that starts with recording a video of a static object, while moving a fully lit tablet around it, which serves as a small area light source. The resulting video provides a couple of minutes of useful data, which we automatically split into short snippets that we later rank based on common goals in product photography and videography. The proposed system allows novice users to create sophisticated lighting designs for both still and video media.

4.1 Introduction

Popular online marketplaces like eBay, craigslist, and Etsy allow everyone to directly sell their own goods. As a consequence, product photography, a domain that used to be reserved to professionals, is now also needed by novice users. Further, the online nature of these platforms favors the sharing of videos and animated gifs. But producing a professional quality product video is challenging and requires a studio and specialized equipment for lighting. Professionals



(a) Our acquisition setup (b) Our user interface (c) Sample of video frames by a novice user

Figure 4.1: *Our new DIY method to assist lighting-design for product videography and photography*. We require minimal equipment from the user: (1) a video recording device, e.g., a smart-phone and (2) a single area light source, e.g., a fully lit table screen is sufficient. We show a typical setup and an example acquisition session in (a), where a novice user (11 years old) waves a tablet around an object with a smart-phone recording a video of the object being lit. After a short acquisition session, of around 4 minutes, we analyze the captured video and automatically split it into snippets which we further re-order based on common goals in lighting design. In (b) we show a screenshot of our prototype user interface, where users can explore a variety of options based on high-level lighting-design principles, e.g., find a rim light to emphasize contours, find a color-revealing light, or find a glitter-revealing light motion, etc. In (c) we show a few frames from a video sequence produced by a novice user with our interface.

carefully arrange lights and reflectors to emphasize the product's shape and material while achieving visually compelling video. Setting up such illumination requires time, money, and a great deal of expertise. For all these reasons, novices are not able to produce nearly as good results, and their videos typically look unflattering, and rarely do justice to the product.

Our work seeks to enable casual users to create quality product videos. We propose a simple do-it-yourself (DIY) setup with a video camera on a tripod

recording the product while the user waves a tablet around it for a couple of minutes. Although the recorded footage is too long and unappealing as is, over the length of the recording, there are always segments that look good. Our strategy is to identify these good *snippets* and assemble them into a short pleasing video that showcases the product.

To achieve this goal, first, we study professionally made product videos and the related literature and formulate design principles to guide our approach. For instance, video artists always use long light sweeps that create highlights moving consistently over the product. They also rely on a few typical standard illumination configurations such as rim lighting and highlights following the main edges of the object, e.g., [141].

We build analysis tools that aim to achieve these design principles. We define scoring functions to rank video segments according to characteristics such as the presence of rim lighting or their emphasis on edges. Robustness is a paramount requirement for these functions because the objects in which we are interested are often highly non-Lambertian, e.g., a transparent and refractive perfume bottle or a bejeweled watch, and many standard analysis algorithms fail on them. We show that our numerical schemes perform well on a wide variety of objects and materials despite these challenges.

With these scoring functions in hand, we present a graphical interface to let users select and assemble snippets to produce a video with sophisticated lighting effects. Our interface features a faceted search to browse the available snippets and allow users to combine several snippets to generate effects akin to what professionals get using complex multi-lights setups. It also provides tools to create a "base image" corresponding to the static illumination on top of which

the highlights move. Capturing the input footage takes about 10 minutes, and editing the final result with our interface about another 15 minutes. We show that novices can go through this process with minimal training and produce quality product videos that they would not be able to produce otherwise.

4.1.1 Overview

Our new workflow comprises three parts: acquisition, analysis, and compositing.

Acquisition. The user records a short video (3-4 minutes) by mounting a camera on a tripod and waving an area light source around the object of interest.

Analysis. The segments are analyzed for features such as the speed and direction of motion of lighting. These features are then used by various metrics to split the video into various snippets. The metrics aspire to capture design principles like highlighting contours, rim lighting, accentuating meso-structure, etc.

GUI and Compositing. Finally, the user explores the collection of extracted snippets, and composites and sequences them in our GUI to produce the final video.

4.2 Design Principles

Lighting designers balance many goals when setting up lights. They illuminate the object to show off its material, emphasize its shape, reveal subtle details, while also producing a visually pleasing video. Artists have acquired an intimate understanding of these objectives and their interactions, but we could not

find any formal comprehensive description of this craft. Instead, we analyzed professionally made clips such as [141] [140] and reinterpreted photographic guidelines in the context of product videos to formulate the following principles.

Lighting Properties. The majority of the clips that we analyzed use white area light sources. The lights are either fixed to create a base illumination, or move slowly along long smooth trajectories to generate highlights that move predictably. Further, it has been demonstrated [60] that area lights are better than point lights for material perception; a swept light integrated over time creates an area light effect.

Video Structure. Product videos are typically made of 4 to 8 shots, each lasting typically between 2 and 10 seconds. There is no (or minimal) camera motion during each shot and the light does not change speed or direction. The first shots are framed to show the entire object or its main part, e.g., the face of a watch. A recurring choice is to use rim lighting on these first shots to show the object silhouette without revealing its appearance. Then, subsequent shots are progressively framed tighter on small details, and the last shot is often a well-lit view of the product in its entirety. In all the videos, to avoid distracting the viewer, the object is shown in front of simple uncluttered background, often black or white.

Shape and Material. Video artists often use the same strategies as photographers to emphasize the shape and material of the product. An exception is the use of a slowly moving light at a grazing angle to generate glittering on surfaces with specular micro-geometry. Besides glittering, several other effects can

be interpreted as an adaptation of well documented guidelines used for static photography.

Placing lights around the object to maximize the contrast around edges helps reveal the shape of the object [27]. Placing them behind produces rim lighting that shows off the silhouette and separates the object from the background [137]. Setting the light behind a glass object with black elements on the side creates a "bright field" that reveals the shape of the object that would be otherwise transparent [15].

For translucent objects, back and side lighting emphasize the translucency of their material by maximizing scattering [154]. Grazing illumination increases the visibility of the mesostructure of rough surfaces by generating fine-scale shadows [126]. For specular objects, an illumination that minimizes highlights while maximizing the diffuse reflection reveals their intrinsic color [27], while lighting them so that highlights align with the main curved regions helps understand their shape and emphasizes the material shininess [85].

4.3 Data Acquisition

We define our acquisition procedure with a few requirements in mind. Inexperienced users should be able to follow it. The equipment needed should be minimal and easily available. Later, we use the captured data to automatically create snippets that users can assemble. Thus, we recommend that the user works on the side of over-acquiring because it has a minimal cost to the user and maximizes the chances to get useful snippets. Finally, we guide the user to capture long smooth sweeps as recommended in our design principles (section 4.2).

From these guidelines, our acquisition proceeds as follows.

1. The capture session is done in the dark so that only our light source illuminates the product, e.g., at night with the room lights turned off.
2. The product is placed on a dark support to prevent undesirable interreflection, e.g., a black table or a table covered with black fabric.
3. The video camera is placed on a tripod so that the product lies in the center of the frame. As we shall see, a smartphone is sufficient.
4. While the camera is recording, the user waves a tablet displaying a fully white image around the product while following these guidelines to the extent possible:
   - the tablet should move along long smooth arcs approximately centered on the product,
   - its motion should be smooth and slow,
   - it should cover a broad range of trajectories: front to back, left to right, top to bottom, and so on, though it does not need to exhaustively sample the entire space,
   - it should not block the camera line of sight,
   - it should be close to the object in order to maximize the effective area of the light source, but the actual distance may vary along the capture session (it is normal for the tablet to be in the frame when it is behind the product).

This procedure satisfies our requirements: no specialized equipment is needed, a smartphone and a tablet are sufficient, and it follows our design principles about using an area light source along long smooth sweeps. In our experiments, acquisition sessions took about 10 minutes, of which 3 to 4 were

dedicated to recording. The result is a 3 to 4-minute long footage that we analyze and decompose into snippets in the next section.

We asked novice users, unrelated to the project, to capture data following those instructions and they found it relatively straightforward to work with (even including a 11 year old as shown in Figure 4.1). We discuss this in more detail later and provide examples in [23].

4.4 Analysis

Given the user's input footage (few minutes), we split it into snippets (about hundred or so), and then develop scoring functions that rank the snippets based on various criteria: color, shape, motion, rim lighting, glitter, vertical and horizontal motion.

The footage that we acquired in the previous section is too long and contains many uninteresting portions, e.g., changes of directions between two sweeps, inconsistent light motion, unappealing highlight location or shape. As is, it would be a poor way to show off the object, but it does have good raw material; some sub-sequences are good. Our goal is to find them to use them as building blocks for the final video. Our approach also eases the burden of the acquisition on users since it is tolerant to bad segments in the recording.

In this section, we first split the captured footage into shorter segments that we call *snippets*. Then we analyze these snippets to give a series of scores that allow us to rank them according to the criteria derived from our design principles (section 4.2). For this analysis, we assume that a mask of the object

is available; in practice, we created it by hand, which was easy since the object stands in front of an uncluttered background. Automated techniques like [31; 87], or semi-automatic techniques like [28; 125] can be explored in the future.

4.4.1 Splitting the Input Footage into Snippets

In this section, we decompose the captured video into short snippets. Following our design principles, we seek to isolate long portions where the light follows a smooth trajectory. Since we do not have access to the 3D position of the light during the recording, we rely on image cues to infer this information. Intuitively, a smooth light motion results in smooth variations of image properties like the optical flow and the pixel colors, and one can analyze these properties to learn about the light. However, image variations can also be triggered by other phenomena such as occlusions and geometric discontinuities. Because of that, analyzing a single cue is brittle. Instead, we rely on three of them that we describe next: the direction of the motion of the highlights, its amplitude, and the image colors. The rationale is that sharp variations in the light trajectory affect all three cues at the same time whereas, from our experiments, other causes perturb only one or two.

Direction smoothness score. We observe the highlights and estimate how fast the direction of their motion is changing at each frame. We use the [100] method to compute the flow at each pixel between each pair of adjacent frames. While Lucas-Kanade is not the best performing on standard optical flow benchmark, unlike other approaches, it assumes very little about the scene, which is critical to track highlights that typically violate the assumptions made by other techniques,

e.g., our objects are not Lambertian. We experimented with other state-of-the-art optical-flow algorithms and found that Lucas-Kanade gives more stable and predictable results on our challenging data sets. We further use the per-pixel confidence values, outputted by the algorithm, to concentrate our analysis on pixels where the algorithm behaves well.

First, we estimate the dominant motion direction of the highlights between frames i and $i+1$ by building a histogram of flow vector directions, but only for pixels with a confidence in the top 5%. Each sample is weighted by the magnitude of the optical flow and the intensity of its corresponding pixel, $\bar{I} = 0.299R + 0.587G + 0.114B$. This weight gives more importance to highlights, while reducing that of small flow vectors more likely to be due to noise. We define the dominant direction as the label of the fullest histogram bin H_1 and estimate a confidence factor $1 - |H_2|/|H_1|$ that favors cases where the selected bin is unambiguously larger than the second fullest bin H_2.

Next, we propose to look at the neighborhood around each frame to reason about the consistency of the motion direction for a short period of time. The intuition is that a single frame-by-frame estimation may be unreliable, due to factors like noise and outliers, whereas analyzing the motion for a short period of time would give us a more robust estimation. For frame i, we consider the dominant directions of the N previous frames, we use $N = 12$. We build a new histogram with them, this time using their confidence factor as weight and again extract the dominant direction that we call D_ℓ. We do the same with the N following frames to get D_r and compute the angle difference $D_{\ell\mathrm{r}} = [(D_\ell - D_\mathrm{r} + \pi) \mod 2\pi] - \pi$. The direction smoothness score is computed as: $S_\mathrm{d}(i) = \exp(-D_{\ell\mathrm{r}}^2/(\pi/6))$. We found the scale factor $\pi/6$ to work well in our experiments

although its exact value had a limited impact on the results. The same is true for the other factors used in the rest of the section.

Highlight speed smoothness score. We now estimate how smoothly the speed of the highlights varies. First, we compute their speed between frames i and $i+1$ as the average of the magnitudes of the flow vectors weighted by the intensity of their corresponding pixel. When computing this average, we discard the amplitudes smaller than 1 pixel because they are likely to be dominated by noise and such small motion is not perceivable. We then compute the median of the N previous and N following frames to get V_ℓ and V_r respectively, and compute the smoothness score $S_\mathrm{a}(i) = \exp((1 - \frac{\min(V_\ell, V_\mathrm{r})}{\max(V_\ell, V_\mathrm{r}) + \epsilon})^2 / 0.5)$ with $\epsilon = 10^{-7}$.

Light speed smoothness score. For the last cue, we seek to estimate how fast the light was moving when the video was recorded. Our approach is inspired by the work of Winnemöller et al. [153] who showed that image color differences relates to 3D light distances. Inspired by this result, we estimate the speed of the light source between frames i and $i+1$ as the sum of the absolute values of the temporal derivatives at each pixel. Then, similarly to the previous case, we compute the medians T_ℓ and T_r, and the smoothness score: $S_\mathrm{s}(i) = \exp((1 - \frac{\min(T_\ell, T_\mathrm{r})}{\max(T_\ell, T_\mathrm{r}) + \epsilon})^2 / 0.3)$.

Overall snippet smoothness score. Finally, we add all three scores to get $S_\mathrm{cut}(i) = S_\mathrm{d}(i) + S_\mathrm{a}(i) + S_\mathrm{s}(i)$. Low smoothness values are likely to correspond to irregular light motion, e.g., between two sweeps, and local minima are good candidates to cut if needed. Our cutting procedure processes the video recursively, starting with the entire recorded video. Given a sequence of frames $[a, b]$,

we first check that it is longer than the threshold L_{min} controlling the shortest sequence that we can cut. If it is longer, we compute $\sum_{i\in[a;b]} S_{\text{cut}}(i)/\min_{i\in[a;b]} S_{\text{cut}}(i)$ and compare it to the threshold L_{max} that defines the maximum length of a snippet. If it is above, we cut at the local minimum $\arg\min_{i\in[a;b]} S_{\text{cut}}(i)$. The advantage of this criterion is that it always cuts sequences longer than L_{max} and for shorter sequences, the shorter they are, the less likely they are cut because the sum in the numerator contains fewer terms. That is, our criterion balances the requirements of not having overly long snippets while at the same time avoiding too short snippets. All our results are generated with $L_{\text{min}} = 20$ and $L_{\text{max}} = 200$.

Discussion. We derived the above formula and parameters empirically during our early experiments. We used them for all our results (with the same parameters), and achieved good results. Other choices aimed at addressing the same high-level objectives may work equivalently well, and could be worth exploring.

4.4.2 Assigning Scores to the Snippets

The above approach generates about a hundred short snippets for a video of a few minutes. To make exploration of these snippets convenient, we assign scores to these snippets ranking them. We now explain how we assign each snippet a set of scores motivated by our design principles (section 4.2). We use these scores later in our user interface (section 4.5) to help users find the snippets that are most useful to follow the design principles.

We compute the scores by first estimating per-pixel quantities that we later sum over a region of interest M. Formally, to compute a score S on a snippet

$[a; b]$, we first define per-pixel values s at each pixel p and sum them over the region M: $S([a; b], M) = \sum_{p \in M} s([a; b], p)$. For brevity's sake, we omit the $[a; b]$ operand. This formulation enables users to create masks to search for snippets that achieve a desired effect on a specific part of the product. We now describe each scoring function in detail. But first we explain how to summarize a snippet with a single image that we call a *still* and that we use in the definition of several score functions.

We propose two sets of criteria based on: (1) analyzing the properties of the *still* snippets and (2) analyzing the observed, image-space motion of the snippets. In our first set of criteria we rank the *still* snippets based on how well they show (a) the underlying material colors, (b) the shape and texture or (c) the contours of the object. Our second set of criteria analyze the motion and rank the snippets based on how well they reveal (a) the typical motion in a region, (b) glittering of faceted geometry or (c) various sweeps of highlights (horizontal vs vertical).

Summarizing a Snippet with a Still. Summarizing a snippet with a single image that we call a *still* is a useful building block when defining the score functions. We also use that image in the production of our final result to create a "base image" representing the static illumination of the scene on top of which the highlights move. Some of our "base image" goals are similar to our basis lights from Chapter 3, but the new method proposed here makes them more appropriate for shiny and reflective objects. We seek an image I_{still} that shows all of the frames at once. A naive solution is to average all the frames but this generates a bland image in which the highlights have been averaged out. Another option is the per-pixel maximum but it is sensitive to noise. Instead, we

use the per-pixel per-channel soft-max over the snippet:

$$I_{\text{still}}(p) = \frac{\sum_{i=a}^{b} I_i(p) \exp(\alpha I_i(p))}{\sum_{i=a}^{b} \exp(\alpha I_i(p))} \tag{4.1}$$

where the computation is carried out independently for each color channel and α controls the effect: $\alpha = 0$ corresponds to standard averaging and larger values make the result closer to the actual maximum. We use $\alpha = 5$ for all the results in the paper.

(A) Color

This function seeks to assign high scores to snippets revealing the color of the object as opposed to the color of the light reflected on it. Since we use a white light source (a tablet displaying an all-white image), we use color saturation to differentiate the object color from that of highlights. This strategy is similar to that of our *diffuse color light* in Chapter 3, but the approach there, based on RGB angles, may favor dark pixels and requires a correcting factor. Instead, here we propose an alternative, simpler method, which we found to work better in practice. We use the RGB distance to the gray diagonal of the RGB cube. We compute this quantity over the still image of the snippet to define the per-pixel score function

$$s_{\text{color}}(p) = \sqrt{(R_{\text{still}} - \hat{I}_i)^2 + (G_{\text{still}} - \hat{I}_i)^2 + (B_{\text{still}} - \hat{I}_i)^2} \tag{4.2}$$

where $(R_{\text{still}}, G_{\text{still}}, B_{\text{still}})$ is the color of the pixel p in the still image of the snippet, i.e. I_{still} (p), and $\hat{I} = (R_{\text{still}} + G_{\text{still}} + B_{\text{still}})/3$ is its projection on the black–white axis. This measure does not favor dark pixels because these are all close to each other in the black corner of the RGB cube. In comparison, well-exposed pixels lie in the middle of the cube and can be farther away from the gray diagonal. We observed

(a) Average image (b) Median score (c) High score

Figure 4.2: *Color criterion*. This wine bottle features a translucent liquid inside a refractive bottle. Our color criterion, computed inside the mask in red, gives high scores to back-lit sweeps that reveal the color of the wine.

in our experiments that this metric is effective even with objects that look gray because in practice, they are never perfectly colorless, which allows our criterion to work.

(B) Shape and Texture

This score is about finding snippets that show off the shape and texture of the product well. The intuition behind our approach is that the structures that repeatedly appear in the captured footage are characteristic of the object (or its texture) while those that are only visible in a few frames are not. Our goal is to rank snippets that reveal these characteristic repeated features higher. We build our scoring function upon *structure tensors*. We first review their definition and their properties, and then explain how we use them in our context.

Background on Structure Tensors. Structure tensors are a tool to analyze 2D vector fields defined over images. In this section, we will use them on intensity gradients, which is their typical use, and later on optical flow vectors. Each pixel p has a 2D vector $\mathbf{u}(p)$ assigned to it. We first build the 2×2 matrices $\mathbf{u}(p)\mathbf{u}(p)^\mathsf{T}$ and average them locally to form the structure tensors $\mathbf{T}[\mathbf{u}] = G_\sigma \otimes \mathbf{u}\mathbf{u}^\mathsf{T}$ where G_σ is a 2D Gaussian kernel with standard deviation σ and $\otimes$ the convolution operator. Intuitively, the larger eigenvector $\mathbf{e}_1$ of T points in the "dominant direction" of the $\mathbf{u}$ vector field, the two eigenvalues λ_1 and λ_2 indicate the "strength" of the vector field along this direction and the orthogonal one, and comparing these two eigenvalues gives an estimate of the "orientation consistency" of $\mathbf{u}$, i.e., when all the vectors point in the same direction, $\lambda_1 \gg \lambda_2$, and when they point in different directions, $\lambda_1 \approx \lambda_2$. To aggregate the information from several tensors, one can simply add them but this is known to produce inaccurate estimates of the orientation consistency [8]. Instead, it is recommended to work in the log-Euclidean space:

$$\log(\mathbf{T}) = \log(\lambda_1)\mathbf{e}_1\mathbf{e}_1^\mathsf{T} + \log(\lambda_2)\mathbf{e}_2\mathbf{e}_2^\mathsf{T} \tag{4.3}$$

Adding two tensors in the log-Euclidean space amounts to $\exp(\log(\mathbf{T}_1) + \log(\mathbf{T}_2))$ where $\exp(\mathbf{T})$ is defined similarly to $\log(\mathbf{T})$. To compare two tensors, we use the *normalized tensor scalar product*:

$$\text{ntsp}(\mathbf{T}_1, \mathbf{T}_2) = \sum_{i=1}^{2} \sum_{j=1}^{2} \lambda_{1,i}\lambda_{2,j}(\mathbf{e}_{1,i} \cdot \mathbf{e}_{2,j})^2 / (\text{tr}(\mathbf{T_1})\,\text{tr}(\mathbf{T_2}) + \epsilon), \tag{4.4}$$

where $\epsilon = 10^{-7}$ as earlier, tr is the trace operator, i.e., the sum of the diagonal elements, and the subscripts indicate first whether the quantity is related to $\mathbf{T}_1$ or $\mathbf{T}_2$, and then the order among the eigenvectors or eigenvalues. We chose this comparison function because the normalization by the traces makes it insensitive to global scale factors, for example, due to comparing tensors that comes from summations over a different number of frames. Finally, one can estimate how well aligned a vector $\mathbf{v}_0$ is with the vectors represented by a structure tensor by computing $\mathbf{v}_0^\intercal \mathbf{T}[\mathbf{u}]\mathbf{v}_0$. That can be shown to be equal to $G_\sigma \otimes (\mathbf{u} \cdot \mathbf{v}_0)^2$ using the definition of $\mathbf{T}[\mathbf{u}]$. That is, $\mathbf{v}_0^\intercal \mathbf{T}[\mathbf{u}]\mathbf{v}_0$ has high values when $\mathbf{v}_0$ aligns well with the $\mathbf{u}$ vectors, i.e., it has dot products with them with large absolute values.

Estimating the Structure Similarity. To find whether a snippet shows off characteristic features of the subject being photographed , we use structure tensors computed with the intensity gradients $\nabla \bar{I}$. For a given snippet, we compare the log-Euclidean sum over the entire recorded video to the tensor computed over its still:

$$s_{\text{struct}}(p) = \text{ntsp}(\exp\Big(\sum_{\text{all } i} \log(\mathbf{T}[\nabla \bar{I}_i])\Big), \mathbf{T}[\nabla \bar{I}_{\text{still}}]) \tag{4.5}$$

where the sum is over all the recorded frames and the gradients are computed at p. The rationale for this scoring function is that, since the sum is over the entire recorded video that comprises thousands of frames, it captures only the features that appear in many frames, the other occasional features that are visible only

(a) Per-pixel anisotropy (b) Glass, median score (c) Glass, high score (d) Logo, median score (e) Logo, high score

Figure 4.3: *Shape & Texture criterion*. In (a), we visualize our per-pixel anisotropy score on the wine bottle (brighter is more anisotropic). Notice how high scores correspond to regions of high geometric curvature and high-frequency texture. The snippet with high score (c) captures the curved edges significantly better than the snippet with median score (b). Similarly, the logo is better represented by (e) with a high score than by the median (d).

in a few frames are negligible. Intuitively, it can be interpreted as a summary of the main structures visible in the video, and snippets with a similar structure tensor fields show off well these characteristic structures, which is the objective of this scoring function. Further, the intention is to use this criterion as a way of producing a "basis image" that reveals the underlying structure of the objects, and that is why we analyze the properties of the still.

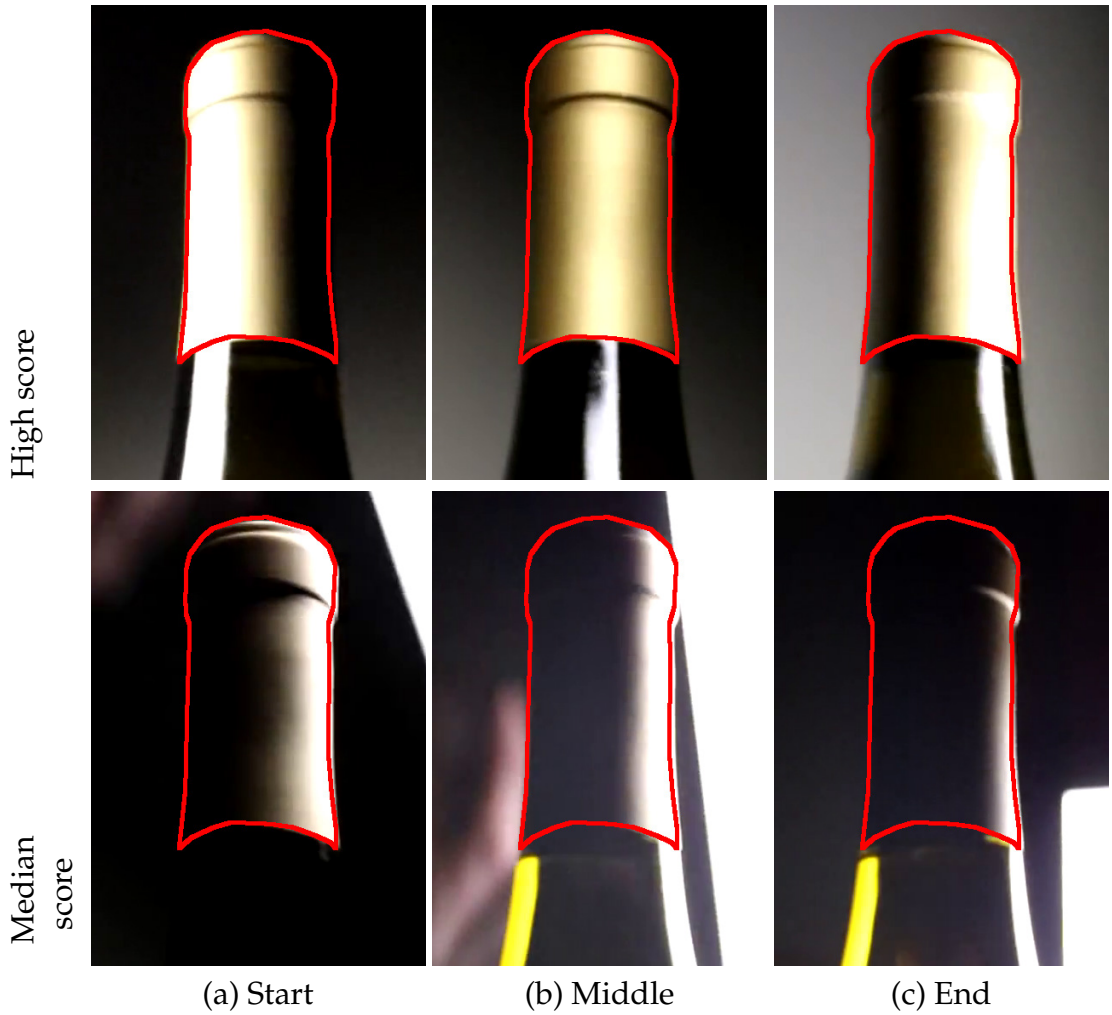
(C) Motion

We proceed similarly to score the snippets according to how well the motion visible in them represents the typical motion visible on the objects. We compute the structure tensor $\mathbf{T}[\mathbf{f}]$ of the optical flow $\mathbf{f}$ for each frame. We aggregate it over the entire recorded video and over the snippet only. The rationale is the same as in the previous case, aggregating over the entire video captures only the most characteristic features of the motion and we seek snippets with similar motion features. We define the scoring function:

$$s_{\mathrm{mo}}(p) = \mathrm{ntsp}\Big(\exp\Big(\sum_{\text{all } i} \log(\mathbf{T}[\mathbf{f}_i])\Big), \exp\Big(\sum_{i=a}^{b} \log(\mathbf{T}[\mathbf{f}_i])\Big)\Big) \tag{4.6}$$

(D) Contours

As we discussed in our design principles, emphasizing the object contours with rim highlights is a standard practice, e.g., [137]. Our scoring function is based on the observation that under rim lighting, the silhouettes of the product are bright and its center is dark, which approximately looks like the distance function to the object border encoded such that 0 is white and large values are black. We apply a distance transform [111] to the mask to get the distance to the border D which we remap to get $\tilde{D} = 1 - 2(D/\max(D))^{\nu}$ that equals 1 on the border and -1 at the center, with ν controlling how thick is the positive region near the border, i.e., how thick the rim highlight should be. Although $\nu = 1$ produces acceptable results, we found that it is better to set ν so that the positive and negative regions have approximately the same area. If we consider an object with a circular silhouette of unit radius, this means that the 0 level set of $\tilde{D}$ is the circle of radius $1/\sqrt{2}$ and since we measure the distance from the border, we have the



(a) Start (b) Middle (c) End

Figure 4.4: *Characteristic Motion*. The characteristic motion criterion on the cylindrical bottle cap. We show a few frames from each snippet (full snippets are availalbe in [23]). The highest-rank snippet (top) captures the characteristic motion along the curved cap much better than the median-rank snippet (bottom).

formula $(1 - 1/\sqrt{2})^{\nu} = 1/2$ that leads to $\nu = \log(2)/(\log(2) - \log(2 - \sqrt{2})) \approx 0.56$.

Then we remap the still's intensity so that bright pixels equal 1 and dark ones equal -1, that is: $\tilde{I} = 2\bar{I}_{\text{still}} - 1$. We use these two quantities to define our scoring function:

$$s_{\text{rim}}(p) = \tilde{D}(p)\tilde{I}(p) \tag{4.7}$$

(a) Inverse distance transform (b) Median score (c) High score

Figure 4.5: *Rim-light criterion*. The rim-light criterion on the wine bottle scene, computed using the inverse normalized distance transform (a). The rim-light effect in the high-ranked snippet (c) reveals the contours of the bottle whereas the median-ranked snippet (b) does not.

This score is high only for bright pixels near the silhouettes and dark pixels near the center, which corresponds to a rim illumination.

(E) Glittering

Gems and surfaces with fine micro-geometry glitter, and artists often emphasize this effect. We characterize glittering as the fast variation of the fine-scale details of the image. Formally, we first apply a high-pass filter to each frame's intensity to get a layer $H_i = \bar{I} \otimes (1 - G_\sigma)$ where G_σ is a 2D Gaussian kernel with standard deviation σ. We use $\sigma = 1$ in all our results to capture only the highest-frequency details. Then, we measure the amplitude of the temporal derivative of this layer to define our score:

$$s_{\text{gli}}(p) = |H_{i+1}(p) - H_i(p)| \tag{4.8}$$

(F) Horizontal and Vertical Motions

A critical artistic choice is the direction in which the highlights move. The two standard choices are horizontal and vertical sweeps. We provide a scoring function for each by estimating how well represented by the optical flow structure tensor the vectors $(1; 0)^\mathsf{T}$ and $(0; 1)^\mathsf{T}$ are:

$$s_{\text{hor}}(p) = \sum_{i=a}^{b} \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{T}[\mathbf{f}_i] \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{4.9a}$$

$$s_{\text{ver}}(p) = \sum_{i=a}^{b} \begin{pmatrix} 0 & 1 \end{pmatrix} \mathbf{T}[\mathbf{f}_i] \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{4.9b}$$

4.5 GUI and Compositing

As the last step in our workflow, the user composites and sequences the analyzed snippets into a final video using our GUI, we show a screenshot in [23]. At

startup, we present to the user the familiar preview pane similar to other photo and video editors. Since the video is often dark, we initially show the temporal average frame.

The two main user interactions are to *create a region*, and to *assign a snippet* to a region. Regions are similar to layers in numerous image editing tools in that they are associated with image data (i.e.,a snippet), and a blending mask. With the mouse, users can create rectangular or lasso-polygon regions, which we rasterize into an alpha mask with a Guided Filter [72], using the average image as the guide.

Our approach to dynamic lighting design is an exploratory endeavor and to that end, we use a *faceted search* approach to navigate the space of snippets, where the user is given a series of *cascaded lists*, similar to the interface found on many shopping websites. After selecting a region and a primary criterion (e.g.,rim lighting), an initial list contains all snippets sorted by the mean criteria score inside the region. The user either chooses one of these snippets, or selects a secondary criterion, in which case our system takes the top 50% of the snippets, sorts their mean scores in the secondary criterion, and presents them in a secondary list. Cascading continues until the user selects a snippet. To help the user quickly understand the selected snippet, we provide a summary panel with a temporal soft-max of the snippet, a tone-mapped score image, and the blending mask. The summary panel also has a button to play the snippet in the main preview window (compositing it over the other regions), and a slider to change playback speed.

Compositing. For novice users, traditional alpha compositing can lead to unexpected results when ordered improperly. Therefore, we use the soft-max operator, which is commutative, for spatial compositing as well. For our application, soft-max significantly outperforms other commutative blending operators such as *plus* because our focus is on bright moving lights, which tend to blur under linear combinations. Given a collection of regions, their alpha masks, and corresponding snippets, we premultiply each snippet frame by the alpha mask, apply soft-max across regions, and normalize by alpha.

Sequencing. In lieu of a timeline, we provide a simple but flexible scripting interface to sequence the final video. Users simply create a series of shots, which are played sequentially with fade to black between. Each shot is a collection of regions, which are either dynamic (at its selected framerate) or a still (its temporal soft-max), an optional zoom window, and an optional background. All of our results are created using this system. In [23] we show a screen capture of an interactive session.

In our user study in section 4.6.1, the more sophisticated users asked for advanced blending modes and nonlinear editing. These features are complementary to our prototype and can be added to make the GUI production quality.

4.6 Results

We demonstrate the effectiveness of our approach to produce compelling lighting-designs on a variety of objects for both type of digital media: still images and short videos. In [23] we used Adobe Premiere to create a fade-in and fade-out

effect between the individual short clips generated by our system, and to add music. Those features are orthogonal to our research and they can easily be added to future versions of our prototype software.

Wine bottle (still). The wine bottle in Figure 4.8 is a common case in product photography. We demonstrate that our technique allows a quick exploration and combination of classical lighting-design objectives, which are captured by the top few high-ranked snippets sorted according to our criteria. In [23] we also show a video result on this sequence.

Sunglasses (still). In Figure 4.9 we show a still lighting-design result on a pair of sunglasses. The highly reflective glass is a particularly challenging case for photography, since reflections from the environment lighting can produce distracting patterns. In Figure 4.9b we show the *edge light* result from our previous method, where the goal is to find a linear combination of the input images such that consistent edges in the scene are emphasized. In Chapter 3 we model the problem as a continuous, non-linear ℓ_2 optimization over the per-image weighting factors. This leads to a smooth redistribution of the error across the full scene which, for highly reflective objects, results in an unpleasant mixture of highlights that cross-fade with different intensities. In Figure 4.9c we show one of the high-ranked (top 2) *still* snippets according to our new shape revealing criterion. The continuity property of our snippets allows us to produce an aesthetically more pleasing result that achieves a good balance between revealing the underlying structure and producing consistent and smooth reflections.

Golden watch (video). Watches are often the subject in the lighting-design videos we studied. In Figure 4.10 we show a few frames of a short, 22 seconds video clip produced with our system. We start by showing the full scene and we play one of the high-ranked snippets that emphasizes the overall shape through rim-lighting. Next, we zoom-in on a few regions and play snippets that reveal various shape and material properties. We zoom into the case and use the glittering criterion to emphasize the diamonds, composited over a *still* snippet that reveals the texture of the glass. Finally, we zoom-out to show a full view of the scene where we *still* a few snippets to get a good base light on top of which we play the highest-ranked snippet that captures a horizontal highlight sweep. This final sweep is often seen in professional videos, where the goal is to attract viewer's attention to the reflective behavior of the glass case. Our *horizontal motions* criterion is designed to capture this common goal.

Shower gel (video). In Figure 4.11 we show a common, everyday object (a bottle), that is often seen in product ads. We show that our criteria are expressive enough to capture common objectives that lighting-designers try to achieve, e.g., color-revealing light on the body, shape-revealing light on the cap, texture-revealing light on the logo, and long and smooth highlights to emphasize the reflective behavior and the shape of the plastic bottle. Our system allows the production of videos that can be used to further emphasize and attract users' attention to these material and shape related properties through proper light-sweeps.

Perfume (video). Perfume bottles made from faceted glass are a challenging case, since they have complex interactions with the moving light. In Figure 4.12

we show a video result on a perfume bottle using our method. To reveal contours, we first play high-ranked rim-light snippets to the left and then right sides of the bottle, which reveals its overall shape. Next, we zoom in on a few regions to emphasize shape (the cap) and material (the logo). Finally, we zoom out and show the *still* image of the highest-ranked snippet that reveals the shape of the body, on top of which we blend an animation of the snippet that highlights the logo.

Lens (video). Camera lenses (Fig. 4.13) combine two materials that are challenging for the lighting designer: the black plastic body is only visible near strong highlights, and the glass elements are both reflective and refractive. Despite these challenges, we can achieve pleasing lighting by compositing a small number of highly-ranked snippets. We choose *rim lighting* to find a snippet that sweeps to depict the overall body shape, then zoom right and select the *color* criteria to emphasize the red ring (an important product quality differentiator). We end by using the earlier snippets as a still base, and adding a *vertical sweep* across the glass to reveal its unusual colored reflection.

4.6.1 Validation

We conducted two small-scale user studies to validate our system.

Study 1: Our pipeline. The first study was designed to ascertain whether novice users, given only a short tutorial, can use our system to produce good product photography. We asked two novice users who have never seen our

system to go through our entire pipeline. They were both tasked with acquiring, analyzing (using our automatic techniques from section 4.4), and editing two objects: one chosen by us (*coins*), and the other chosen by them (*tool* and *camera*, respectively). Neither user had much video editing experience, although both have a fair amount of experience with image editing using Photoshop.

The first question both users asked even after our tutorial was "what do I do?" Despite the fact that they chose and acquired a personal item, they were unsure what is good lighting design for product videos. For this study, we did not show the users any professional videos and simply encouraged them to explore the dataset. User A was photographically motivated and spent a significant amount of time creating masks to blend various still lights, before adding a single moving light. User B was more exploratory and browsed around until he found a "favorite". He only applied one light sweep per region before quickly moving on to the next.

The results of the first study demonstrate that amateurs can definitely use our pipeline, although quality does still depend on "knowing what you want." Novice users still need to be trained in lighting design principles before they can take full advantage of our system. We show their results in [23].

Study 2: Snippet analysis and GUI. The goal of the second study was to assess the quality of our snippet analysis and its usefulness as part of a video production tool. We asked five users to work on the same sequence (*golden watch*. Each user was given a single viewing of an actual 40-second watch commercial as a template, a short tutorial on our system, and unlimited time to practice on a training sequence of a leather watch. They were then given 15 minutes to create

a video in the spirit of the template, and given a short exit survey (see [23]).

We found that overall, everyone liked the concept. One user wrote: "I like that it is giving me a tool to emulate professional product shots without having to buy a bunch of gear. Lighting is a big separator between professional and non-professional photographers/videographers." Most users found our snippet analysis "generally helpful, although not completely reliable.", and that "The classifiers help, but its still a pretty big list to sift through." Although half the users mentioned that they found the fully-automatic compositing intuitive, everyone felt that the biggest pain point was the lack of a fully-featured nonlinear editor. Users all wanted the ability to trim, or reverse snippets, and the more advanced users wanted the ability to apply more sophisticated blending.

To summarize, all the users liked the concept of a one-person DIY tool to create professional-looking product videos. They found faceted search of cataloged snippets to be a fast way to find the right effect. However, with regards to the user interface, nearly all users wanted additional features and would have preferred that our tool be part of a nonlinear video editor such as Premiere or iMovie. This request is orthogonal to our research, but is important to judge practical utility.

4.6.2 Discussion and Limitations

We describe a user-driven approach meant to help users create compelling lighting-design videography. However, not all criteria are necessarily useful in all situations, e.g., if a scene does not have materials with glittering properties, our criterion can not return a snippet that has the expected behavior. That said,

our experiments show, that even if one (or two) criteria do not produce the expected behavior on a given scene, many of the others would.

In general, our results do not correspond to a physical setup, since our per-region blending does not alter the illumination in a physical way. However, it is close to what could be produced using blockers, and our results look plausible. Further, our soft-max operator, which blends frames across snippets in a non-linear manner, is close to what one can get in practice with a longer exposure, and it also looks plausible.

Our snippets extraction procedure may not always cut at the right place where a designer would desire, i.e., our assumed smoothing scores may not always correspond to what a designer would perceive as a smooth and complete sweep. Nevertheless, we found that our automatically extracted snippets and the proceeding ranking that we do on top of that, are useful to quickly "send" the designer to a desired place of the video where the effects of various criteria are observed. Sliders to refine the start/end frames of the snippets can easily be provided to let a designer further refine the snippets.

Finally, our current prototype interface lacks advanced video editing tools, which was also the remark of some users during the study. However those features are beyond the scope of our paper. Some useful additional features would be to let the user select the length of the snippet if our snippets are not quite what they desire.

Discussion of 2D vs. 3D Our current approach is entirely image-based, which is a deliberate decision that we made in order to keep the requirements of the acquisition stage accessible to novice users. However, some knowledge of 3D

information, such as the positions of the light sources, could potentially be helpful to improve our reasoning about the smoothness of the light paths. For example, Isomap embedding, similar to [152], can be used to bridge the gap between our 2D capturing system and retrieving information about the 3D configuration of the input lights. In Figure 4.14 we show an example of one such embedding, computed on the golden watch data set. In this experiment, we use Isomap embedding [139] based on the RGB Euclidean distance between images. Similar to [152] we set the Isomap algorithm to produce 3-dimensional embedding which we further project onto a sphere, since this is our prior believe about the space of the input light paths. This could potentially be used to estimate the 3D positions of the input lights and reason about the light paths and configurations from that perspective. It could also be used for other applications such as image-based relighting, [42].

4.7 Conclusion

The growth of sites like craigslist and eBay is increasing the need for tools that enable easy-to-produce product photographs and videos. Additionally, the increasing ubiquity of electronic billboards and online advertising is increasing the importance of product videography.

We introduce a do-it-yourself lighting design system for product photography and videography. Our pipeline of acquisition, analysis, compositing lets novice users produce high quality lighting design for products without too much effort. Our simple acquisition pipeline requires no specialized hardware beyond a tablet and a smartphone. We automatically analyze videos to produce snippets that are

ranked on various criteria based on whether they reveal shape, texture, motion, glitter, or achieve rim lighting.

Many future avenues of research remain. A more complete production quality UI would improve the user experience. Automatic summarization of the input video could decrease user interaction, except when they want to artistically control the results. Combining this approach with an Arqspin [7] type product can let us expand the range of achievable effects by combining varying viewpoints and illumination of an object. Exploiting knowledge of the 6DoF tracking of the light and camera to get 3D information could also significantly expand the possibilities. Further development of the user interface could improve the usability of our system. For example, introducing brush-like tools where users are allowed to "draw" constraints on various parts of the objects, e.g., highlights or colors revealing brush, could make the exploration more intuitive to some users. Evaluation with novice users could be used to determine the complexity of the interface that is needed to allow them to explore good results. For example, novice users can be given an example of a professionally lit photograph, and they can be asked to use different versions of our interface to achieve a similar look. The results of this study could potentially give us insights on improving our interface.

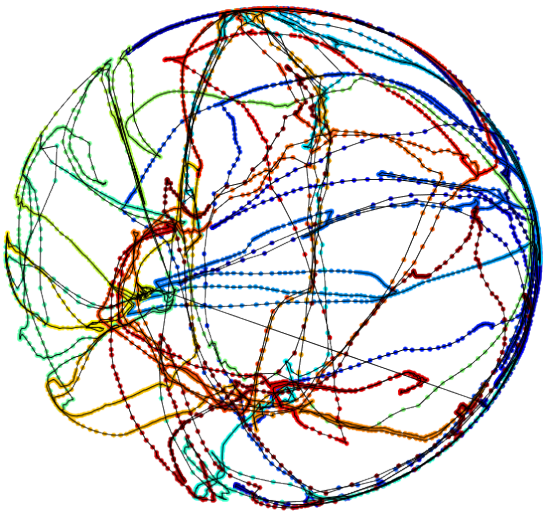
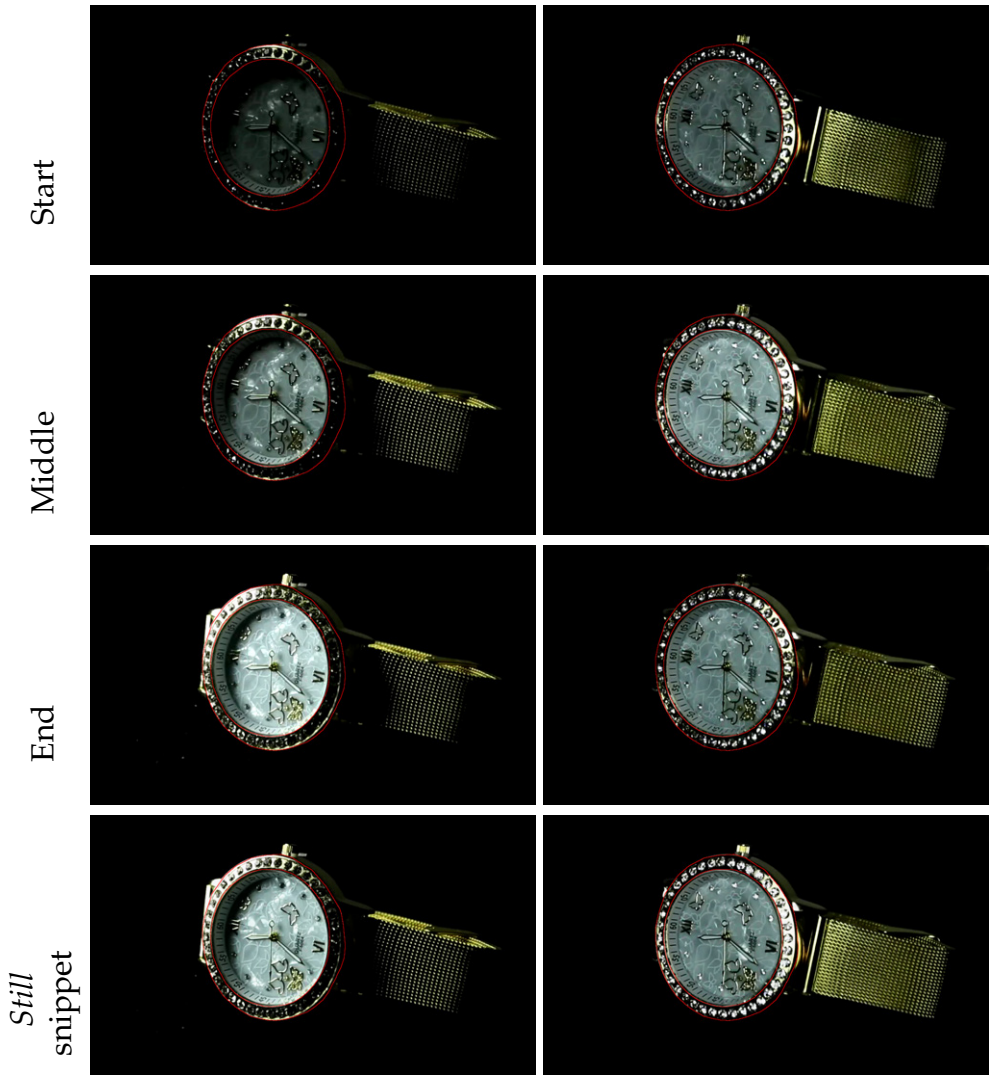
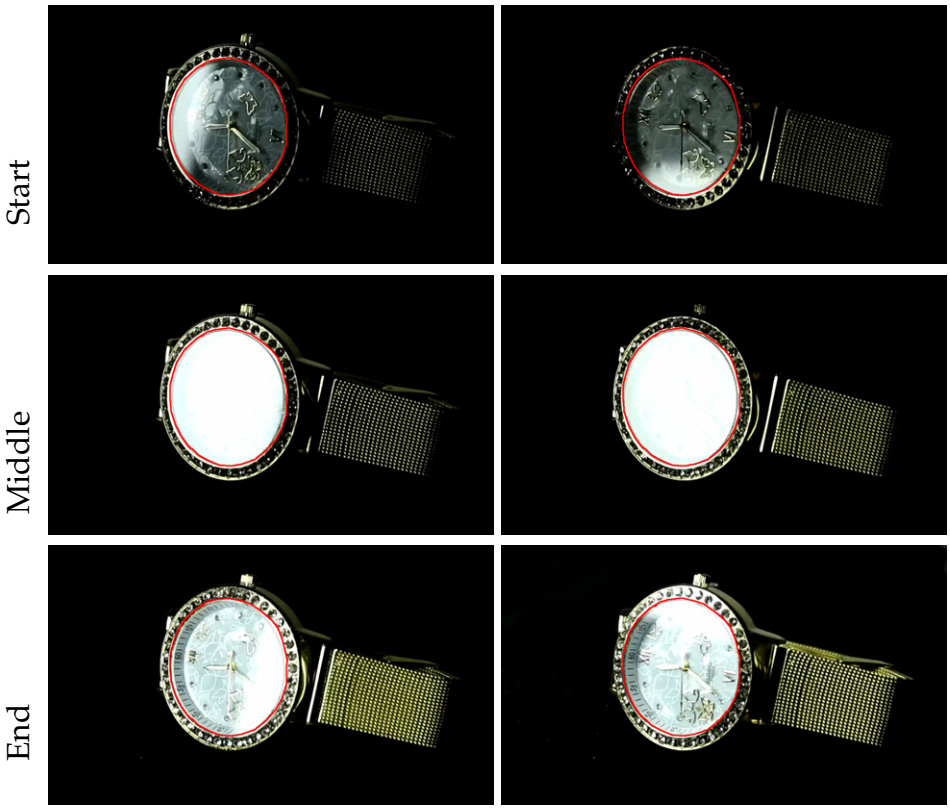


Figure 4.14: *Visualization of isomap 3D embedding*. We show a 3D isomap embedding, similar to [152], computed on the golden watch data set. Each point represents an input frame and the colors encode the time of each frame in the video, where blue corresponds to the beginning and red to the end of the video.



(a) Median score (b) High score

Figure 4.6: *Glittering criterion*. The glittering criterion on the studded watch bezel. We show a few frames from both snippets along with a *still* computed with soft-max. The restricted side motion captured by the median-ranked snippet (a) does not produce discernible glittering. The highest-ranked snippet (b) captures a variety of light directions from the moving light to produce a compelling glitter effect. Please see [23] for better comparison.



(a) High score horizontal (b) High score vertical

Figure 4.7: *Highlight sweeps criteria*. The highlight sweeps criteria on the glass watch face. Again, we show a few frames from both snippets. The snippet in column (a) captures a horizontal light sweep across the reflective glass. Likewise, the snippet in column (b) captures a vertical sweep.

(a) Professionally lit bottle, from Internet

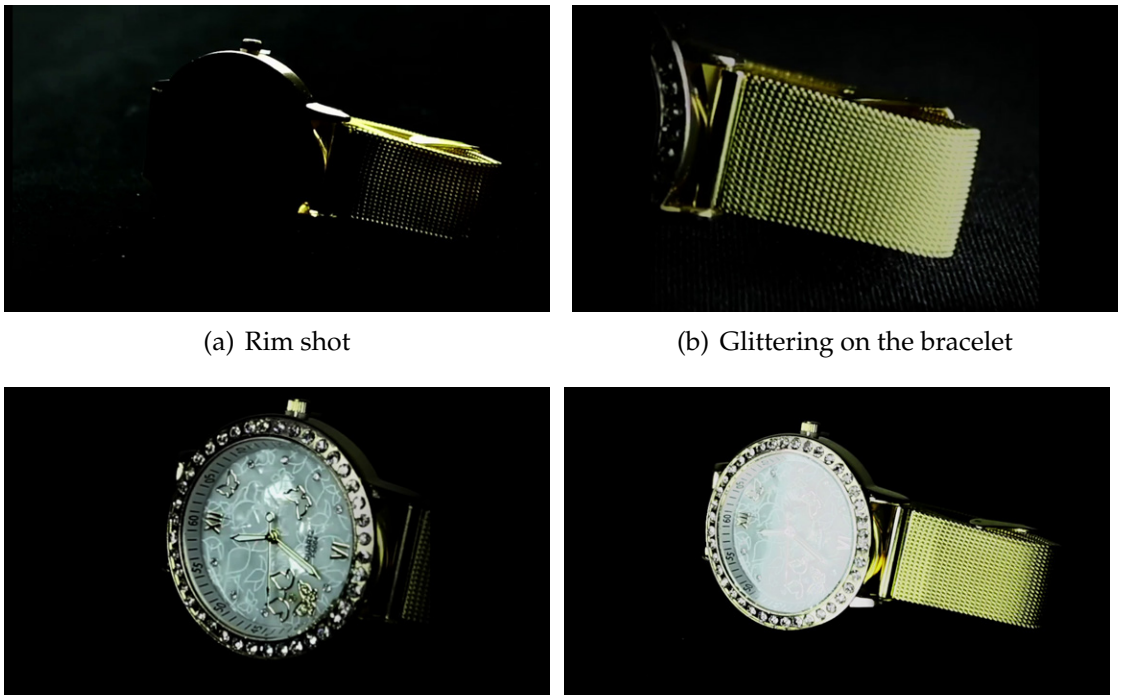
(b) Our criteria applied on different parts

(c) Our combined result, on a white background

Figure 4.8: *Combined still result on the white wine scene*. In the professionally lit photo (a), (1) the *back-lit* body reveals the colors and darkens the contours, (2) the *side highlight* reveals the reflective glass body, and (3) the *shape and texture* on the cap and the logo are emphasized through directional lighting. We achieve these effects (b) by applying the *color* and *vertical motion* criteria to the body, to capture the translucent color, dark contours, and vertical highlight ((1) and (2)), and (3) the *shape & texture* criterion to the cap and logo. Finally, we composite all these effects over a white background (c).

(a) Average image (b) *Edge light* by [27] (c) Our high-ranked shape revealing *still*

Figure 4.9: *Still result on the sunglasses scene. Comparison with our previous work.* We compare our *shape & texture* criterion to the *edge light* criterion of Chapter 3. Our previous system is designed to work with hundreds of unordered images whereas we have a video with thousands of frames. To adapt our new data for our old system, we sub-sampled a few hundred frames and ran our previous system on them. Although our *edge light* result nicely emphasizes the main edges such as the frames, it produces distracting cross-faded reflections on the glass. In contrast, our new *shape & texture* criterion finds a snippet that corresponds to a long, smooth, left-to-right light sweep, which produces aesthetically pleasing reflections while revealing the main edges (*still* shown in (c)).



(a) Rim shot (b) Glittering on the bracelet

(c) Glittering on the diamonds (d) Highlight sweep

Figure 4.10: *Video result on the golden watch scene*. A few representative frames of the *Golden watch* video produced using our system.

(a) Professionally lit example from Internet

(b) Average image

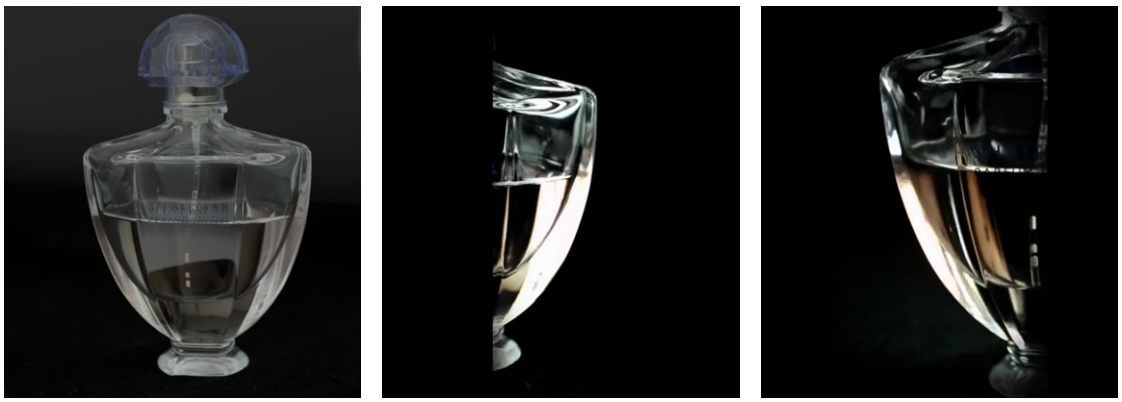
(c) Vertical sweep #1

(d) Vertical sweep #2

(e) Shape on the cap

(f) Our final shot: good base + horizontal sweep on the bottom label

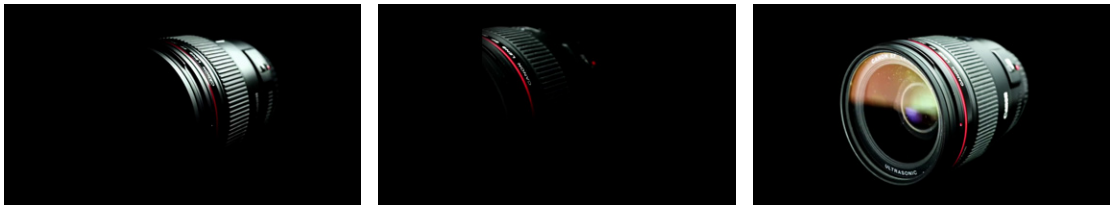
Figure 4.11: *Video result on the shower gel scene*. (a) a professionally lit shower gel, downloaded from Internet. (b) average image of our shower gel data set. (c)-(f) Representative frames of one possible video result generated with our method. All snippets are played on top of a high-ranked *still* snippet that reveals best the colors of the object. First, we zoom-in on the left (c) and then the right (d) part of the object, where we play the top two high-ranked snippets that reveal the vertical motion of highlights. Then, we zoom-in on the cap, where we play a high-ranked shape-revealing snippet (d). Finally, we show a full view of the object with the previously animated snippets blended, as *stills*, on top of the base *still* that reveals colors. Additionally we play a high-ranked snippet that captures a horizontal highlight sweep on the bottom part of the logo (e). Our final still shot captures many of the features in the professionally lit example in (a), e.g., (1) well emphasized colors, and (2) reflective behavior of the plastic body is revealed in an aesthetically good way by placing long and smooth highlights on both sides. See [23] for the final clip.



(a) Average image (b) Rim shot right (c) Rim shot left

(d) Shape on the cap (e) Colors on the logo (f) Final shot

Figure 4.12: *Video result on the perfume scene*. Representative frames from shots in our *Perfume* video.



(a) Rim shot (b) Color on the body (c) Highlight sweep

Figure 4.13: *Video result on the lens scene*. Representative frames from the shots in the *Lens* video. Note how the composite (c) simultaneously captures the shiny plastic body, the bright red ring, and the unusual colored reflection on the glass.

CHAPTER 5

USER-GUIDED WHITE BALANCE FOR MIXED LIGHTING CONDITIONS

In this chapter, we describe our second direction for assisting advanced photographic tasks: white balance correction under mixed lighting conditions. White balance is the process of removing unrealistic color casts, so that materials appear as if the color of all lights in the scene was neutral. This is a necessary step to produce realistic looking material colors in photography [76]. Since lighting design requires a mixture of multiple lights, especially for the case of real estate photography where indoor and outdoor lighting can be mixed in a complicated way, we develop better technique for the common white balance correction problem, which becomes much more challenging in those settings. We propose a user-guided solution to the ill-posed multi lights white balance problem based on simple user interactions, related only to relative reflectance properties of materials in the scene. We demonstrate the ability of our method to produce satisfying white balance corrections for hard scenes, without any assumptions of the color or the number of the light sources. This work originally appeared at ACM SIGGRAPH Asia 2012 [26]

5.1 Introduction

White balance correction is a critical photography step, where the goal is "to compensate for different color temperatures of scene illuminants" [78]. For example, tungsten lights cause images to have a yellowish cast. Proper white balance compensates for this color cast and yields photos where objects have their natural colors, as if taken under a neutral light [73]. When all the lights

have the same color, this problem is easy to solve for a photographer who often indicates a white or gray object in the image, from which it is straightforward to recover the illuminant color. Unfortunately, many scenes exhibit a combination of illuminants such as artificially-lit indoor scenes with additional light from a window (Fig. 5.1) or from a flash. Adjusting the white balance is then a challenging task, even for skilled users. Each point can be lit by the mixture of several light sources, depending on their relative distances and orientations. Worse, modern low-consumption fluorescent and LED lights vary widely in their color temperature, and rooms with multiple bulbs exhibit a plethora of color casts (Fig. 5.1).

A few automatic techniques have been proposed, but the severely ill-posed nature of the problem restricts them to specific scenarios. For instance, Ebner [49; 50] produces perceptual renderings that are often not ideal from a photography perspective. Hsu et al. [75] can handle only two light colors, need to know their exact values a priori, and cannot treat scenes with "a strong foreground-background separation." Riess et al. [124] assume that photos can be decomposed into regions where a single illuminant dominates.

We introduce a user-guided approach to produce high-quality white balanced images for a broad range of photos with multiple light sources. We carefully designed a set of scribbles that are easy for humans to specify, such as, neutral-color objects, regions of constant color, and places where the color looks correct. From our experience, it is difficult for a human to estimate quantities related to illumination. Therefore, as a general guideline, our scribbles are related to reflectance properties, rather than illumination. For instance, the local light color on a textured material, such as fur or fabric, varies in nontrivial ways

that cannot be easily understood by a human observer and described with scribbles. In comparison, the same observer can easily recognize and indicate regions where the fur has the same color. Similarly, human observers have no problem recognizing that a wall lit by a complex mixture of lights has a constant color. These situations are common in photographs and easy to identify. Our surroundings are full of constant-color objects such as walls and man-made objects – and our method is robust enough to also leverage textured materials, such as fur and fabric.

We formulate our method as an optimization problem that seeks to retrieve the color of the light mixture at each pixel under the constraints provided by the scribbles. We show that for canonical light-reflectance configurations, the white balance solution lies in the null space of the Matting Laplacian [91] , which motivates our use of this energy for regularization. We further identify points with similar visual look in the scene and constrain them to have the same reflectance. This scribble extension strategy allows users to achieve satisfying results with only a small number of scribbles. We demonstrate that our approach yields good results on a wide range of scenes; it can handle two or more light sources; and it does not require knowledge of the absolute values of lights or reflectances, since such knowledge is typically not easy to obtain. In practice, we show that it also copes with light mixtures and materials, beyond the theoretically studied base cases.

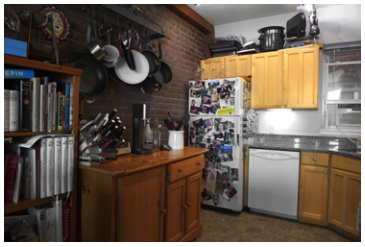
(a) Input with mixed lighting (daylight + neon under the cabinets + low-energy bulbs on ceiling) exhibits unsightly color casts everywhere

(b) Naive single-light white balance makes the ceiling white, but other color casts remain

(c) User indicates regions that are neutral (white strokes) or correct after the single-light white balance (grey strokes)

(d) The image is improved, but color variations can still be observed, e.g., on the wooden cabinet

(e) User adds marks to specify uniform color, e.g., the cabinet and the wall



(f) Our final output with no color casts

Figure 5.1: *White balance workflow*. In this photo, the ambient lighting, the cabinet light, and the ceiling lights all have different colors, which produces unpleasant color casts (a). In such situations, the single-light white balance tool provided in all photo editing software only improves a portion of the image, but the result is not satisfying (b). We address this issue by letting users make annotations on the photo. First, they mark objects of neutral color (i.e., white or gray), and regions that look fine after the standard white balance (c). This improves the result, but undesirable color variations are still visible, e.g., on the cabinetry and on the wall (d). Users can indicate that these elements should have a constant color (e), which yields a result free of color cast (f).

Table 5.1: Table of Notation

C_c^I	$\triangleq$	Input chromaticity, I_c / $(I_\mathrm{r} + I_\mathrm{g} + I_\mathrm{b})$	
C_c^O	$\triangleq$	Output chromaticity, O_c / $(O_\mathrm{r} + O_\mathrm{g} + O_\mathrm{b})$	$c \in \{\mathrm{r}, \mathrm{g}, \mathrm{b}\}$
C_c^R	$\triangleq$	Reflectance chromaticity, R_c / $(R_\mathrm{r} + R_\mathrm{g} + R_\mathrm{b})$	
W_c	$\triangleq$	Correction factors, I_c / O_c	

5.2 Mixed lighting white balance

Our approach starts with the standard workflow used for single-light white balance, but then introduces a set of scribbles to address the much more challenging problem of mixed white balance. First, the user globally adjusts the image to get an approximate result. This step is the same as the standard single-light white balance, and can be achieved with existing tools such as clicking on a white patch or adjusting the global color temperature. Then, we provide three brushes to annotate the image. *Neutral color* scribbles indicate objects that are white or gray. *Same color* scribbles show regions of constant reflectance or texture, but where the actual reflectance need not be specified. Finally, *Correct color* scribbles indicate areas that look correct in the current view. We designed these scribbles such that they are related to reflectance only, and do not require users to specify absolute values. We avoided scribbles related to lighting, because, in our experience, precise illumination properties are elusive for humans, especially on complex materials. After the user has marked the image with these scribbles, we solve a linear optimization problem that estimates the spatially varying RGB gain factors that explain the formation of the observed image with mixed lighting. We render the white-balanced image by applying the inverses of these factors. If needed,

users can iterate and add more scribbles to refine the result. Figure 5.1 illustrates this process.

We first present our image formation model. We explain how we formulate the white-balance problem as a least-squares optimization based on the Matting Laplacian, and motivate this approach by analyzing the null space of this energy. Finally, we express scribbles as constraints in the optimization and show how we can extend these scribbles to find additional constraints.

5.2.1 Image formation model and problem statement

The input of the algorithm is an image $(I_\mathrm{r}, I_\mathrm{g}, I_\mathrm{b})$ that represents a scene with reflectance $(R_\mathrm{r}, R_\mathrm{g}, R_\mathrm{b})$ lit by n_ℓ lights $\{(L_{i\mathrm{r}}, L_{i\mathrm{g}}, L_{i\mathrm{b}})\}$. We name λ_i the attenuation of the i-th light due to factors such as light travel and foreshortening. Further, we assume Lambertian materials and no inter-reflections. Using this notation, we model the observed image at a pixel p by:

$$\forall c \in \{\mathrm{r}, \mathrm{g}, \mathrm{b}\}, \quad I_c(p) = R_c(p)\left(\sum_{i=1}^{n_\ell} \lambda_i(p) L_{ic}\right) \tag{5.1}$$

In this equation, we only observe $(I_\mathrm{r}, I_\mathrm{g}, I_\mathrm{b})$, everything else is unknown. Our objective is to produce a white-balanced image $(O_\mathrm{r}, O_\mathrm{g}, O_\mathrm{b})$. Intuitively, we aim for rendering the scene as if each light had a neutral color, i.e., $(L_{i\mathrm{r}}, L_{i\mathrm{g}}, L_{i\mathrm{b}}) = (\ell_i, \ell_i, \ell_i)$ for some positive scalar ℓ_i. That is, we seek:

$$\forall c \in \{\mathrm{r}, \mathrm{g}, \mathrm{b}\}, \quad O_c(p) = R_c(p)\left(\sum_{i=1}^{n_\ell} \lambda_i(p) \ell_i\right) \tag{5.2}$$

However, this problem is severely under-constrained since we do not know the number of lights n_ℓ, their colors L_i, the corresponding spatially varying attenuation factors λ_i, and the spatially-varying scene reflectances R. Further, it

is unclear what the ℓ_i values should be. In practice, this problem is intractable without additional hypotheses.

In this work, in addition to the "neutral light color" goal (Eq. 5.2), we also seek to preserve image intensities, i.e., we also aim for:

$$O_\mathrm{r} + O_\mathrm{g} + O_\mathrm{b} = I_\mathrm{r} + I_\mathrm{g} + I_\mathrm{b} \tag{5.3}$$

For the sake of clarity, we use a simple model of intensity represented by the sum of RGB channels. Optionally we could weight each channel according to its perceptual importance, which would not affect the rest of our formulation. Intensity preservation, as defined in Equation 5.3, has an intuitive interpretation in the photography context. Our approach alters only the chromaticity values. Everything related to intensities remains unchanged, e.g., our operator is orthogonal to tonal adjustments such as brightness and contrast. For the rest of the chapter, we define the chromaticity of a pixel as its RGB channels divided by its intensity, that is, $C^I_c = I_c \;/\; (I_\mathrm{r} + I_\mathrm{g} + I_\mathrm{b})$ for $c \in \{\mathrm{r}, \mathrm{g}, \mathrm{b}\}$.

Our strategy to produce the white-balanced image $(O_\mathrm{r}, O_\mathrm{g}, O_\mathrm{b})$ is not to estimate all the unknown quantities in Equations 5.1 and 5.2. Instead, we seek $(W_\mathrm{r}, W_\mathrm{g}, W_\mathrm{b})$ factors such that at a pixel p:

$$\forall c \in \{\mathrm{r}, \mathrm{g}, \mathrm{b}\}, \quad I_c(p) = W_c(p)\; O_c(p) \tag{5.4}$$

This formulation drastically reduces the number of unknowns. Moreover, we show in the next section that in a number of cases, the W factors can be expressed as an affine combination of the observed chromaticity values $(C^I_\mathrm{r}, C^I_\mathrm{g}, C^I_\mathrm{b})$, which is the key of our approach based on the Matting Laplacian.

As we shall see later, it is useful to express W as a function of C^I. First, we use Equation 5.2 to get: $\sum_c O_c = (\sum_c R_c)(\sum_i \lambda_i \ell_i)$. Dividing Equation 5.2 by this result,

we obtain

$$\forall c \in \{\mathrm{r}, \mathrm{g}, \mathrm{b}\}, \quad C_c^O = C_c^R \tag{5.5}$$

since the light term $\sum_i \lambda_i \ell_i$ cancels out. Then dividing Equation 5.4 by Equation 5.3, we get

$$\frac{I_c}{I_\mathrm{r} + I_\mathrm{g} + I_\mathrm{b}} = \frac{W_c \, O_c}{O_\mathrm{r} + O_\mathrm{g} + O_\mathrm{b}} \tag{5.6}$$

$$\frac{I_c}{I_\mathrm{r} + I_\mathrm{g} + I_\mathrm{b}} = W_c \, \frac{O_c}{O_\mathrm{r} + O_\mathrm{g} + O_\mathrm{b}} \tag{5.7}$$

$$C_c^I = W_c \, C_c^O \text{ by definition of } C_c^I \text{ and } C_c^O \tag{5.8}$$

$$\tag{5.9}$$

Finally, using 5.5, i.e. $C_c^O = C_c^R$, we obtain the following equation which will be useful in the next section:

$$C_c^I = W_c \, C_c^O = W_c \, C_c^R \tag{5.10}$$

5.2.2 Standard scenarios and the Matting Laplacian

We study a number of standard cases under our model. We show that in all these cases, the W factors are an affine combination of the input chromaticities. We then use this result to adapt the Matting Laplacian introduced by Levin et al. [91] in the context of image matting to the problem of white balance under mixed lighting.

Case studies

We now study a few standard cases in increasing order of complexity. We start with the single-light and single-reflectance scenarios, and then discuss the more

complex case with two reflectance values lit by a 2D illumination.

Single-color illumination. Using the von Kries hypothesis [39], the effect of a single-color illumination can be modeled by globally scaling the RGB channels. That is, the W factors are constant over the image, which can be seen as a special case of an affine combination with zero coefficients affecting the chromaticity channels.

Monochromatic scenes. In the case where the reflectance R is constant over the scene, Equation 5.10 gives $W_c = C_c^I / C_c^R$. Since C^R is constant, it means that W is proportional to C^I, which is a special case of affine combination.

Duochromatic scenes under 2D illumination. In the appendix, we show that under some reasonable assumptions, the previous result extends to scenes with two reflectances lit by an illumination that lies on 2D subspaces of the RGB cube. In this case, W can be expressed as an affine combination of $(C_\mathrm{r}^I, C_\mathrm{g}^I, C_\mathrm{b}^I)$ in which all the coefficients are nonzero. This case illustrates that it is beneficial to consider all the channels at the same time and allow cross-talk, e.g., the red channel C_r^I is useful to estimate the blue factor W_b. Intuitively, the two reflectance values cannot induce arbitrary variations, and we build an affine combination that recovers the W factors while being insensitive to these variations. We provide a detailed derivation in appendix A.

Discussion. These results show the strong relationship between the observed values $(C_\mathrm{r}^I, C_\mathrm{g}^I, C_\mathrm{b}^I)$ and the unknown $(W_\mathrm{r}, W_\mathrm{g}, W_\mathrm{b})$ factors that we seek. We use this link to guide the interpolation of a sparse set of user-specified constraints and

obtain meaningful results over the entire image. In the next section, we explain how this affine relationship is related to the null space of the Matting Laplacian, and build upon this result to formulate our approach as a standard least-squares problem. Our least-squares approach is robust to other cases, as visible from our examples. Local windows that do not satisfy this model generate a higher residual but do not make the algorithm fail. As long as most windows satisfy these cases or are close to them, our approach produces satisfying outputs. Users can also add scribbles to constrain the solution.

Link with the null space of the Matting Laplacian

First, we summarize the properties and formulation of the Matting Laplacian, and then we explain how to adapt it to white balance.

Background on the Matting Laplacian. Levin et al. [91] introduced the Matting Laplacian in the context of matting, i.e., to extract a foreground element from its surrounding background. They argued that the alpha values that represent the foreground-background mixture at each pixel should locally be an affine combination of the RGB channels. And they showed that this can be modeled with a least-squares functional based on a matrix $\mathbf{M}$ that they call the *Matting Laplacian*. $\mathbf{M}$ is a $n_\mathrm{p} \times n_\mathrm{p}$ matrix, with n_p the number of pixels in the image. Its $\mathbf{M}_{ij}$ coefficient is:

$$\sum_{\substack{k \text{ such that} \\ (i,j) \in w_k}} \left(\delta_{ij} - \frac{1}{|w_k|} \left(1 + (\mathbf{I}_i - \boldsymbol{\mu}_k) \left(\boldsymbol{\Sigma}_k + \frac{\epsilon}{|w_k|} \mathbf{Id} \right)^{-1} (\mathbf{I}_j - \boldsymbol{\mu}_k) \right) \right. \tag{5.11}$$

where i, j, and k refer to pixels, $\mathbf{I}_i$ is a vector containing the RGB components at pixel i, δ_{ij} is the Kronecker symbol , w_k is a window centered on pixel k, $\boldsymbol{\mu}_k$

and $\mathbf{\Sigma}_k$ are the mean vector and covariance matrix of the pixels within w_k, $\mathbf{Id}$ is the 3×3 identity matrix, and ϵ is a parameter controlling the smoothness of the result. Levin et al. showed that one can impose, in a least-squares sense, that the alpha values within each w_k are an affine combination of the RGB channels by minimizing the quadratic form $\mathbf{x}^\mathsf{T}\mathbf{M}\mathbf{x}$ where $\mathbf{x}$ is a n_p-dimensional vector containing all the alpha values.

Null space of the Matting Laplacian. We have shown that for a number of standard cases, the W factors are an affine combination of the chromaticities $(C^I_\mathrm{r}, C^I_\mathrm{g}, C^I_\mathrm{b})$. By definition of the Matting Laplacian, this means that if we build $\mathbf{M}$ using the chromaticity values of the input image instead of the RGB channels, the W factors are in its null space. That is, $\mathbf{W}_c^\mathsf{T}\,\mathbf{M}\,\mathbf{W}_c = 0$ for all c in $\{\mathrm{r}, \mathrm{g}, \mathrm{b}\}$, where $\mathbf{W}_c$ is a large vector containing all the W_c factors of the image. This is a critical result for our task. While several other options exist to interpolate user scribbles, e.g. [97; 35; 6], using the Matting Laplacian, constructed with the chromaticity values, ensures that we produce the correct result in the cases that we studied. In the next section, we build upon this result to design our energy function.

5.2.3 Mixed lighting white balance as optimization

We model white balance as a least-squares optimization. We build the energy, term by term, with the Matting Laplacian first, and then the user scribbles. We get the final result by minimizing the least-squares energy that comprises all the terms that we define below. The Matting Laplacian represents our main prior about the smoothness of the solution. The user scribbles provide additional, linear constraints that further restrict the space of possible solutions to a plausible

one, directed by the user.

Affine combination of the chromaticities. We have shown that in a number of standard cases, the W factors are an affine combination of the chromaticity values, which can be expressed as $\mathbf{W}_c^\mathsf{T}\,\mathbf{M}\,\mathbf{W}_c = 0$ for all c in $\{\mathrm{r}, \mathrm{g}, \mathrm{b}\}$. However, on real-world images, there may be windows that do not fall in one of these standard scenarios. For instance, three or more different reflectances or lights can appear in some windows. We cope with these cases by modeling the affine-combination constraint in a least-squares sense. With the Matting Laplacian, this amounts to a quadratic term:

$$E_\mathbf{M} = \sum_{c\in\{\mathrm{r},\mathrm{g},\mathrm{b}\}} \mathbf{W}_c^\mathsf{T}\,\mathbf{M}\,\mathbf{W}_c \tag{5.12}$$

We use two settings to prevent the system from returning a trivial solution when the users have specified only neutral color scribbles. If the regularization is too weak, and only neutral colors have been indicated, the image can be interpreted as an uniformly white scene illuminated by many different light sources. To prevent this trivial solution, we use a strong regularization with $\epsilon = 10^{-2}$. When other scribbles are provided, the white-scene interpretation does not hold anymore and we relax the system with $\epsilon = 10^{-4}$ so that it better respects edges. Intuitively, the Matting Laplacian regularization factor controls the smoothness prior of our interpolation energy. In practice, we have found that values between $\epsilon = 10^{-4}$ and $\epsilon = 10^{-6}$ work fine for our application.

Neutral color. Users can indicate pixels that have a neutral color. These are usually the first scribbles made by users. For these pixels, the RGB channels should be equal, and using Equation 5.3, we obtain $O_\mathrm{r} = O_\mathrm{g} = O_\mathrm{b} = \frac{1}{3}\sum I$. With Equation 5.4, this gives $\frac{1}{3}W_c = I_c \,/\, \sum I = C_c^I$, which we translate into a

least-squares energy:

$$E_{\mathrm{n}} = \sum_{p \in \mathcal{S}_{\mathrm{n}}} \sum_{c \in \{\mathrm{r,g,b}\}} \left(\frac{1}{3} W_c(p) - C_c^I(p) \right)^2 \tag{5.13}$$

where, $\mathcal{S}_{\mathrm{n}}$ is the set of pixels covered by the scribbles indicating a neutral reflectance.

(a) input (b) user scribbles (c) output, no extensions (d) extended scribbles (e) output, with extensions

Figure 5.2: *Scribbles extension*. Starting from an image under mixed lighting (a), using only the user-provided scribbles (b) does not fully remove the undesirable color cast (c). With the same user input and using our extension algorithm (d), we obtain a visually pleasing result with no color cast (e).

Correct color. Users can also specify that the chromaticity $\hat{C}^R$ currently visible in a region is correct: C^O should be equal to $\hat{C}^R$. Equation 5.10 leads to $\hat{C}_c^R W_c = C_c^I$ and the corresponding energy:

$$E_{\mathrm{c}} = \sum_{p \in \mathcal{S}_{\mathrm{c}}} \sum_{c \in \{\mathrm{r,g,b}\}} \left(\hat{C}_c^R W_c(p) - C_c^I(p) \right)^2 \tag{5.14}$$

Same color. Users can also mark regions that have the same chromaticity. They need not provide the common chromaticity, they only mark pixels that share it. The pixels p covering a scribble $\mathcal{S}_{\mathrm{s}}$ share the same $(\bar{C}_{\mathrm{r}}^R, \bar{C}_{\mathrm{g}}^R, \bar{C}_{\mathrm{b}}^R)$ values. Using Equation 5.10 and summing over all the pixels, we get:

$$\forall c \in \{\mathrm{r,g,b}\}, \quad \bar{C}_c^R \left(\sum_{p \in \mathcal{S}_{\mathrm{s}}} W_c(p) \right) = \sum_{p \in \mathcal{S}_{\mathrm{s}}} C_c^I(p) \tag{5.15}$$

which leads to a linear relationship between $1/\bar{C}_c^R$ and the W_c factors under the scribble:

$$\frac{1}{\bar{C}_c^R} = \frac{\sum_{p \in \mathcal{S}_s} W_c(p)}{\sum_{p \in \mathcal{S}_s} C_c^I(p)} \tag{5.16}$$

Expressing $1/\bar{C}_c^R$ for a single pixel q gives: $W_c(q)/C_c^I(q)$. Since $\bar{C}_c^R$ is constant, we have:

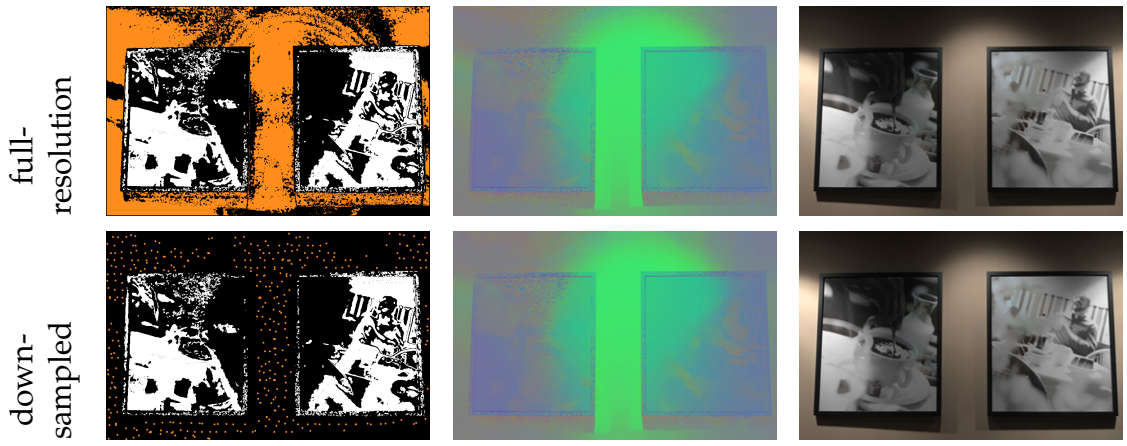
$$\frac{W_c(q)}{C_c^I(q)} = \frac{\sum_{p \in \mathcal{S}_s} W_c(p)}{\sum_{p \in \mathcal{S}_s} C_c^I(p)} \tag{5.17a}$$

$$W_c(q) \sum_{p \in \mathcal{S}_s} C_c^I(p) = C_c^I(q) \sum_{p \in \mathcal{S}_s} W_c(p) \tag{5.17b}$$

Equation (5.17b) avoids division and leads to a numerically more stable scheme. We turn it into a least-squares term E_s:

$$\sum_{q \in \mathcal{S}_s} \sum_{c \in \{r,g,b\}} \left(W_c(q) \Big(\sum_{p \in \mathcal{S}_s} C_c^I(p) \Big) - C_c^I(q) \Big(\sum_{p \in \mathcal{S}_s} W_c(p) \Big) \right)^2 \tag{5.18}$$

We add one such term for each scribble made by users.



(a) constraints (b) W factors (c) results

Figure 5.3: *Subsampling extended scribbles*. We speed up our algorithm by subsampling the constraints defined by the scribbles. We produce similar results with full-resolution scribbles and with their downsampled counterparts. We use the same input as in Figure 5.2.

Extending the scribbles. Our goal is to minimize the amount of user's input that will lead to a satisfying output. We use a similar strategy to Shen et al. [134] and detect points that are likely to have the same reflectance as the pixels covered by a scribble and aggregate these points to this scribble. This automatically adds constraints into our system. This helps propagate scribble information more efficiently, and makes it possible to create satisfying results with only a small number of scribbles as shown in Figure 5.2.

We compare pixels using the Euclidean distance in chromaticity space. Using this metric, our goal is to find unmarked pixels that unambiguously relate to a scribble. For each unmarked pixel p and each scribble $\mathcal{S}$, we robustly estimate how related they are by averaging the distance between p and its 10 nearest neighbors in $\mathcal{S}$. We assign a pixel to the most similar scribble $\mathcal{S}_1$ if two conditions are satisfied. First, the distance s_1 has to be below a threshold t_s. That is, a pixel is added to a scribble only if it is closely related to it. Second, we check that the ratio $s_2/(s_1 + s_2)$ is above a threshold t_r, where s_2 is the second best choice. This ensures that we make only unambiguous assignments for which the second best choice is significantly worse than the best one. In practice, we always use $t_s = 0.01$ and $t_r = 0.8$ (with color channel values between 0 and 1).

Subsampling the same-color scribbles. The same-color scribbles create off-diagonal terms in the linear system. If we use all the pixels covered by these scribbles to define our energy, the corresponding linear system would be dense and slow to solve. Instead, we overlay a grid on the image and select a single pixel for each grid cell. Although a random selection achieves satisfying outputs, we found that we can improve the results by selecting pixels in smooth areas. The rationale is that if the signal varies quickly near a pixel, it may be on an

edge or a corner where the W factors may also be discontinuous. In practice, we estimate the local amount of variation as the variance of I in a 3×3 window centered on the pixel, and we use 10×10 grid cells. To further improve the robustness of our automatic scribbles extension, we do not pick a representative point in a grid cell, if the number of similar pixels in it, as defined in the previous section, is less than 30% of all the points in the cell. Figure 5.3 shows the effects of subsampling.

In addition, we also discard samples where the image is dark because the signal-to-noise ratio in these regions is poor and the signal is unreliable. In practice, we discard pixels for which $I < t_{\mathrm{d}}$ with $t_{\mathrm{d}} = 0.01$. Later on, we fill in missing data in those regions, using interpolation from the nearest image pixels, whose W factor is reliable, i.e., $I_q > t_{\mathrm{d}}$.

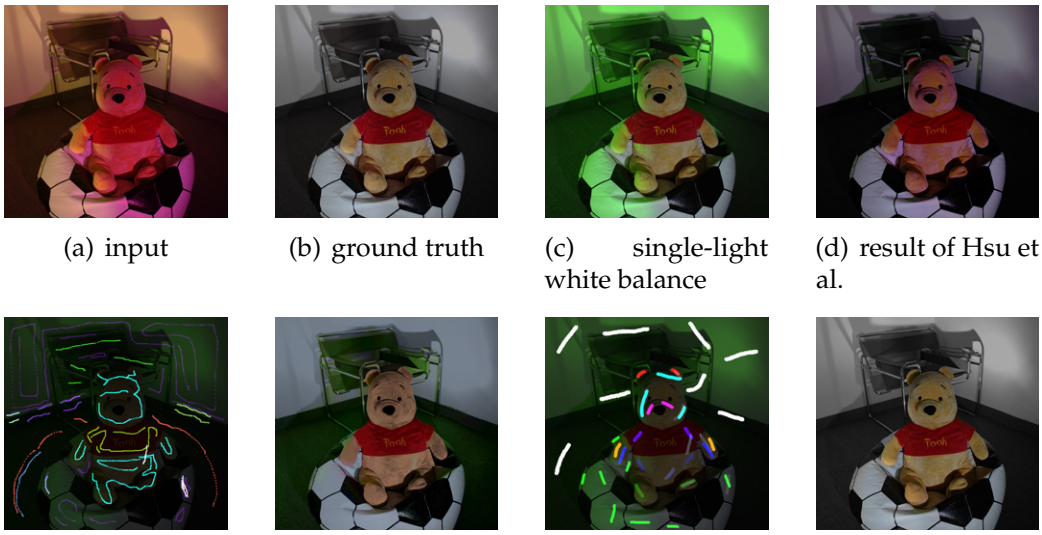
Putting it together. We get the final result by minimizing a least-squares energy that comprises all the terms that we have defined, that is, $E_{\mathbf{M}}$ that seeks to represent the W as an affine combination of the chromaticity channels, E_{n}, E_{c}, and E_{s} that model the users' scribbles. We weight each term to get the energy:

$$E = w_{\mathbf{M}} E_{\mathbf{M}} + w_{\mathrm{n}} E_{\mathrm{n}} + w_{\mathrm{c}} E_{\mathrm{c}} + w_{\mathrm{s}} E_{\mathrm{s}} \tag{5.19}$$

In practice, we use the following weights in all our results: $w_{\mathbf{M}} = 1$, $w_{\mathrm{n}} = 10^3$, $w_{\mathrm{c}} = 10^3$, and $w_{\mathrm{s}} = 10^2$.

5.3 Results

We demonstrate our approach on a variety of scenes, and compare to previous work, when possible. Our prototype is implemented in Matlab. We use the

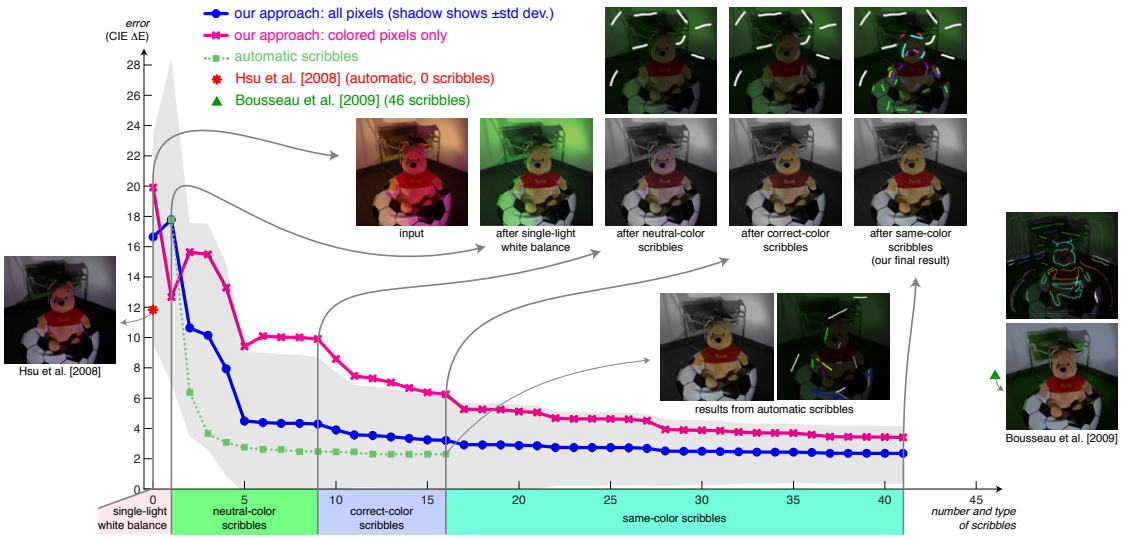


(a) input (b) ground truth (c) single-light white balance (d) result of Hsu et al.

(e) scribbles of Bousseau et al. (f) result of Bousseau et al. (g) our scribbles (h) our result

Figure 5.4: *Ground truth comparison against existing approaches*. We captured two photos with a white light at different positions. We applied different color filters to each image and combined them to get the input image (a). The ground-truth version is a direct combination of the photos taken under the white light (b). Compensating for the color of one of the lights only does not yield a satisfying result (c). Because the areas lit by each light are mostly disconnected, the Pooh for the red light and the wall for the yellow light, the automatic technique of Hsu et al. [75] does not produce a good result (d). The technique by Bousseau et al. [17] relies on scribbles made users (kindly provided by A. Bousseau) (e) and works better, but the colors are desaturated (f). In comparison, our approach uses a number of scribbles on the same order as Bousseau et al. and renders a satisfying result (h) close the ground truth.

standard "backslash" function to minimize Equation 5.19. As long as no same-color strokes are specified (Section 5.2.3), the solver is interactive. When these strokes are present, they add off-diagonal terms and the solver can take up to 1 minute per channel for a 900×900 image. For larger resolutions, we first



(a) Scribbles error evaluation

Figure 5.5: *Scribble evaluation*. We evaluated the effect of our scribbles on the semi-synthetic image (Fig. 5.4). The use of neutral scribbles alone fixes the gray floor and the white wall, but extrapolates wrong colors elsewhere. By adding correct-color scribbles to fix a few of the natural looking colors, produced by the single-light white balance stage, we get better looking regions. Finally, we introduce a couple of same-color scribbles to propagate correct color information to other parts of the image. The solution produced by our system is free of color casts, visually (Fig. 5.4h) and numerically closer to the ground truth, compared to the methods of [17] and [75].

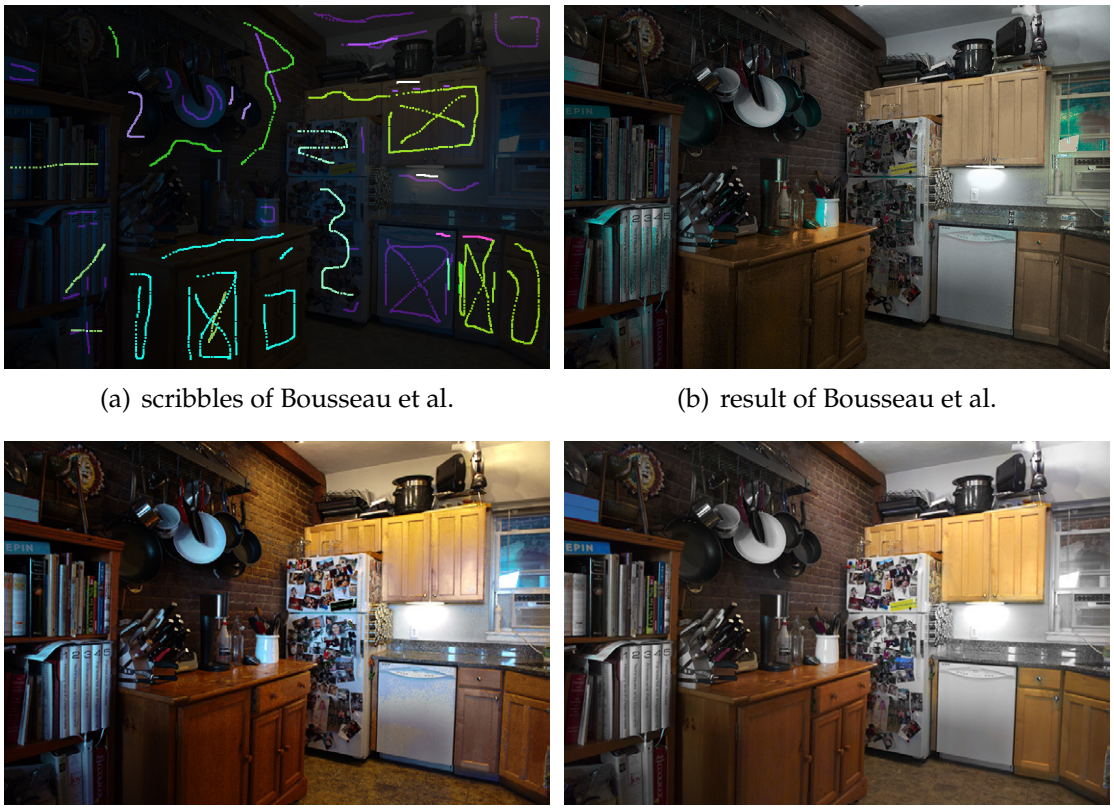
downsample the image, solve for the W factors, and upsample them using Joint Bilateral Upsampling [86].

The kitchen in Figures 5.1 and 5.6 is a common case of an interior scene with multiple light chromaticities, which our technique can tackle. In contrast, existing tools cannot remove the color cast in many regions. Single-light white balance only improves part of the scene and produces strong color casts everywhere else (Fig. 5.1b). The method of Hsu et al. [75] (Fig. 5.6c) shows its limitations

on this photo because there are three light sources, all of a different color, and they only handle two. Even though it was not designed specifically for this task, we can use the chromaticity of the shading obtained by Bousseau et al. [17] for white balance correction. However, in scenes such as the kitchen, this produces an overall desaturated result (Fig. 5.6b). This is probably because their method was designed for intrinsic images, not white balance, and their equations are derived for the case of monochromatic lights only. They explain that the use of the equations on a per-channel basis is only a heuristic. Furthermore, their method cannot handle well black-and-white reflectance variations such as the books in the lower-left of Fig. 5.6(b), which leads to blue color artifacts there.

Figures 5.10 and 5.8 show that, in the two-light scenario, our approach performs as well as the method by Hsu et al. [75], and better than that of Ebner [50]. The main difference with the former is that our approach does not assume the light colors to be known a priori and relies on user input. Moreover, as previously discussed, and unlike the technique of Hsu et al., our approach can also cope with more than two lights.

Figure 5.11 shows that our approach can deal with complex materials, such as fur. A few, easy to specify, same-color scribbles are enough for our system to produce a result that is free of color casts (Fig. 5.11e). For comparison, the system of Hsu et al. [75] struggles to estimate the two light mixtures because of the complex appearance of the fur (Fig. 5.11c).



(a) scribbles of Bousseau et al. (b) result of Bousseau et al.

(c) result of Hsu et al. (d) our result

Figure 5.6: *Comparison against prior work on the kitchen scene*. Bousseau et al. [17] produce results with desaturated colors (b), e.g., the cabinetry. The two-light technique of Hsu et al. [75] does not fully remove color casts on the left wall and on the dishwasher because there are three different lights (c). In contrast, our approach produces a satisfying output (d).

5.3.1 Evaluation using ground-truth data

We use the ground-truth data provided by Hsu et al. [75] to evaluate the performance of our approach by comparing it to their results (reproduced from their article), and to the results of Bousseau et al. [17] (kindly provided by A. Bousseau). Our result (Fig. 5.9c) is numerically and visually closer to the ground-truth image (Fig. 5.9e). We provide more details in [18].

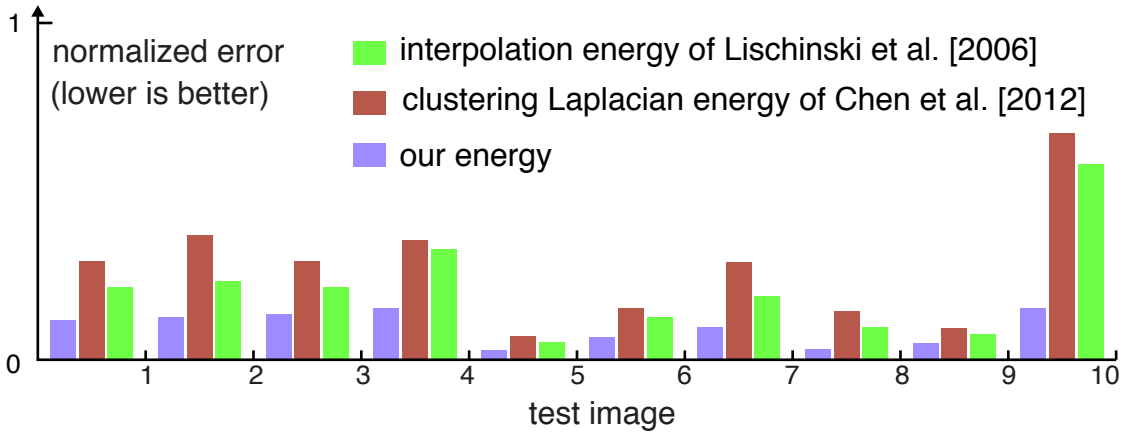


Figure 5.7: *Comparison of our energy to prior work.* We compare our energy to the all-purpose interpolation energy proposed by Lischinski et al. [97], and to the clustering Laplacian energy by Chen et al. [37]. With a sparse set of constraints, our energy interpolates missing values that are closer to the ground truth light mixtures. This corresponds to visually better white balanced results (see [18]).

Energy evaluation. Figure 5.7 evaluates how well our energy models the W factors for various ground-truth mixed lighting conditions. We show comparisons to the all-purpose interpolation energy proposed by Lischinski et al. [97], and to the KNN Matting energy [37], which improves the clustering Laplacian based on the nonlocal principle [90]. We used 10 pairs of input/ground truth images, and for each one of them we randomly placed a small set of known-white-balance constraints through the entire image; approximately 0.001% of all pixels in an image were constrained (every 300×300 pixels on average). Using the same constraints, we optimized the three energies to interpolate the missing values. We repeated this step 10 times for each of the 10 data sets, and report the mean relative per pixel error for the W factors. The plot shows that for the problem of white balance our energy performs consistently better than both the all-purpose energy and the clustering Laplacian energy (visual results are in [18]). We also

tested the clustering Laplacian energy computed using the chrominance channels and observed only a minor improvement. On average, the Matting Laplacian still performed 2.5× better. The general-purpose energy and clustering Laplacian rely on appearance similarity to interpolate the missing data. While this performs well for some applications, e.g., [37] achieves state-of-the-art matting results, our experiment suggests that the affine model of the Matting Laplacian is better suited to the white balance problem.

Scribble impact. In Figure 5.5, we evaluate the impact of our scribbles on an image with ground truth (Fig. 5.4). The neutral-color scribbles quickly improve the overall result. Then, correct-color scribbles help identify the regions that look good at this stage. Finally, the same-color scribbles help propagate this information to other areas through the scribbles extension mechanism. To isolate the effect of the white wall in the background, we also plot the energy restricted to the color pixels as determined by the ground-truth data. The initial gain from the neutral-color scribbles is lower, but the trend remains the same. We provide another such analysis in [18].

User-independent evaluation of our model. Scribble-based techniques are hard to evaluate because they depend on a user's decisions. It makes it difficult to get a sense of their convergence or the amount of user annotations that are fundamentally needed. We propose a new methodology to evaluate scribble-based approaches independent of human input and based on ground truth data. Ideally, we would like to know how a method performs with the best possible user input. To make things tractable, however, we propose an approximation based on a set of tentative scribbles obtained from an image segmentation, and a

greedy strategy. By definition, this does not inform us on how easy it would be for a user to choose these scribbles, but it provides strong information about the adequacy and conciseness of the mathematical formulation, and a reference to evaluate human performance.

First, we segment the input image using Quick Shift [145], and pick the scribble in each segmented region to be as long as possible by fitting an ellipse to the segmented region, and finding its maximal axis. Each segmented region is then assigned a chromaticity value equal to the mean chromaticity of the pixels composing it. By comparing to ground truth, and simple thresholding, segments can be determined to be "neutral" or "correct" in the single-light white balance image. Further, pairs of segments can be deemed to be "same color". This segmentation and assignment gives us our set of potential scribbles.

For each potential scribble, we evaluate the error, the CIE Lab L2-norm, of the image, compared to ground truth, if we applied that scribble. We then greedily apply the scribble that decreases the error most. Then we iterate, currently using a brute-force evaluation of all possible scribbles at each iteration.

The green curve in Figure 5.5 plots the error of the greedy evaluation. Even though it is greedy, and not optimal, it is lower than any current approach. In particular, these greedily-chosen scribbles, based on full ground truth knowledge, show that our user did not make optimal choices, but got close with additional scribbles.

5.3.2 Discussion and limitations

Since our approach is user-driven, the quality of the result depends on the amount of interaction that the user performs. We found that simple scenes require only a few scribbles, e.g., 9 in Figure 5.2, and more complex scenes may need up to 50 scribbles (Figure 5.1). Although it is difficult to quantify, our scribbles are relatively easy to use. Qualities such as "this is a white wall" and "this region has a uniform color" are easy to determine for users.

The usefulness of the "correct single white balance" strokes depends on the choice of the white balance settings. However, such strokes could be extended to allow the user to explore multiple options for single white balance and use constraints from these multiple versions.

We designed our algorithm using linear RGB values, which can be easily obtained from the RAW image files produced by DSLR cameras. Although JPEG files produced by cameras are processed to make them more appealing, e.g., to increase their contrast and saturation, we found that our approach is robust enough to handle them. For instance, Figure 5.12 shows a sample result on a JPEG photo for which we assumed a standard gamma of 2.2.

Our current implementation uses a vanilla solver. The first few strokes result in interactive feedback, but later passes take about a minute per color channel. A multi-grid solver would dramatically reduce this cost.

Our user-driven approach to the many-lights white balance problem relies on the subjective judgement of the users. This provides important constraints for our interpolation scheme, which tends to produce results that are plausible or pleasing, but not necessarily physically accurate. For example, in Figure 5.8, the

ground-truth left and right pages have slightly different reflectances, which the user did not know, which leads to numerical error.

5.4 Conclusions

We present a practical method for high-quality white balancing in scenes with complex lighting based on user-provided scribbles. It relies on what is most intuitive to humans, reflectance properties. Our contributions are a new formulation of the white-balance problem based on intensity preservation, a study that shows that important canonical local configurations are in the null space of the Matting Laplacian, an interpolation energy that performs better for white balance than generic approaches and strategy to extend constraints and reduce the required user interaction.

In the future, we would like to explore posibilities of setting the sparse per-pixel constraints automatically, based on the image content. A data-driven approach can be used to learn a mapping from image features to typical colors under white light. We would also like to explore faster ways for solving the resulting large linear system with linear constraints. A GPU solver with a coarse-to-fine approach is one possible way, but further structure of the matrix can be exploited.

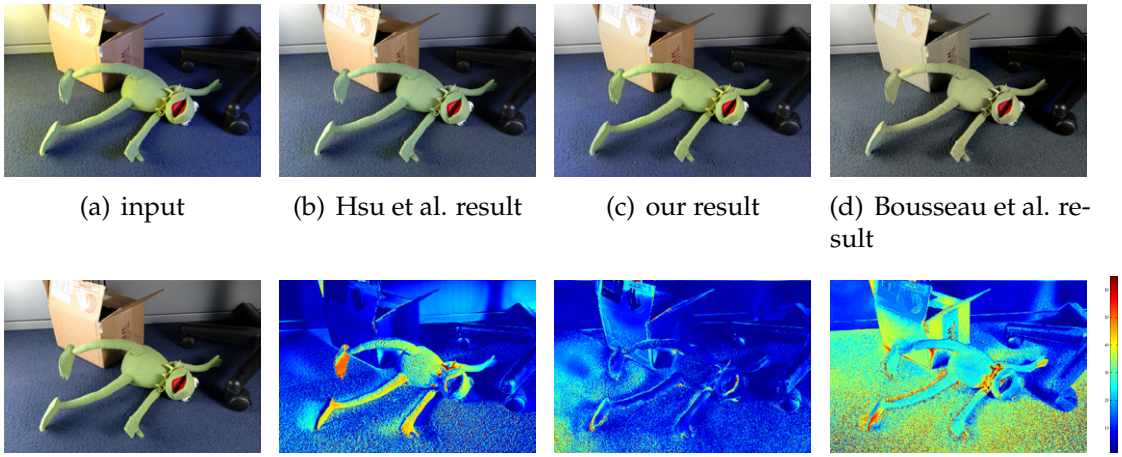
(a) input and results

(b) difference from ground truth

(c) ground truth

(d) scribbles used by our method

Figure 5.8: *Groun truth comparison on the apple scene*. On a ground truth data set, provided by Hsu et al. [75], our scribble based method produces results, comparable to their automatic system. However, notice that we do not assume anything about the lighting in the scene, while their approach is limited to two known light sources.



(a) input (b) Hsu et al. result (c) our result (d) Bousseau et al. result

(f) ground truth (g) Hsu et al. error (h) our error (i) Bousseau et al. error

Figure 5.9: *Ground truth comparison on the kermit scene*. Ground truth evaluation with image data from [75], shows that our approach produces the result (c), closer to the ground truth image (e), compared to competing approaches, (b) and (d). We plot the numerical error, using the L2-norm of the Lab difference between each method and the ground truth data.

(a) input (b) Ebner et al. [50] (c) Hsu et al. [75] (d) our result

Figure 5.10: *Comparison on a prior data set*. Comparison on images from Hsu et al. [75] Ebner's method [50] tends to produce desaturated results (b). These results are reproduced from Ebner et al. [50]. Hsu et al. [75] (c) and our method (d) produce equivalent results with no color cast. The two approaches differ in that Hsu et al. is automatic but assume two lights of known colors, whereas our method is user-assisted and does not make assumptions about the illumination. The results by Hsu et al. are reproduced from their article.

(a) input (b) single-light white balance (c) Hsu et al. result (d) our scribbles (e) our result

Figure 5.11: *Mixture of outdoor and indoor lighting*. The scene is lit by outdoor and tungsten lights, making the cat looks yellow (a). The single-light white balance improves the white part of the fur, but makes the gray suitcase appear blue (b). The technique of Hsu et al. [75] does not produce a good result because of the complex appearance of the fur (c). Using our scribbles (d), our result correctly captures the white fur and the gray suitcase (e).

(a) input (b) neutral and correct color scribbles (c) after adding neutral and correct color scribbles (d) adding same color scribbles (e) final output

Figure 5.12: *Result on a JPEG photo*. We assumed a standard 2.2 gamma correction to convert the RGB values into linear space. Our method is sufficiently robust to produce a satisfying result (e).

CHAPTER 6

BAND-SIFTING DECOMPOSITION FOR IMAGE BASED MATERIAL EDITING

In this chapter, we switch gears from lighting design and focus on surface properties, which in combination with lighting governs how materials appear when photographed. We introduce a new image-based approach for perceptually manipulating material properties related to shininess, smoothness and weathering. We validate the perceptual consistency of our approach through user studies; and we demonstrate its usefulness for both image and video post-process material editing. This work has been accepted, pending a minor revision, to ACM Transaction on Graphics 2015 [25].

6.1 Introduction

Photographers care a great deal about the surface appearance of objects they photograph; indeed, much of the craft of traditional photography involves controlling material appearance using physical techniques. Portrait photographers control the appearance of skin wrinkles by adjusting the lighting, apply makeup to hide variation in skin color (e.g., blemishes or mottling) and powder to make skin appear less shiny. In product photography, dulling spray is used to reduce specular highlights, while in food photography, where specularity may be desirable, a glycerine spray may be used to make the food look fresher or juicier.

Such adjustments can be performed digitally, after the photo was taken, rather than physically during the photo session, which greatly simplifies the process and enables more control on the result. However, altering the appearance

of material properties such as wetness, gloss, wrinkles, or mottled coloration remains a tedious task that requires advanced skills that are beyond the reach of casual users. Further, in the case of video, laborious manual retouching is simply impractical; not only are there multiple frames, but it is difficult to get the effects to align and adjust across the sequence without introducing temporal artifacts.

An alternative route to manipulating material appearance is to build a fully renderable 3D description of the scene, and to change the physical parameters as needed. For example, multi-image capture with multiple cameras and light sources can create a highly detailed model of an object. However, this acquisition pipeline is not helpful to a casual or professional photographer working with a single image from an ordinary camera.

Our goal is to work with an ordinary photograph, and to allow the photographer to alter the appearance of a 3D surface without using a 3D representation. By using 2D image operations, we gain both speed and simplicity. We want our image-based technique to accept many kinds of source images as input, and to avoid the errors that can arise when attempting a full 3D scene analysis. At the same time, we must recognize that 2D operations can be limited and work best when there is a straightforward mapping between 3D surface properties and 2D image properties.

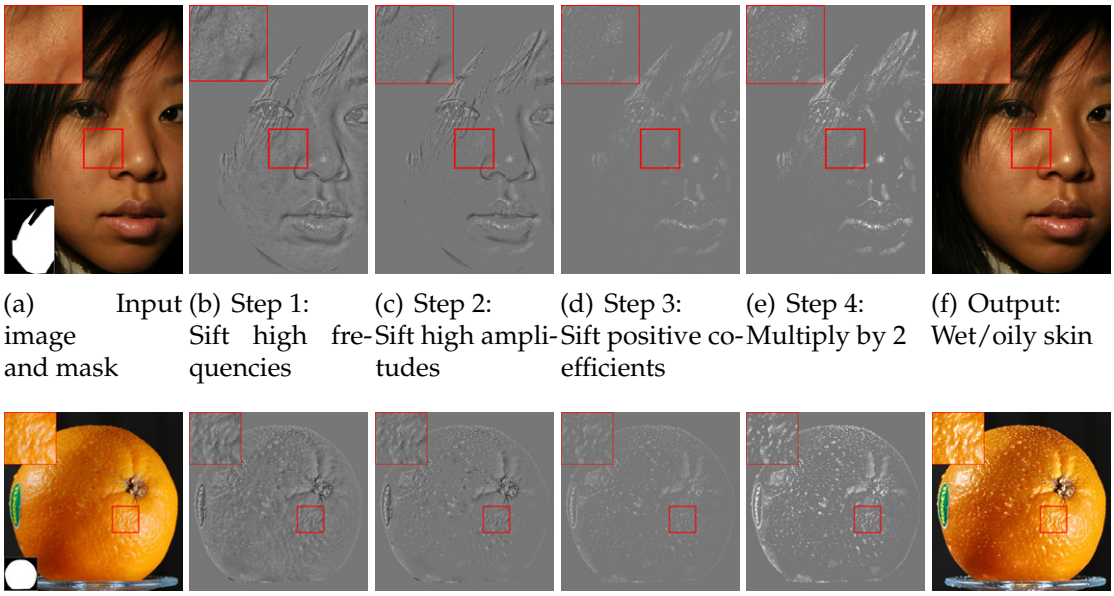
This chapter explores a space of image operators that can be used to modify a variety of visual surface properties. The operators decompose an image by first applying a series of splitting operations based on frequency, amplitude, sign, and then *sifting* through these decompositions and recombining them to compose a new image. This sifting operation can create a range of visual effects that affect perceived material properties, for example, by changing perceived

shininess/gloss, aging/weathering, and glow. Figure 6.1 shows an example of one such *sifting* procedure for two example images.

By selectively modifying coefficients in different subsets, one can achieve a variety of distinct image operators that we call *sifting operators*. While some of these are known and well studied; for instance, increasing high-frequency coefficients enhances detail in an image, the combination of the several criteria has not been explored so far. This chapter seeks to fill in this gap and focuses on aspects related to material perception in particular. That is, our goal is to characterize which band-sifting operators generate physically plausible change that lead to perceptually consistent effects. For example, in Figure 6.1 we show that the same band-sifting operator makes both the human skin (row 1) and the orange surface (row 2) look more shiny and wet.

We explored the space of band-sifting operators and found that depending on the selected coefficients, operators modify properties such as the material shininess or its degree of weathering. On faces, the effects were particularly interesting with, for instance, variations in the appearance of oiliness, glow, wrinkles and pigmentation of the skin. Figure 6.2 shows examples of these various effects. As expected, we also observed that applying a modification too strongly yields unnatural looking results. This motivated two user studies. First, for each band-sifting operator, we characterized how strongly it can be applied before producing an unnatural look. And second, we studied how human observers describe the effect of each operator. This allowed us to isolate a subset of band-sifting operators that produce consistent effects across images. Finally, we also demonstrate the use of the band-sifting operators on videos. They are stable enough to achieve temporally coherent results without additional

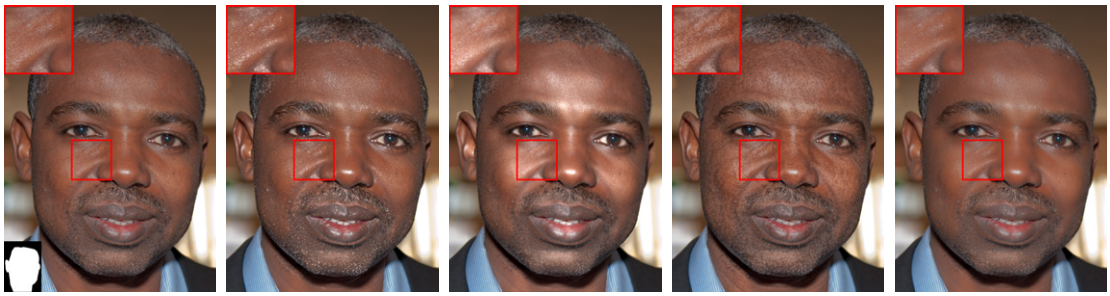
processing, they are fast enough to run at interactive rate, and they naturally "follow the scene content" without the need to estimate the optical flow explicitly. As an example, in a video of someone talking, we can add some glow onto them or make them look sweaty by simply applying our operators frame by frame, which is both simple and efficient.



(a) Input image and mask
(b) Step 1: Sift high frequencies
(c) Step 2: Sift high amplitudes
(d) Step 3: Sift positive coefficients
(e) Step 4: Multiply by 2
(f) Output: Wet/oily skin

(g) Input image and mask
(h) Step 1: Sift high frequencies
(i) Step 2: Sift high amplitudes
(j) Step 3: Sift positive coefficients
(k) Step 4: Multiply by 2
(l) Output: Wet/shiny surface

Figure 6.1: *Example of one of our band-sifting operators*. Starting from a single input image and a mask (a), we selectively manipulate the subband coefficients of the luminosity channel by *sifting* them through a cascade of decisions based on the scale, amplitude and sign of the coefficients. Here we show one of these decision paths. First, we *sift* the high-spatial frequencies from the low-spatial frequencies (b). Then we *sift* the high amplitudes from the low amplitudes (c), and finally we *sift* the positive from the negative coefficients (d). Multiplying the *sifted* coefficients (e), adding them back, and reconstructing the image gives the skin a more oily or wet look (f). In the second row, we show that a similar perceptual effect is achieved on a non-face object, where the orange is given a more shiny or wet look. We found that *sifting* subband coefficients allows us to produce a variety of physically plausible effects that lead to perceptually consistent modifications across a variety of scenes.



(a) Original image (b) Wet/oily skin (c) Smooth/shiny glow (d) More blemishes (old) (e) Fewer blemishes (young)

Figure 6.2: *Example of our band-sifting operators on human faces.* Our band-sifting operators are particularly useful for manipulating material properties in human faces. (a) Original image, with detail inset at upper left, and mask inset at lower left. (b) We sift and then boost the high amplitude, positive coefficients in the high-spatial frequencies which gives the skin a more shiny or wet look. (c) We manipulate the positive low-spatial frequencies coefficients which gives the skin a soft glow. (d) We produce an aging effect by emphasizing blemishes and pores that are not noticeable in the input image. We achieve this by sifting and then boosting the low amplitude coefficients in the high-spatial frequencies. (d) We reverse the effect, i.e., reduce blemishes and pores, by decreasing the sifted coefficients from (d).

6.2 Band-sifting Operators

In this section, we describe the space of band-sifting operators that is at the core of our work. We strike a balance between two objectives. We define a space that is both expressive enough to include an interesting variety of effects and concise enough to allow for an exhaustive study.

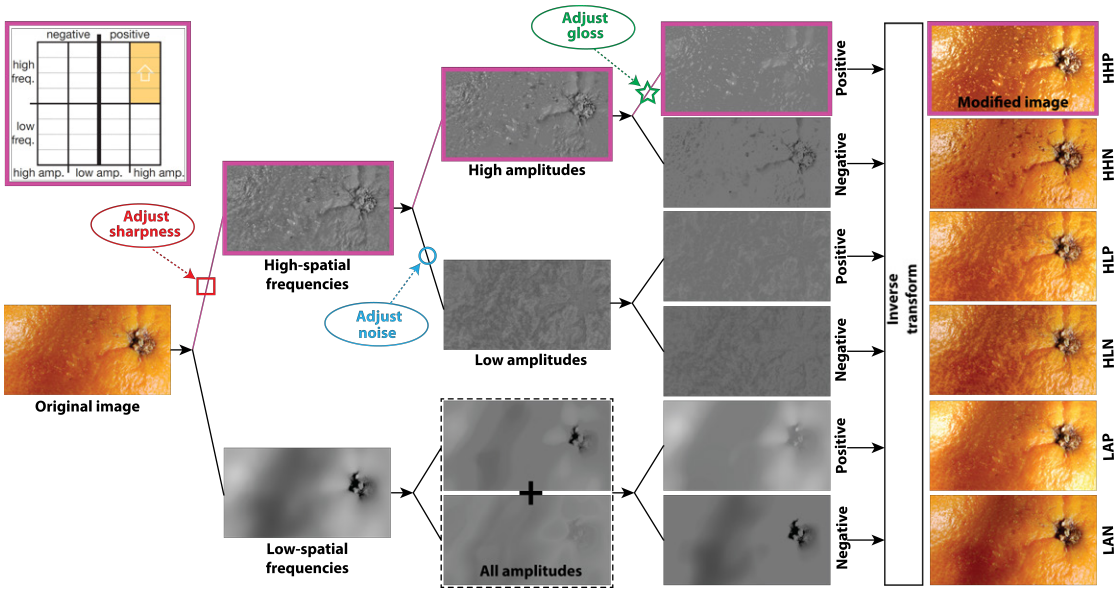


Figure 6.3: *Conceptual diagram of our band-sifting space*. Given an image, we split it into high and low frequency subbands. These are then split into high and low amplitude parts. These are further split into positive and negative parts. For visualization purposes we show only two frequency splits, but in practice we create $\log_2(\min(width, height))$ frequency subbands and work on each one of them. Further, in order to make the size of the space more tractable, we "compress" the set of possible choices by looking at two categories of frequencies. We consider the high-to-mid frequencies as one category, which we refer to as "high spatial frequencies", and we look at the mid-to-low frequencies as another category, which we call "low spatial frequencies". Further, as shown in the diagram, we do not split the low-spatial frequencies category based on the amplitude of the coefficients, since numerically, *sifting* based on this criterion does not give much differentiation. However, the sign of the coefficients is still a useful *sifting* criterion along the low-spatial frequencies paths, e.g. it differentiates between broad-gloss and broad-shadow effects. With colored text and arrows we show how various operators can be mapped into paths in our space. With purple borders we show the path of *sifted* coefficients that was used to generate the orange result in Figure 6.1, second row.

6.2.1 Motivation for Band-sifting Operators

As we discussed in Section 2.3, many existing techniques can be interpreted as splitting an image according to a specific criterion like scale and amplitude, manipulating one of the generated components, usually with a simple operation like a multiplication, and recombining the result to form the final image. Our work extends this approach by decomposing images using several criteria at the same time. Figure 6.3 shows how we build our operators. We first split the original image into high and low-frequency subbands. We then separate the subbands into their high and low-amplitude parts. And we finally split the coefficients according to their sign, positive or negative. By placing multiplicative "control knobs" at specific points in this flow diagram, one can modify sharpness (shown with a square), noise (shown with a circle), or gloss (shown with a star), similarly to previous work (section 2.3). This illustration also suggests that there are many other ways to use our image decomposition, which raises several questions. What can one do by putting control knobs in other places? Are there more useful operators waiting to be found? What about yoked control knobs working on more than one component at a time? And what subband transforms are best?

Of course, there are any number of ways to increase the efficacy and complexity by adding in other techniques from image processing, computer vision and machine learning. However, our purpose here is to understand what is possible while staying within this scheme. Even with this restriction, there is plenty of territory to explore, and useful operators can serve as a starting point for later improvements.

Our goal here is to ask what can be done by simple manipulations of multi-

scale transforms. By staying close to the original image data, we maintain locality and avoid any propagation of artifacts. We also avoid the fragility that can occur, for example, when imposing specific physical models or elaborate priors. We accept arbitrary images as input, and in our experience, the image modifications look "natural" as long as they are not pushed too far.

6.2.2 Three Sifting Stages

We now describe the stages that we use to decompose images. We start by constructing a multiscale image decomposition and sift the subband coefficients based on three criteria: scale, amplitude, and sign.

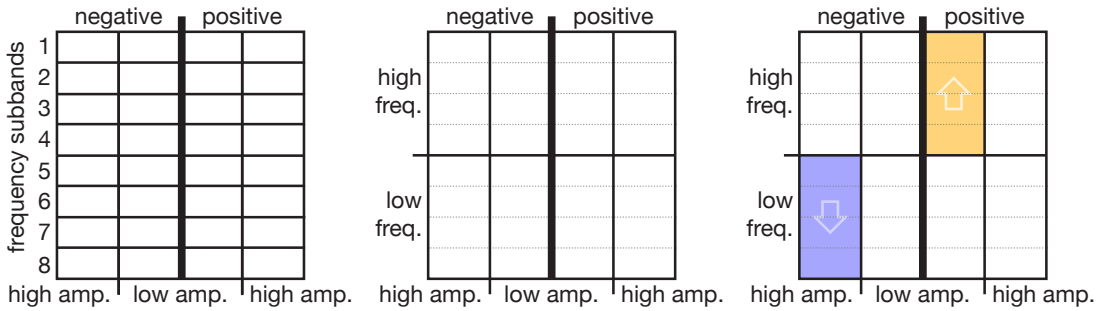
Scale. Our design space follows a common trend and acts upon a multiscale image decomposition [29; 36; 56; 53; 54; 70; 116]. Such decomposition provides us with a set of subbands that can be thought of as an over-complete wavelet representation in the sense that each coefficient represents details at a given location and scale. This latter aspect is our first sifting stage: we allow our "sieve" to act selectively upon the large-scale or small-scale coefficients, or on all of them (i.e., both large and small scale). Intuitively, this separation differentiates between small elements, such as skin pores on a face, and bigger ones like large-scale shading and shadow variations.

Amplitude. Our second sifting criterion is the amplitude of the coefficients. We separately manipulate coefficients with a high or low amplitude, or both (high and low). This sieve separates low-contrast from high-contrast features. It is

related to wavelet coring [45; 136] with the major difference that we keep and process the low-amplitude coefficients instead of discarding them.

Sign. The third sifting criterion differentiates coefficients based on their sign: positive or negative. Recent studies [132] have shown that the skewness of the subband coefficient distributions, i.e., the asymmetry of the coefficient histograms, is correlated with the perception of lightness and gloss. From a numerical perspective, there is not a single well-defined way to alter skewness, that is, the same skewness value can be achieved with many different transforms. In our work, we modify coefficients based on their sign, which gives us a direct control on the distribution symmetry. This approach has an intuitive interpretation, the positive subband coefficients describe bright features like highlights, and the negative ones capture features like crevices, holes, and shadows in wrinkles.

6.2.3 Refining the Scale Sifting Criterion



(a) Full decomposition (b) Pruned space (c) Candidate operator

Figure 6.4: *Compact diagram of our space*. In (a), we show a full decomposition into 8 subbands with each subband partitioned into 4 parts based on amplitude and sign. This gives 32 cells. Each cell can have one of three knob settings: boost, reduce, or leave the same. This gives 3^{32} potential configurations, which is hopelessly large. So we cluster the bands into two categories (high spatial frequencies and low spatial frequencies) giving 8 cells, as shown in (b). For each band-sifting operator, we show the knob setting of each cell with an arrow and a color, as shown in (c), where the low-amplitude positive high-frequency coefficients have been boosted, and the high-amplitude low-frequency negative coefficients have been reduced. The set of 3^8 configurations is still large; see text for further methods to reduce dimensionality.

The scale criterion raises two nontrivial issues: how many subbands to use and how to compute them. The rest of this section discusses these two issues.

Constructing the Scale Subbands. We started our study using a standard Laplacian pyramid [29] that has the advantage of great simplicity. However, early in our investigations it became apparent that the Laplacian pyramid introduced artifacts at edges, which is a common issue of using linear filters on natural

images. We therefore investigated pyramids based on edge-aware filters. We tried three such filters: the Bilateral Filter [142], the Weighted Least-Squares filter [53], and the Guided Filter [72]. All gave a significant reduction in edge artifacts (see [22] for comparison). We chose to use the Guided Filter (used for all results in this paper), but other filters would presumably give similar results.

Number of Subbands. For our study, we used images at the resolution of typical monitors, e.g., the longer side set to 512 pixels. Using a factor of 2 in resolution between each subband, this yields 8 subbands. Then, the sign and amplitude sifting generates 4 components for each subband, and each of these components can either be boosted, reduced, or left unchanged. This would leave us with 3^{32} operators to explore, which is impractical. We take a few steps to make this number more tractable. The first one is to group the subbands into two sets: the high and low-frequency subbands, which leaves us with 3^8 possible combinations. Figure 6.4 illustrates the decomposition we use. However, this number is still too large for the purposes of our exhaustive perceptual studies. In the next section, we further discuss how to reduce the space to a more manageable size, while ensuring that we still have a variety of distinct nontrivial effects to study.

6.2.4 Early Pruning

Even with the subband grouping described in the previous section, the space of possible band-sifting operators remain challenging to explore. In this section, we explain how we structured the space to make its exploration tractable.

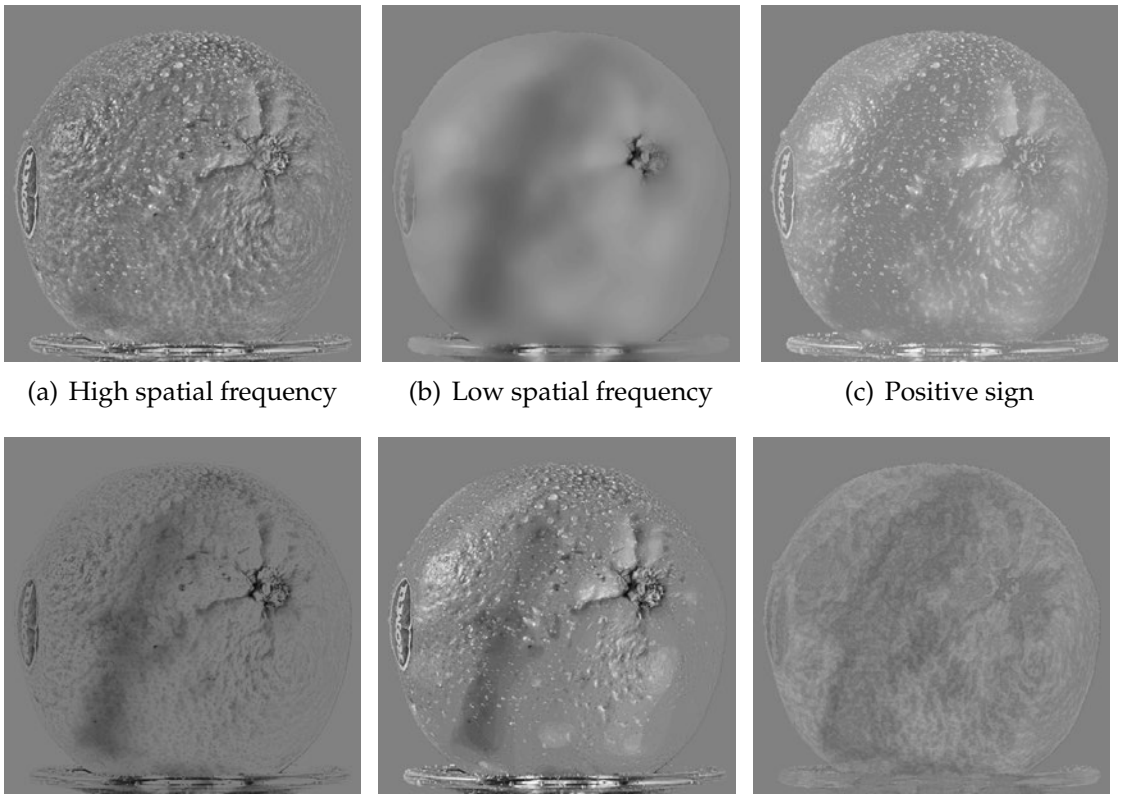
Independent Criteria. First, we apply the sifting criteria independently of each other. For instance, for the sign, we choose between positive, negative, or both, and apply this choice to all the subbands. This gives us 3 sifting criteria (scale, amplitude and sign), with 3 options for each of them: {high (H), low (L), all (A)}, {high (H), low (L), all (A)}, and {positive (P), negative (N), all (A)} respectively. Once we have selected which coefficients to modify we can either boost (B) them or reduce (R) them. This defines $3 \times 3 \times 3 \times 2 = 54$ combinations in total. Figure 6.5 illustrates these 3 criteria.

Removing Redundancy. We explored and evaluated the space of operators in a set of pilot studies. We found that it could be reduced to a more useful set due to some redundancy in the effects achieved. Therefore, we applied the following pruning based on our observations.

– High frequencies tend to mask low frequencies and there is no visually significant difference between paths that sift only the high frequencies and those that sift all of them. Therefore, we do not include the latter in our study (pruned space: 2×3×3×2 = 36).

– All-amplitudes and high-amplitudes paths also produce visually similar results because they differ only due the low-amplitudes coefficients that are small by construction. We do not include the all-amplitudes paths in our study (pruned space: 2×3×2×2 = 24).

– Low-spatial frequency coefficients come from repetitively smoothing the input image and most of them are very small. Sifting these coefficients based on their amplitudes leave only very few significant values, and the corresponding modifications have almost no effect as can be seen in Figure 6.3. We avoid the high vs low-amplitudes paths and only include the all-amplitudes ones for the



(a) High spatial frequency (b) Low spatial frequency (c) Positive sign

(d) Negative sign (e) High amplitude (f) Low amplitude

Figure 6.5: *Visualization of the subband coefficients* sifted *by each independent criteria*. We visualize each possible selection by setting to 0 all the non-selected coefficients. We show the individual subbands for this input image in [22]. High spatial frequencies (a) capture features like small-scale bumps and wrinkles, and low spatial frequencies (b) mostly represent large-scale shading variations. Positive (c) and negative (d) coefficients show highlights and shadows respectively. Finally, high-amplitude coefficients (e) represent specular highlights and deep shadows, while low-amplitude values (f) capture more subtle reflectance variations.

low-spatial frequency paths in the study. So for high spatial frequencies there are (P|N|A) $\times$ (H|L) $\times$ (B|R) = $1 \times 3 \times 2 \times 2 = 12$ operators. And for low spatial frequencies there are (P|N|A) $\times$ (A) $\times$ (B|R) = $1 \times 3 \times 1 \times 2 = 6$ operators. For a total of 18 possible operators.

– Also, reducing the low-amplitude coefficients does not have a noticeable effect since their value is already low. We do not include these paths in our study. This eliminates 3 possible operators, giving us a final total of 15 operators.

Another way of thinking of the space is: 9 operators where boost is applied, and 6 operators where reduce is applied.

6.2.5 Physical Observations

Appearance properties, such as luminance variations on surfaces, come from many physical sources, and a key reason why our approach works well is that in many cases these sources correspond to different bins of our subband decomposition. We discuss a few such examples below.

- Specularities typical of wet and glossy surfaces are bright and small, and mostly fall in the high spatial frequencies, with high amplitude and positive sign.
- Pits and grooves, including the wrinkles and pores of the skin, tend to be dark due to self-shadowing, and their magnitude is often medium or large. Because of this, they appear in the high spatial frequencies, high amplitude, negative coefficients.
- Variations in albedo, caused by dirt, stains, age, wear, or other degradations, tend to be low in amplitude compared to dark pits and bright highlights, and often shows up in the low-amplitude negative coefficients.

Such characteristics of physical objects are common and provides a solid underpinning to our approach. This also points at a limitation of our operators. If

an object does not exhibit such properties, our operators are not effective. For instance, we cannot make a perfectly smooth object look rough. To do so, one would need to hallucinate surface details that do not exist in the original image. We believe that this is an interesting direction for future work.

6.3 Implementation

In this section, we describe the actual implementation of the band-sifting operators that we described in the previous section. We provide detailed pseudo-code in Algorithm 1.

Our multi-scale decomposition is akin to that of Farbman et al. [53]. We repetitively process the input image with an edge-aware filter, doubling its spatial extent each time. This produces a series of images of increasing smoothness. Taking the difference between two such successive images gives us frequency bands, a.k.a. subbands, that contain details of a given size. Since we preserve edges in this construction, we do not downsample the subbands to prevent aliasing, i.e., each subband has the same resolution as the input. We use the Guided Filter with its default regularization parameter ($\sigma_r = 0.1^2$) [72] for edge-aware filtering.

In our prototype, the multi-scale decomposition and the subband-sifting procedure on the GPU were implemented using C++ and OpenCL. To accelerate the Guided Filter on the GPU, we implemented an efficient summed-area table algorithm [71] that we use as a building block for all the box filters, mean and standard deviation computations required by the Guided Filter approach. This allows us to achieve interactive frame rates (5–6 fps) for 1-megapixel videos, which

is sufficient for preview purposes before running the full-resolution computation off-line.

For our study, we fix the long edge of the input image to 512 pixels, and compute 8 subbands. We split them into 4 low-frequency subbands and 4 high-frequency ones. For the amplitudes, we use the standard deviation of each subband as the threshold between the high and low categories. To avoid the artifacts that a hard threshold would introduce, we use a soft transition spanning ±20% around the standard deviation. For instance, if an operator multiplies by 2 the high-amplitude coefficients in a subband where the threshold is σ_t, the multiplication factor is 1 below $0.8\sigma_t$ and 2 above $1.2\sigma_t$, and smoothly varies in between. Finally, the increasing or decreasing of the selected coefficients is performed with a simple multiplication factor greater or lower than 1.

6.4 User Studies and Results

In this section, we describe the user studies that we performed to characterize which operators produce effects that are natural and how they affect material appearance. We then present more results on still images and videos.

To understand the visual impact of our band-sifting operators, we conducted two user studies to validate their perceptual effects. The goal of the first study was to find which operators are natural, i.e., for a given image, what is the range of multiplication factors within which an operator produces a discernible and natural-looking change? This study tells us how much we can boost or reduce an effect before it starts to look unnatural on a certain image.

The second study asks users to describe the visual change that operators produce. This task shares some similarities with a recent line of work in computer vision, where the goal is to describe images through high-level *attributes* [117]. In our work, we are interested in assigning attributes related to the perceptual changes produced by the band-sifting operators. We use 16 categories of words describing various material-specific properties. We designed the set of words in a pilot study, between 3-4 people related to the project, by looking at the perceptual effects produced by the operators on tens of examples. Figure 6.8 lists these words. Some of those categories of words describe low-level features, such as "wrinkled, pitted, bumpy" and other categories describe high-level properties, such as "young, new, fresh". Then, in our study with casual users, participants were shown the original image and the modified image, and were asked to pick all categories of words that apply.

6.4.1 Study 1: Natural vs Unnatural

The goal of this study is to find whether there is a reasonable range of multiplication factors where the operators produce natural looking results. We test a few multiplication factors and run a study to find the threshold between natural and unnatural.

Given a pair of an input image I and a band-sifting operator F, we seek to sample a few versions of the operator, $F(I, m_1), F(I, m_2), ..., F(I, m_s)$, acting on the original image with different multiplication factors $m_1, m_2, \ldots, m_s$. Our early experiments showed that using the same m factors across operators perform poorly; the same value can produce a strong effect with one operator and a

weak one with another. Instead, we define the factor as $m_0 = 1$ and $m_{i+1} = \arg\min_m \|F(I, m_i) - F(I, m)\| > 1$ using the CIE-Lab L_2 norm. We use binary search to efficiently find m_{i+1}. This procedure generates samples regularly spaced in the CIE Lab color space akin to Ngan et al. [113], which approximates a perceptually uniform sampling. We observed that for the increase effects 5 iterations of the above procedure were usually enough to produce too strong results. For the decrease effect, 2 iterations reduced the coefficients close to 0.

With the above sampling procedure, we produced 5 images of different strength for each of the 9 operators that increase the coefficients plus 2 images for the 6 operators that decrease the coefficients, for a total of $5 \times 9 + 2 \times 6 = 57$ variations per image. We used 21 images, 11 faces of various genders and races and 10 non-face objects with uniform materials, e.g., metal, leather, ceramic, and fruits. Users were shown a single modified image at a time and were asked whether it looked natural to them. We provide a snapshot of this task in [22], as well as the full set of images in [21].

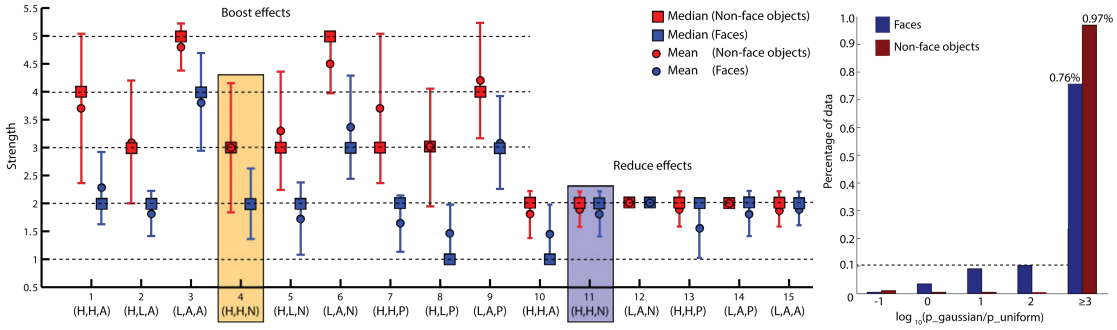
Every user was shown 15 sets of 21 images. Each set was made of each of the 21 test images modified using a randomly picked setting. Users saw the same scene 15 times. Occurrences of any scene were separated by 20 other images to limit the effect of users' getting trained by the previous viewing of that scene. Users were asked to base their decision only on the current image, and they had no reference original image. This study had a total of 47 participants and on average we got 7.5 votes per setting, since we assigned them uniformly across participants. Figure 6.6 summarizes the results of the study and confirms our initial observation that our band-sifting operators can produce nontrivial natural looking variations even for familiar objects, such as human faces. We

also show separate statistics for face and non-face objects, which reveals some interesting trends. For example, in essentially all cases where we boost the signal, images of non-face objects withstand larger modifications than images of face objects. Another interesting observation is about reducing the high-frequency high-amplitude coefficients on faces. If we manipulate the positive or negative coefficients separately, we produce larger modifications compared to reducing them both at the same time. This happens because modifying the positive and negative coefficients at the same time leads to dampening of all high-frequency features, such as dark skin pores and bright skin gloss, which quickly produces unrealisitc smoothening of the skin, compared to modifying skin pores independent of skin gloss. In [22], we show an example image that demonstrates this effect.

Figure 6.7 shows examples of natural and unnatural adjustments using our band-sifting operator that manipulates the high-frequency high-amplitude negative coefficients. We also mark the data points in Figure 6.6a that correspond to this operator. For example, as we pass beyond the realistic threshold of the boost operator, e.g., 3 steps, it starts to produce unrealisitc looking images as we show in Figure 6.7d.

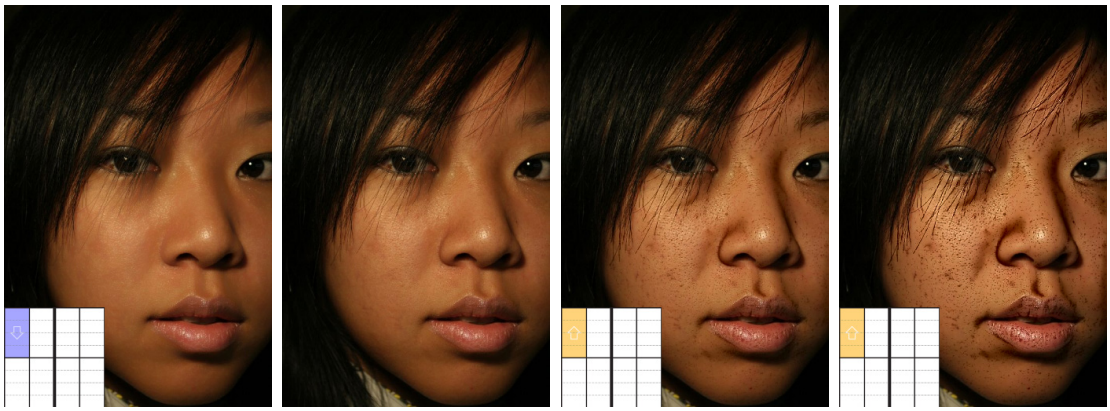
Statistical Significance. To confirm the statistical significance of our results, we assumed that each user has a naturalness threshold. We compared two hypotheses: purely random thresholds, i.e., uniformly distributed over the tested range, and Gaussian distributed thresholds centered on the value that we reported in Figure 6.6a (we used a unit variance for simplicity). We compared the probabilities of obtaining the users' answers under these two assumptions. As shown in Figure 6.6b, for 76% of the face image and 97% of the non-face

images, the results of our study are more than 300× more probable under the Gaussian hypothesis than under the uniform one, which confirms the hypothesis of a consistent threshold across images and users.



(a) Summary statistics (b) Significance

Figure 6.6: *Natural-vs-unnatural study quantitative results*. The plot on the left reports the mean and standard deviations of the votes on the natural-vs-unnatural study (a). We use the notation introduced in Section 6.2.4 on the horizontal axis. We also indicate the median vote that we use later in the second user study. These votes confirm that the threshold between natural and unnatural settings is statistically significant since the uniform-distribution hypothesis is much less probable than the Gaussian-distribution hypothesis in most cases (b). See the text for details.



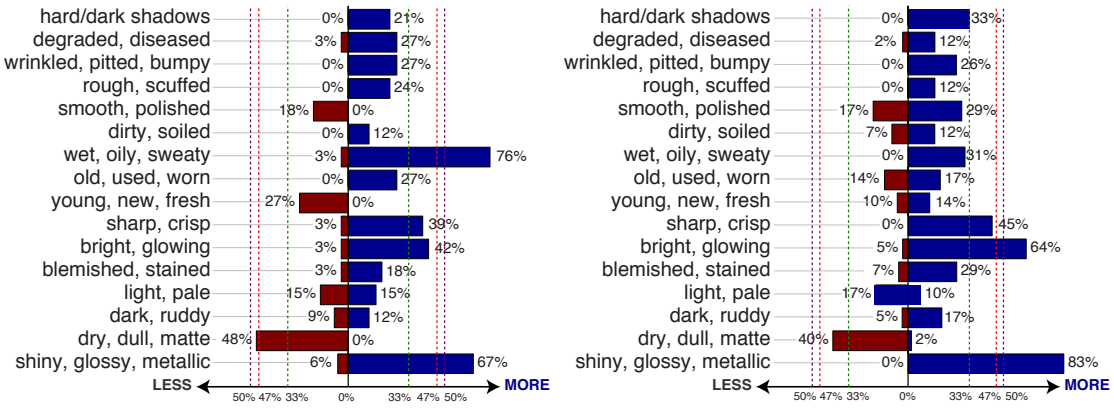
(a) Reduced (b) Original (c) Boosted natural (d) Boosted unnatural

Figure 6.7: *Natural-vs-unnatural qualitative results*. Our first user study characterized how much we can reduce (a) or boost (c) a set of coefficients while maintaining a natural look. Increasing the coefficients past this point eventually generates unnatural images (d). The insets explain which coefficients are affected, see Table 6.1.

6.4.2 Study 2: Name the Effects

The goal of the second study is to determine the perceptual effects associated with the band-sifting operators and to evaluate their consistency across different users and different scenes. Users were shown pairs of images where image A was the original input image I, and image B was a modified version, $F(I, m)$. We seek a parameter value m that produces a visible and natural effect, which we achieved with the quasi-median of the votes in the first study, i.e., the multiplication factor with an equal number of natural and unnatural votes above and below. We showed users the 16 groups of keywords, and for each group, asked them to choose between 3 options for the direction of the perceptual change: "Less", "More" or "N/A". 20 users participated in this study and half of them had not

taken part in Study 1. On average, we got 60 responses per operator (30 for faces and 30 for non-face objects) for a total of 1100 responses. Figure 6.8 summarizes the responses for one of our operators for both face and non-face objects. We show similar plots for all the 15 operators in [22].



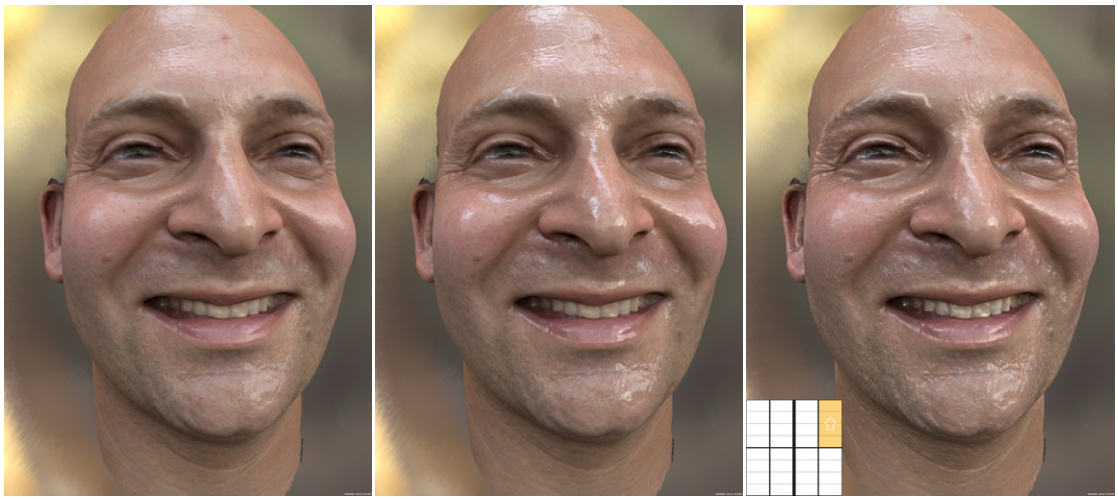
(a) Faces (b) Non-face objects

Figure 6.8: *Name-the-effects quantitative results*. Our second user study determined the perceptual effects associated with our band-sifting operators. In these plots we show results for one of our more consistent effects: boost high amplitude positive-valued high-spatial frequencies. The red bars show the percentage of votes for the "Less" option, the blue bars show the percentage of votes for the "More" option, and the difference to 100%, which we do not show on the plot, is the percentage of votes for the "N/A" option. The majority of participants agreed that this band-sifting operator tends to make human faces more wet, oily or sweaty, whereas, for objects the band-sifting operator tends to make them look more shiny, glossy or metallic. The green lines indicate the probability of chance, i.e., where the results have been generated by picking between the three options, {"Less", "More" or "N/A"}, uniformly at random. The purple lines indicate the 95% confidence interval, i.e. results above this threshold, $\approx 50\%$, are statistically significant with high probability. The red lines, at $\approx 47\%$, indicate the 90% confidence interval. See the text for details about the test of significance.

Statistical Significance. We tested our results against the null hypothesis that the choice between the 3 options is uniformly random. This hypothesis corresponds to a standard multinomial distribution with a 33% mean. For 30 votes, the standard deviation is 8.6%, and using a 95% confidence interval, we can rule out the null hypothesis for results below 16% and above 50%. For a 90% confidence, the interval is (19%, 47%). We give the detailed derivation of these numbers in [22]. We show the 50% and 47% thresholds in Figure 6.8 and use them to report the results in Table 6.1.

Consistent Effects. We found 7 operators that produce consistent and perceptually discriminative effects: boost/reduce shininess, boost/reduce roughness, boost weathering patterns and boost/reduce glow. Table 6.1 summarizes this finding. The number of word sets above the significance threshold ranges from 1 to 4. In general, operators have a more consistent effect on faces. We hypothesize that the diversity of scenes and materials present in the non-face images makes it "harder" for an operator to be consistent. In comparison, the only material in face images is skin and although human observers recognize subtle differences, these are not as important as those between bronze and potatoes for instance.

From a photo editing perspective, these band-sifting operators cover several common tasks on objects such as reducing or increasing weathering, smoothness, and shininess. For faces, they provide a simple and effective means for attenuating blemishes and wrinkles, controlling the dryness of the skin, and adding a photographic glow typically observed in studio portraits.



(a) Input CG image (b) Shinier via 3D model and rendering (c) Shinier via 2D band-sifting operator

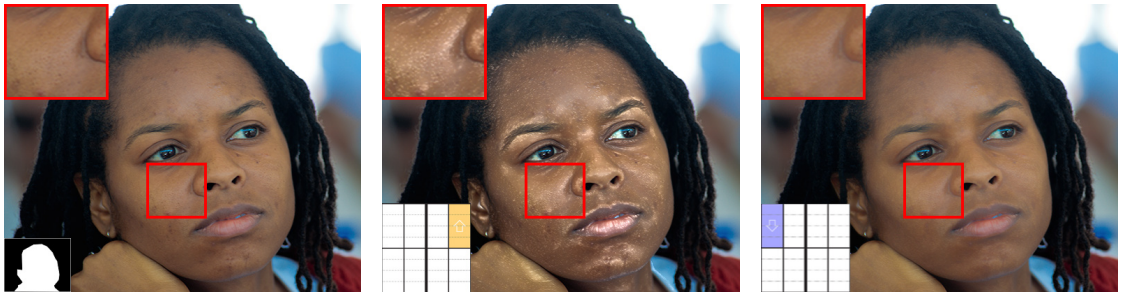
Figure 6.9: *Evaluation on a CG face scene*. Case study on a photorealistic scanned CG model of a face, courtesy of [147]. In (a) and (b) we rendered the face under natural lighting conditions [44], using the isotropic Ward BRDF model with two different values of the parameter α, that controls the spread of the specular lobe. In (c) we show that our band-sifting "wet/oily/shiny" operator, when applied to image (a), can produce a perceptually similar change in shininess.

6.4.3 Image Results

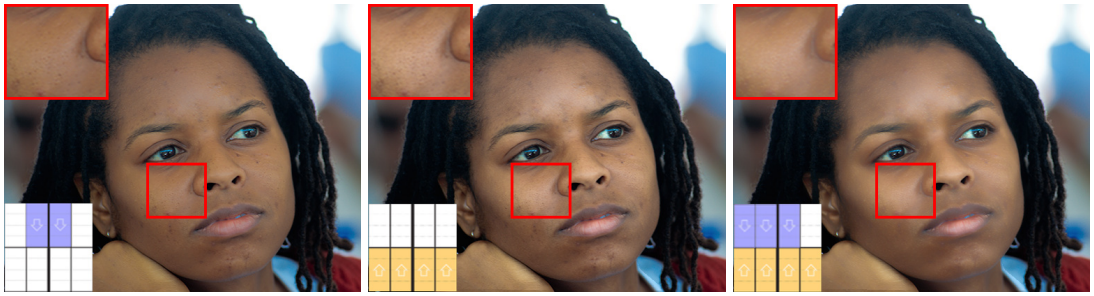
We now demonstrate the effects of band sifting on a range of static scenes and qualitatively discuss the results.

In Figure 6.9, we show that our purely image-based band-sifting operators can produce results visually similar to what can be achieved with a 3D model rendered with a physically inspired BRDF model. We used a photorealisitc scanned 3D model of a face, where we control perceptual parameters related to shininess

by changing physical parameters of the underlying rendering model [149]. We rendered the face with two different values of the α parameter that controls the spread of the specular lobe. Smaller values of α increase the sharpness of the reflected image and make the object look shinier. The question is whether we can get a similar effect using just 2D image manipulations. We show that



(a) Original image (b) Unsatisfactory effect (c) Fewer skin defects/pores



(d) Fewer blemishes (e) More skin glow (f) Combined effect

Figure 6.10: *Showing combinations of band-sifting operators*. Starting from an input image of a very dry and matte face (a), our "wet/oily/shiny" band-sifting operator fails to produce a plausible looking effect (b). Even though one operator may fail to produce a satisfactory effect on some image, others might still work. In (c) we reduce skin defects and pores. In (d) we reduce skin blemishes and pigmentation. In (e) we add a smooth skin glow. Furthermore, simple combinations of band-sifting operators can be used to achieve advanced material editing tasks. In (d) we show the combined effect, which achieves a younger look with a nice skin glow, an advanced effect often seen in professional magazines.

our band-sifting operator that boosts the positive high-amplitude high spatial frequencies produces a perceptually similar change in shininess.

In Figure 6.10b we show a failure case of one of our band-sifting operators. When the visual cues are not presented in the original input image or they are not well isolated by our *sifting* criteria, our band-sifting operators fail to convey consistent perceptual effects (b). Even though one operator may not work well on an image, we found that usually others might still be useful. For example, we can reduce skin defects (c) and blemishes (d) or add a smooth skin glow (e). Furthermore, the independent band-sifting operators that we studied can be combined to achieve even more advanced material editing effects. In (f) we combine the previous three operators to achieve the combined perceptual effect, a younger looking face with a nice skin glow, which is often seen in professional magazines.

Figure 6.12 illustrates the diversity of effects that can be achieved with band-sifting operators. For brevity's sake, we use the notation previously introduced where the amplitude is selected in {H, L, A}, the frequency in {H, L, A}, and the sign in {P, N, A}.

Gargoyle. Boosting the HHP coefficients enhances the gloss, and also brings out whitish "distress" marks, which gives an overall shinier look. Boosting the AHN coefficients produces a patina with dark mottling.

Grapes. We show a combination of two of operators: boosting the ALP coefficients while reducing the AHN coefficients gives the grapes a luminous glow. We also achieve a weathering effect by boosting the LHN coefficients to bring out the patterning on the grape skins.

Onion. Reducing the HHA coefficients removes texture details, while retaining the smooth shiny appearance of the onion; whereas boosting the same coefficients reveals the mottled coloration of the onion skin.

Sweet potatoes. Boosting the LHN coefficients reveals dark blotchy patches while boosting the HHN ones reveals sharp dark spots.

Orange. Boosting the HHP coefficients emphasizes the highlights and makes the orange look shinier. Alternatively, we can emphasize pores and dark spots by boosting the HHN coefficients.

6.4.4 Video Results

We now demonstrate the effects of the band-sifting operators on video sequences. Editing videos consistently is particularly challenging and typically requires many hours of painstaking manual editing. In Figures 6.13 and 6.14, we show example input frames from the video sequences and the results on two different frames. We show results on three video categories where post-process material editing would be a desired tool: (1) closeup views of people giving interviews; (2) 360 spins for product photography; (3) static objects under dynamic lighting. In each case, we applied our operators frame by frame. As can be seen in [24], the results are artifact-free and temporally consistent, which demonstrate the robustness and stability of our band-sifting operators.

Interview A. First, we boost the HHP coefficients to give the skin a more wet/oily look. Then we demonstrate our skin glow effect, which is a common appearance professional photographers aim to achieve through a combination of lighting

and facial cosmetics. We achieve that effect through entirely image-based manipulations, by sifting and then boosting the ALP coefficients. Finally, we emphasize blemishes and spots by boosting the LHA coefficients. We did not use a detailed mask around the face in this case to show that the operators could be directly applied and used in settings like this one.

Interview B. We demonstrate the perceptual consistency of our wet/oily and skin glow effects used in the previous example by applying them onto a different subject. Then, we reduce blemishes and spots to produce a cleaner looking face. To localize the effects on the face only, we created and tracked a detailed facial mask, using the Roto Brush tool in Premiere Pro.

Leather shoes. Boosting the HHA coefficients emphasizes the highlights and some scratches, which gives an overall shinier looking leather. Then, we give the leather a smoother and more polished look by boosting the ALP coefficients. Finally, we achieve a weathering effect by boosting the LHA to bring out the patterning on the leather.

Grapes. Boosting the HHP coefficients emphasizes the highlights and makes the grapes look shinier and wetter. Then, we reduce the same coefficients to produce a more diffuse look. Finally, we bring out the weathering patterns, which also make the grapes look a bit more dirty.

Helmet. We demonstrate that the operators produce a consistent look under dynamic lighting conditions. Boosting the HHP coefficients emphasizes the highlights and some scratches, which makes the helmet appear shinier. Boosting the ALP brings out the broad gloss, which gives a smoother, less rough looking metal. Finally, boosting the LHA coefficients brings out the weathering patterns,

which makes the metal look like it has more patina.

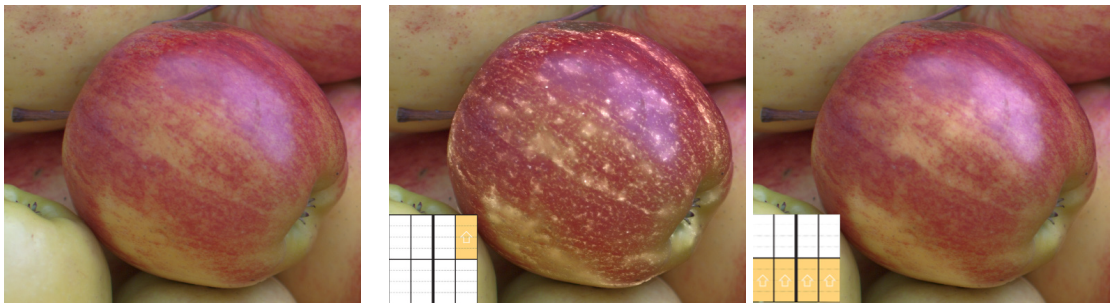
6.4.5 Discussion and limitations

Our purpose in this paper is to devise 2D image operators, which can make visually plausible modifications of surface properties. Such operators are simple to implement and can be applied to arbitrary images. However, they are only useful for properties that are manifested simply within the distributions of subband coefficients. Prior research indicates that such descriptions are useful for certain tasks involving natural images (e.g., in denoising [45; 136], and in texture analysis [74; 120]). Here we have tested the utility of similar representations in modifying material appearance.

We have identified several kinds of material appearance that are commonly associated with certain subband properties, and that can often be manipulated. Specularities from fine-scale features show up in positive-valued, high-amplitude, high-spatial frequencies. Boosting them leads to an enhanced "glistening" appearance, which may be interpreted as oiliness or wetness on skin. Specularities from smooth, large-scale features show up in positive-valued, medium-spatial frequency coefficients. Boosting them leads to a smooth shine or sense of skin glow. Small spots, pits, and wrinkles, typically manifest themselves as small dark features that show up in the high-amplitude negative coefficients of high-spatial frequencies. Boosting them often emphasizes the visibility of fine-scale texture (both fine-scale geometry and fine-scale albedo). These features are often associated with the aging of human skin, or the weathering of natural surfaces. In addition, the low-amplitude coefficients of the high-spatial

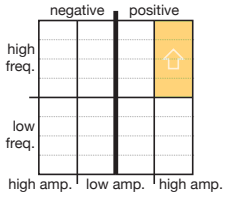
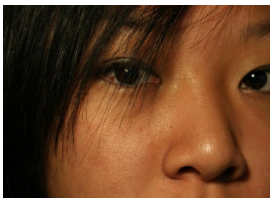
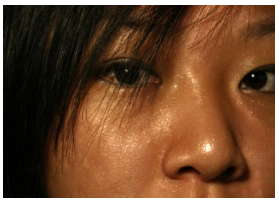
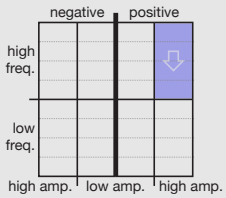
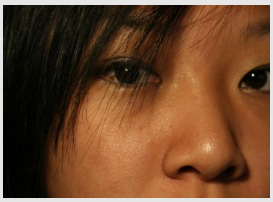
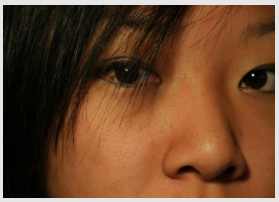
frequencies (both negative and positive) are often associated with splotchy or mottled pigmentation. These can also enhance the sense of age, weathering, and discoloration. In the past, those coefficients were typically artifacts of the imaging chain, i.e., sensor noise or image coding artifacts. However, modern digital cameras offer clean images, and the low amplitude signals tell us about the scene, not about the camera.

We find it remarkable that these effects tend to look natural and realistic, rather than being the result of some artificial manipulation. The realism presumably results from the fact that the band-sifting operators are not inventing any information that is not already there; they are just emphasizing or de-emphasizing visual patterns that are already part of the image. As we have discussed earlier, when the visual cues are not presented in the original input image, or they are not well isolated by our *sifting* criteria, our band-sifting operators may fail to convey consistent perceptual effects. High frequency albedo can also lead to unsatisfactory effects for band-sifting operators that manipulate those frequencies. In Figure 6.11, starting from an input image that has high frequency texture (a), our band-sifting operator fails to produce a satisfactory "shiny" effect (b). Although one operator may fail on a certain image, other band-sifting operators might still be useful. In (c) we manipulate material properties related to smoothness, by sifting and then boosting the low-spatial frequency coefficients, which gives the apple a more polished look.



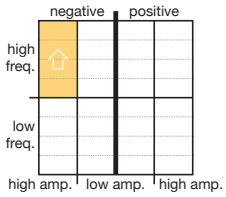
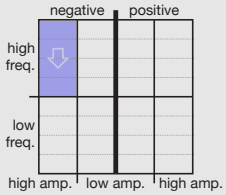
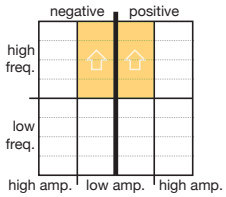
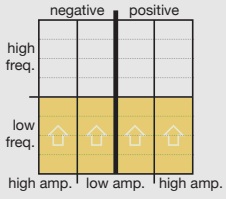
(a) Original image (b) Unsatisfactory effect (c) More polished

Figure 6.11: *Failure case on object with high-frequency albedo.* Working on an object that has high frequency albedo (a), our "shiny/glossy/metallic" band-sifting operator is less effective in conveying the more shiny look since it picks mainly on the albedo (b). Although one band-sifting operator may fail, others might still be useful. In (c) we manipulate material properties related to smoothness, which makes the apple look more polished.

	coefficient selection	input	output	associated properties
1.		**Objects** (strength: 4) moreshiny/glossy/metallic (83%) morebright/glowing (64%) **Faces** (strength: 2) morewet/oily/sweaty (76%) moreshiny/glossy/metallic (67%)		
2.		**Objects** (strength: 2) moredry/dull/matte (62%) less shiny/glossy/metallic (54%) **Faces** (strength: 2) moredry/dull/matte (62%) less shiny/glossy/metallic (55%)		

(to be continued on next page)

(continued from previous page)

	coefficient selection	input	output	associated properties
3.		**Objects** (strength: 3.5) moresharp/crisp (48%) **Faces** (strength: 2) morehard/dark shadows (60%)		
4.		**Objects** (strength: 2) **Faces** (strength: 2) moresmooth/polished (56%) less wrinkled/pitted/bumpy (53%) less hard/dark shadows (53%)		
5.		**Objects** (strength: 3.5) moresharp/crisp (57%) moreold/used/worn (49%) **Faces** (strength: 2) moreblemished/stained (62%) moreold/used/worn (55%) morewrinkled/pitted/bumpy (55%)		
6.		**Objects** (strength: 5) moreshiny/glossy/metallic (79%) morebright/glowing (56%) **Faces** (strength: 4) morebright/glowing (70%) moreshiny/glossy/metallic (68%)		

(to be continued on next page)

(continued from previous page)

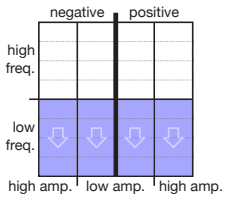
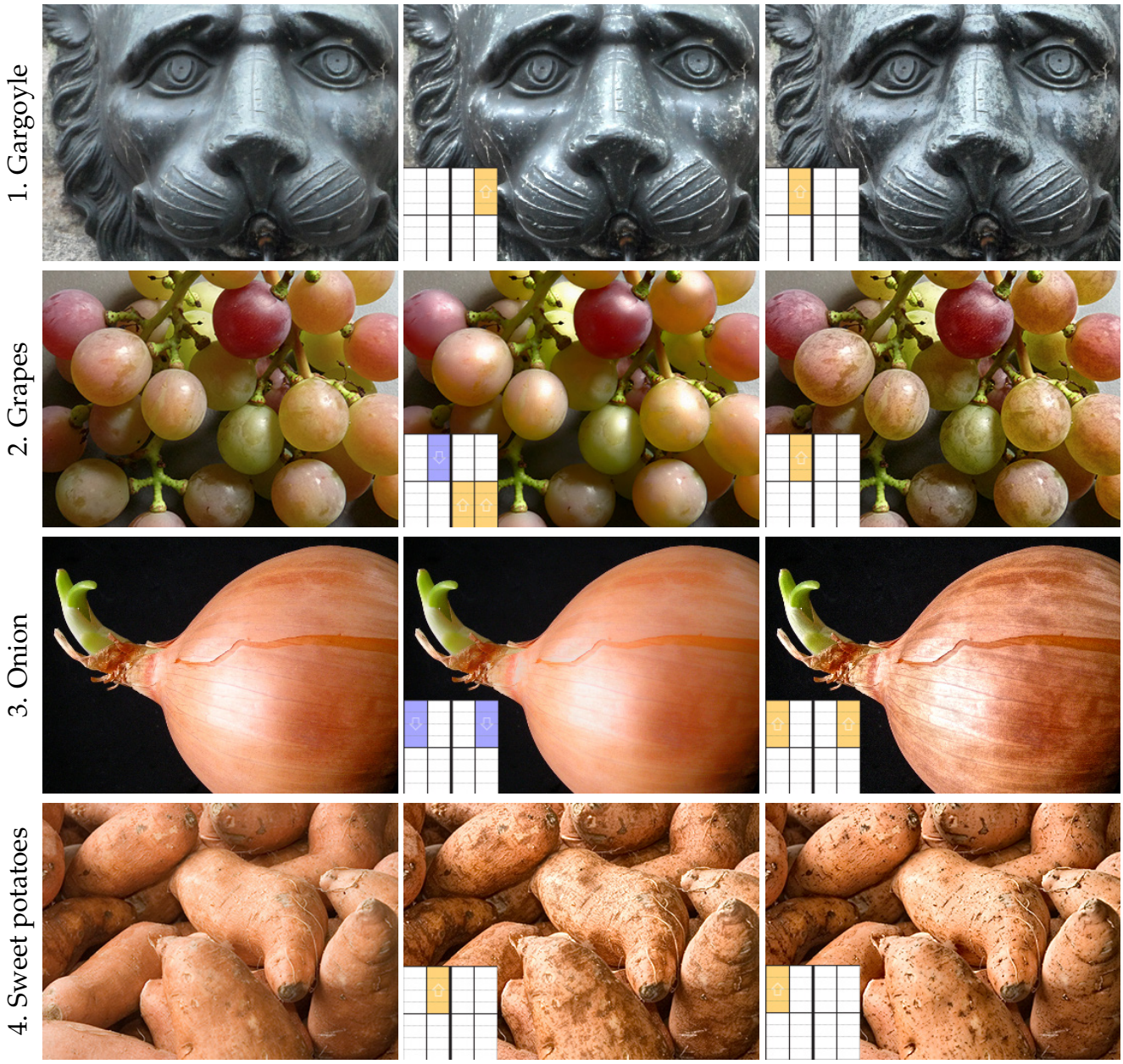
coefficient selection	input	output	associated properties
7. 	**Objects** (strength: 2) moredry/dull/matte (58%) less shiny/glossy/metallic (52%) **Faces** (strength: 2) moredry/dull/matte (64%) less shiny/glossy/metallic (57%) less bright/glowing (57%) less sharp/crisp (54%)		

Table 6.1: Recap of our most effective band-sifting operators.

6.5 Conclusions

We present band-sifting operators, and demonstrate their use in manipulating surface appearance. The band-sifting operators selectively alter coefficients within a subband decomposition, where the selection is based on spatial scale, sign, and amplitude. We explored a reasonable subspace of such operators and demonstrated their ability to modify a variety of surface properties in natural scenes. We use only 2D operations, but they can give the visual impression of acting on the materials in 3D scenes. We found some operators that were useful in controlling smoothness or gloss, which could alter the appearance of wetness, shininess, or degree of polish. Other operators altered the apparent



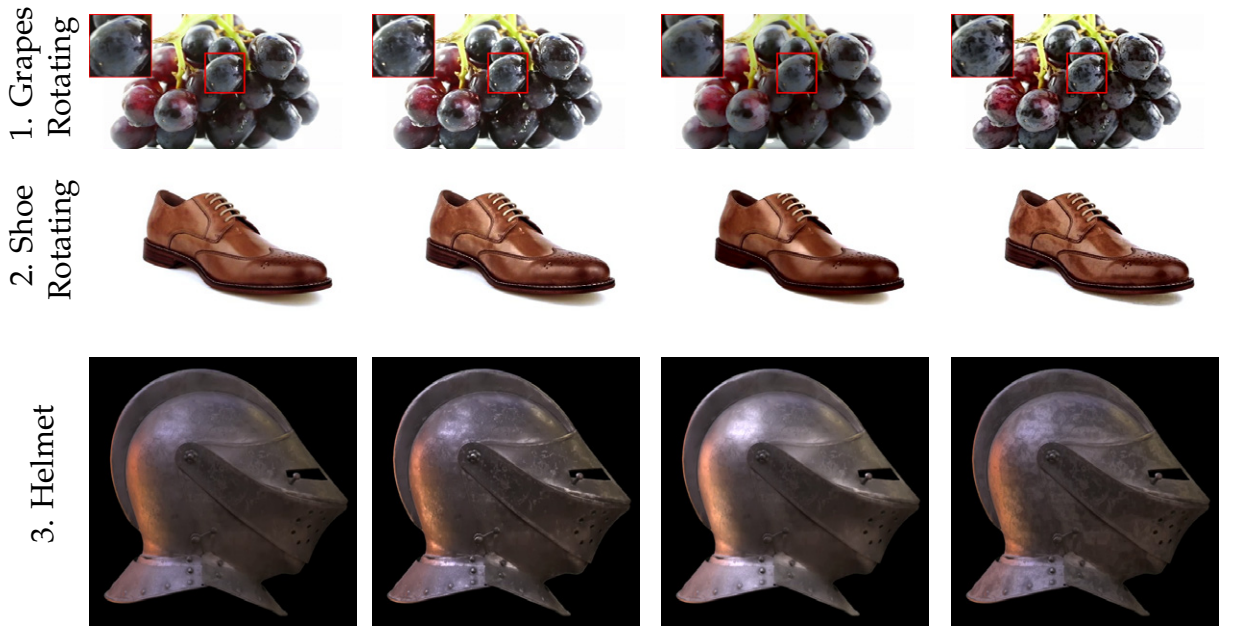
(a) Original (b) Filtered result #1 (c) Filtered result #2

Figure 6.12: *Showing a variety of effects produced with our band-sifting operators* . In row (1) we make the gargoyle look more glossy (1,b) or more weathered, by emphasizing the patina (1,c). The grapes can be given a more glowing (2,b) or more dirty look (2,c) by emphasizing the patterning on the skin. The skin of the onion can be given a more fresh (3,b) or a more worn-out look (3,c). In row (4) we show that by treating the low and high amplitude coefficients separately we can get very different perceptual effects, weathering patterns (4,b) vs surface roughness (4,c).

(a) Original (b) Filtered result #1 (c) Filtered result #2 (c) Filtered result #3

Figure 6.13: *Results on videos of faces, downloaded from the Internet*. Our band-sifting operators can be used to efficiently post-process material appearance in videos without introducing temporal artifacts, see [24]. For example, we can make the actor's skin look more oily, column 2, or add more skin glow, column 3. In column 4 we control wrinkles and blemishes by manipulating coefficients in the corresponding combination of band-sifting paths. We boost the coefficients in rows 1 and 2 which gives the face a more aged look, whereas in rows 3 and 4 we reduce them to render a more clean and young looking face. The simplicity of our model allows all this to be done interactively by manipulating a few sliders, without having to model the effects pixel-by-pixel on every frame.

pigmentation, roughness, or weathering of surfaces. We performed user studies and determined that there are certain operators that lead to consistent perceptual effects across various images and across multiple observers. Image class does matter: with images of faces, subjects reported that that the filters would change face-specific properties such as oiliness, blemishes, wrinkles, skin age, and skin health. Given the importance of perceptual surface qualities, we expect that these



(a) Original (b) Filtered result #1 (c) Filtered result #2 (d) Filtered result #3

Figure 6.14: *Evaluation of our band-sifting operators for product photography videos*. In column (a) we show example input frames from the video sequences, and the next columns show some of the effects our band-sifting operators can achieve. In column (b) we demonstrate that our "shiny/glossy/metallic" effect, which boosts the high amplitude positive-valued high-spatial frequencies, produces a perceptually consistent effect across different scenes. In column (c), rows 1 we reduce the corresponding coefficients, which gives the grapes a more diffuse look. In the rest of column (c), rows 2 through 3, we show our "bright/glowing" effect, which boosts the positive-valued low-spatial frequency coefficients. This gives the leather shoes a more smooth and polished appearance, row 2, and it makes the metal helmet look more smooth and less rough, row 3. The helmet scene is courtesy of [43]. In column (d) we show our "old/used/worn" effect, which boosts the low amplitude high-spatial frequencies. In agreement with our user study #2, this effect produces a persistent perceptual effect of aging or damage by bringing out weathering patterns such as spots and dust on fruits, row 1, leather stains, rows 2, and patina on metals, rows 3.

band-sifting operators can offer an important tool for photography. Our band-sifting operators can also be used with video sequences. The visual effects tend to be consistent across a sequence, making it possible, for example, to change the apparent shininess of an actor's skin. In the future, further exploration of band-sifting, e.g., by a finer subdivision of our proposed space or by introducing new sifting criteria, could open the door for many more operators for image-based material editing. Identifying the conditions when an operator would succeed or fail to produce a desired effect, based on the content of an arbitrary given image, is an interesting avenue for future work. One way of addressing this problem is to use a data-driven approach, where an object could be photographed or rendered with different material properties. Machine learning could then be used to explore a variety of per-pixel features that relate the changes of the physical parametes with the observed changes in the image domain. This could potentially give us a way to fine-tune the band-sifting operators specifically for each pixel in a given image.

CHAPTER 7

CONCLUSION

Understanding the intricate relation between lighting and surface properties is one of the key components in photography. Providing computational tools to assist both lighting design and image-based material editing can benefit not only the rapidly increasing number of casual photographers but also other industrial applications including online marketing and entertainment.

This thesis has brought a new set of computational methods for advanced photographic tasks, such as lighting design, white balance under mixed lighting and image-based material editing. Our contributions include:

- A novel, user-assisted computational method to lighting design, based on a set of *basis lights* inspired by common goals in photography (Chapter 3). This approach allows both novice and professional photographers to explore a non-trivial set of sophisticated lighting designs in a much more efficient way than what could be done using traditional methods. In a follow-up project, Chapter 4, we extended our work for product photography and videography, while further simplifying the required equipment.
- A new, user-guided approach to the ill-posed white balance problem under mixed lighting conditions (Chapter 5). We have demonstrated the ability of our approach to handle hard practical cases, such as a mixture of indoor and outdoor lighting conditions.
- We have presented band-sifting operators, a new space based on multi-scale image analysis (Chapter 6). We have demonstrate their use in manipulating surface appearance related to shininess, smoothness and weathering in both images and videos.

7.1 Future research directions

Automating various aspects of photography is more important than ever. The digital camera revolution has led to a widespread availability of recording devices, putting them in the hands of millions of casual photographers. This has led to an increasing demand of more effective tools to assist the process of visual communication. In this dissertation, we have taken one step forward by introducing new approaches to assist advanced photographic tasks such as lighting design and surface appearance. We believe that these techniques can inspire future research that has the potential to significantly benefit variety of applications in fields such as consumer photography, online marketing and entertainment.

Computational Lighting and Image-Based Material Design. We have demonstrated the success of our approaches for static scenes in both architectural and commercial product photography. One of our main objectives was to use a simple equipment that is easily available to novice and enthusiast photographers. In the future, further automation of the acquisition process could be achieved with the use of personal lighting drones, as those are becoming widely available. A drone can quickly sample a variety of lighting configurations, which is a rather mundane and mechanical task. Then, our algorithms can be used to analyze the captured data and extract basic building blocks, based on common photography goals. Finally, the user can explore a variety of lighting designs based on our basis building blocks. A future system can use our basis elements to automatically build a variety of final solutions, based on expert designed rules or possibly learn those rules from data. That way the user would only be responsible to browse and pick the solution that he or she likes most. Further, a machine learning

approach could potentially be used to learn a combination of basis elements that form final compositions in professionally lit images.

In the future, machine learning could also be used to learn image-based material editing operators using a set of before and after images with different material properties. For example, we can use the physically based infrastructure from computer graphics to render a variety of objects with different geometry, pose, scale and lighting. A material property of interest, such as surface roughness or translucency, can then be varied to create before and after pairs of images that could serve as a training set. Real world data could also be collected for a variety of phenomena, such as weathering effects due to atmosphere or lighting exposure.

In the future, a similar approach could also be used to learn a single image-based lighting design. For example, before and after images could be rendered, or captured in real life, where the first image would be taken under a typical everyday lighting setup, and the second image would be taken under a professionally placed lighting equipment. Machine learning could potentially be used to learn a set of 2D image operators that predict the difference that transforms a casually lit image to a professionally lit one. This could also make image-based lighting design applicable to dynamic scenes, where the learned rules can be applied per-frame, with a possible regularization term to enforce temporal smoothness.

APPENDIX A

APPENDIX FOR CHAPTER 5

In this appendix we give a detailed derivation of the local duochromatic model under multi-lights conditions. This shows that for a number of useful specific cases, the null space of our energy based on the Matting Laplacian, contains the solutions to those configurations.

Duochromatic scenes under 2D lighting. We show that the W factors of scenes made of two reflectance values R_1 and R_2 lit by a combination of two lights L_1 and L_2 can be expressed as an affine combination of the input chromaticities. We also assume that the observed intensities are not affected by the lights, only the chromaticities vary. That is, $\sum L_1 R_1 = \sum L_2 R_1$ and $\sum L_1 R_2 = \sum L_2 R_2$. We name these two quantities i_1 and i_2. We show that the W_r factor is an affine combination of the $(C^I_\mathrm{r}, C^I_\mathrm{g}, C^I_\mathrm{b})$ triplet; similar results can be derived for W_g and W_b. From Equation 5.1, we have:

$$I_\mathrm{r} = \begin{cases} (\lambda_1 L_{1\mathrm{r}} + \lambda_2 L_{2\mathrm{r}}) R_{1\mathrm{r}} & \text{if in } R_1 \text{ region} \\ (\lambda_1 L_{1\mathrm{r}} + \lambda_2 L_{2\mathrm{r}}) R_{2\mathrm{r}} & \text{if in } R_2 \text{ region} \end{cases} \tag{A.1}$$

where λ_1 and λ_2 are the spatially varying intensities of the lights L_1 and L_2. We seek affine coefficients $(a_\mathrm{r}, a_\mathrm{g}, a_\mathrm{b})$ and b independent of λ_1 and λ_2 such that $W_\mathrm{r} = \sum_{c\in\{\mathrm{r,g,b}\}} a_c C^I_c + b$. We first derive useful relationships in the R_1 region, similar formulas can be obtained with R_2. We divide Equation A.1 by $\sum I$, and name $\alpha = \lambda_1/(\lambda_1 + \lambda_2)$ and $k_1 = \sum_c R_{1c} \,/\, i_1$ to get:

$$C^I_\mathrm{r} = \frac{(\lambda_1 L_{1\mathrm{r}} + \lambda_2 L_{2\mathrm{r}}) R_{1\mathrm{r}}}{\sum_c (\lambda_1 L_{1c} + \lambda_2 L_{2c}) R_{1c}} = \frac{(\lambda_1 L_{1\mathrm{r}} + \lambda_2 L_{2\mathrm{r}}) R_{1\mathrm{r}}}{(\lambda_1 + \lambda_2) i_1} \tag{A.2a}$$

$$= (\alpha L_{1\mathrm{r}} + (1-\alpha) L_{2\mathrm{r}}) \frac{\sum_c R_{1c}}{i_1} \frac{R_{1\mathrm{r}}}{\sum_c R_{1c}} \tag{A.2b}$$

$$= k_1 \left(\alpha L_{1\mathrm{r}} + (1-\alpha) L_{2\mathrm{r}}\right) C^{R_1}_\mathrm{r} \tag{A.2c}$$

Using Equation 5.4, we have $W_\mathrm{r} = k_1\,(\alpha L_{1\mathrm{r}} + (1-\alpha)L_{2\mathrm{r}})$. We first show the result for $\alpha = 0$ and $\alpha = 1$ and use superposition to extend it to other α values. That is, we seek $(a_\mathrm{r}, a_\mathrm{g}, a_\mathrm{b})$ and b such that $W_\mathrm{r} = \sum_{c\in\{\mathrm{r,g,b}\}} a_c C_c^I + b$ for $\alpha = 0$ and $\alpha = 1$ in both R_1 regions and R_2 regions. Using a matrix formulation, this gives:

$$\begin{pmatrix} k_1\,L_{1\mathrm{r}}\,C_\mathrm{r}^{R_1} & k_1\,L_{1\mathrm{g}}\,C_\mathrm{g}^{R_1} & k_1\,L_{1\mathrm{b}}\,C_\mathrm{b}^{R_1} & 1 \\ k_1\,L_{2\mathrm{r}}\,C_\mathrm{r}^{R_1} & k_1\,L_{2\mathrm{g}}\,C_\mathrm{g}^{R_1} & k_1\,L_{2\mathrm{b}}\,C_\mathrm{b}^{R_1} & 1 \\ k_2\,L_{1\mathrm{r}}\,C_\mathrm{r}^{R_2} & k_2\,L_{1\mathrm{g}}\,C_\mathrm{g}^{R_2} & k_2\,L_{1\mathrm{b}}\,C_\mathrm{b}^{R_2} & 1 \\ k_2\,L_{2\mathrm{r}}\,C_\mathrm{r}^{R_2} & k_2\,L_{2\mathrm{g}}\,C_\mathrm{g}^{R_2} & k_2\,L_{2\mathrm{b}}\,C_\mathrm{b}^{R_2} & 1 \end{pmatrix} \begin{pmatrix} a_\mathrm{r} \\ a_\mathrm{g} \\ a_\mathrm{b} \\ b \end{pmatrix} = \begin{pmatrix} k_1 L_{1\mathrm{r}} \\ k_1 L_{2\mathrm{r}} \\ k_2 L_{1\mathrm{r}} \\ k_2 L_{2\mathrm{r}} \end{pmatrix} \tag{A.3}$$

Since the matrix is square, the system is either well-posed or under-constrained, which guarantees that there is at least one solution. Further, because the values of C^I and W for an arbitrary α are a linear interpolation of the values at $\alpha = 0$ and $\alpha = 1$, this ensures that a solution of Equation A.3 is valid for any α value.

Discussion. The assumption $\sum L_1 R_1 = \sum L_2 R_1$ and $\sum L_1 R_2 = \sum L_2 R_2$ may not always be satisfied. Nevertheless, since we freely scale up and down L_1 and L_2, as long as we apply the inverse scale factors to λ_1 and λ_2, we can use these degrees of freedom to minimize the differences between $\sum L_1 R_1$ and $\sum L_2 R_1$, and $\sum L_1 R_2$ and $\sum L_2 R_2$.

APPENDIX B

APPENDIX FOR CHAPTER 6

In this appendix we provide a pseudo-code implementation of our band-sifting operators. Algorithm 1 shows the core of our method, where all the major steps can be implemented efficiently using a fast summed-area table algorithm. The same is true for our choice of multi-scale decomposition shown in Algorithm 2.

Algorithm 1: BANDSIFTINGOPERATOR

Require: image I, multiplication factor λ, $sign \in \{\mathsf{pos}, \mathsf{neg}, \mathsf{all}\}$,

1: $freq \in \{\mathsf{high}, \mathsf{low}, \mathsf{all}\}$, $amp \in \{\mathsf{high}, \mathsf{low}, \mathsf{all}\}$

Ensure: image O

2: $L \leftarrow \mathrm{luminance}(I)$ // process only the L channel of Lab

3: $\{S_\ell\} \leftarrow \textsc{Decompose}(\log(L+\epsilon))$ // multiscale decomposition of log luminance (Alg. 2), use small ϵ to avoid 0 values

4: $n \leftarrow \#\{S_\ell\}$ // define n, the number of subbands

5: **for all** levels $\ell \in [1; n]$ **do**

6: $R \leftarrow S_\ell$ // keep a copy of the subband as reference

7: **for all** coefficients $c \in S_\ell$ **do**

8: // check if c selected by its sign and frequency (Alg. 3)

9: **if** SIGNANDFREQUENCYSELECTED(c, $sign$, $freq$) **then**

10: // check if c is selected by its amplitude

```
        if amp = all then
          c ← λc // if all amplitudes, directly apply multiplier
        else
          // else smooth transition between high and low amplitudes
          σ = StdDev(S_ℓ)
          α ← SmoothStep(0.8σ, 1.2σ, |c|) // returns 0 if |c| < 0.8σ, 1 if > 1.2σ; transitions in between
          // orient transition depending on amplitude selection
          if amp = high then
            c ← c × (1 + α(λ − 1))
          else if amp = low then
            c ← c × (1 + (1 − α)(λ − 1))
          end if
        end if
      end if
    end for
    // smooth applied gain map
    Gain ← S_ℓ/R
    Gain ← Gain ⊗ G(2^ℓ)
    S_ℓ ← R × Gain
  end for
  O = exp(∑_1^n S_ℓ) − ϵ // sum subbands to get output
```

Algorithm 2: DECOMPOSE

Require: single-channel image C

Ensure: multiscale stack $\{S_\ell\}$

1: $n \leftarrow \log_2(\min(C.width, C.height))$ // number of layers

2: $Tmp_1 \leftarrow \text{GUIDEDFILTER}(C, 0.1^2, 2)$ // first operand is input image, second is range sensitivity, third is spatial extent

3: **for** $\ell = 1 \ldots n$ **do**

4: $\quad Tmp_2 \leftarrow Tmp_1$

5: $\quad Tmp_1 \leftarrow \text{GUIDEDFILTER}(Tmp_2, 0.1^2, 2^\ell)$ // double the spatial extent each time

6: $\quad S_\ell = Tmp_2 - Tmp_1$ // a subband is the difference of two succcessively filtered versions of the image

7: **end for**

Algorithm 3: SIGNANDFREQUENCYSELECTED

Require: $c \in \mathbb{R}$, $sign \in \{\text{pos}, \text{neg}, \text{all}\}$, $freq \in \{\text{high}, \text{low}, \text{all}\}$

Ensure: $isSelected \in \{\text{true}, \text{false}\}$

1: $signSelected \leftarrow sign = \text{all}$

or $(c < 0$ **and** $sign = \text{neg})$

or $(c > 0$ **and** $sign = \text{pos})$

2: $freqSelected \leftarrow freq = \text{all}$

or $(\ell \leq n/2$ **and** $freq = \text{low})$

or $(\ell > n/2$ **and** $freq = \text{high})$

3: $isSelected \leftarrow signSelected$ **and**; $freqSelected$

BIBLIOGRAPHY

[1] Edward H. Adelson and James R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*. MIT Press, 1991.

[2] Aseem Agarwala. *Authoring effective depictions of reality by combining multiple samples of the plenoptic function*. PhD thesis, University of Washington, 2006.

[3] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. *ACM Trans. Graph.*, 2004.

[4] Muhammad Ajmal, Muhammad Husnain Ashraf, Muhammad Shakir, Yasir Abbas, and Faiz Ali Shah. Video summarization: techniques and classification. In *Computer Vision and Graphics*, pages 1–13. Springer, 2012.

[5] David Akers, Frank Losasso, Jeff Klingner, Maneesh Agrawala, John Rick, and Pat Hanrahan. Conveying shape and features with image-based relighting. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*. IEEE Computer Society, 2003.

[6] Xiaobo An and Fabio Pellacini. Appprop:All-pairs appearance-space edit propagation. *ACM Trans. on Graphics*, 27(3), 2008.

[7] ARQBALL. Arqspin 360 photography. `https://arqspin.com/`, 2015.

[8] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. volume 56, pages 411–421, August 2006.

[9] Soonmin Bae, Sylvain Paris, and Frdo Durand. Two-scale tone management for photographic look. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3):637 – 645, 2006.

[10] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[11] T.E. Bishop and P. Favaro. The light field camera: Extended depth of field, aliasing and super-resolution. *IEEE Trans. Pattern. Anal. Mach. Intell.*, 2011.

[12] M. Bleier, C. Riess, S. Beigpour, E. Eibenberger, E. Angelopoulou, T. Tröger, and A. Kaup. Color constancy and non-uniform illumination: Can existing algorithms work? In *IEEE Color and Photometry in Comp. Vision Workshop*, 2011.

[13] Steven Bourke, Kevin McCarthy, and Barry Smyth. The social camera: A case-study in contextual image recommendation. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, IUI '11, pages 13–22, New York, NY, USA, 2011. ACM.

[14] Adrien Bousseau, Emmanuelle Chapoulie, Ravi Ramamoorthi, and Maneesh Agrawala. Optimizing environment maps for material depiction. In *CGF*, EGSR'11. Eurographics Association, 2011.

[15] Adrien Bousseau, Emmanuelle Chapoulie, Ravi Ramamoorthi, and Maneesh Agrawala. Optimizing environment maps for material depiction. In *Computer Graphics Forum (Proc. of the Eurographics Symposium on Rendering)*, volume 30, 2011.

[16] Adrien Bousseau, Sylvain Paris, and Frédo Durand. User-assisted intrinsic images. *ACM Trans. Graph.*, 28(5), December 2009.

[17] Adrien Bousseau, Sylvain Paris, and Frédo Durand. User-assisted intrinsic images. *ACM Trans. on Graphics*, 28(5), 2009.

[18] Ivaylo Boyadzhiev. Supplemental material for our multi-lights white balance work. `http://www.cs.cornell.edu/projects/white_balance/download/supplemental_material.pdf`, 2012.

[19] Ivaylo Boyadzhiev. Supplemental material for our computational lighting design work. `http://www.cs.cornell.edu/projects/light_compositing/download/light_compositing_supplemental.pdf`, 2013.

[20] Ivaylo Boyadzhiev. Supplemental video for our computational lighting design work. `https://vimeo.com/64976852`, 2013.

[21] Ivaylo Boyadzhiev. Supplemental images for our image-based material editing work. `http://www.cs.cornell.edu/projects/band_sifting_filters/download/band_sifting_all_images.zip`, 2015.

[22] Ivaylo Boyadzhiev. Supplemental material for our image-based material editing work. http://www.cs.cornell.edu/projects/band_sifting_filters/download/band_sifting_supplemental.pdf, 2015.

[23] Ivaylo Boyadzhiev. Supplemental video for our dynamic lighting design for product photography and videography. http://www.cs.cornell.edu/projects/light_compositing/download/dynamic_lighting_design_video.mp4, 2015.

[24] Ivaylo Boyadzhiev. Supplemental video for our image-based material editing work. http://www.cs.cornell.edu/projects/band_sifting_filters/download/band_sifting_video.mp4, 2015.

[25] Ivaylo Boyadzhiev, Kavita Bala, Sylvain Paris, and Edward Adelson. Band-sifting decomposition for image based material editing. *ACM Transactions on Graphics*, to appear.

[26] Ivaylo Boyadzhiev, Kavita Bala, Sylvain Paris, and Frédo Durand. User-guided white balance for mixed lighting conditions. *ACM Trans. Graph.*, 31(6), November 2012.

[27] Ivaylo Boyadzhiev, Sylvain Paris, and Kavita Bala. User-assisted image compositing for photographic lighting. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*, 32(4):36, 2013.

[28] Y.Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112 vol.1, 2001.

[29] Peter J. Burt and Edward H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communication*, 31(4):532–540, 1983.

[30] Peter J. Burt and Edward H. Adelson. A multiresolution spline with application to image mosaics. *ACM Trans. Graph.*, 2(4), 1983.

[31] Joao Carreira and et al. Constrained parametric min-cuts for automatic object segmentation, 2010.

[32] Robert Carroll, Ravi Ramamoorthi, and Maneesh Agrawala. Illumina-

tion decomposition for material recoloring with consistent interreflections. *ACM Trans. Graph.*, 30(4), July 2011.

[33] Robert Carroll, Ravi Ramamoorthi, and Maneesh Agrawala. Illumination decomposition for material recoloring with consistent interreflections. *ACM Trans. on Graphics*, 30(3), 2011.

[34] Ayan Chakrabarti, Daniel Scharstein, and Todd Zickler. An empirical camera model for internet color vision, 2009.

[35] Jiawen Chen, Sylvain Paris, and Frédo Durand. Real-time edge-aware image processing with the bilateral grid. *ACM Trans. on Graphics*, 26(3), 2007.

[36] Jiawen Chen, Sylvain Paris, and Frédo Durand. Real-time edge-aware image processing with the bilateral grid. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), 2007.

[37] Q. Chen, D. Li, and C.K. Tang. KNN matting. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[38] Yung cheng Liu, Wen hsin Chan, and Ye quang Chen. Automatic white balance for digital still camera. *IEEE Transactions on Consumer Electronics*, pages 460–466, 1995.

[39] Hamilton Chong, Steven Gortler, and Todd Zickler. The von Kries hypothesis and a basis for color constancy. In *IEEE International Conf. on Computer Vision*, 2007.

[40] Michael F. Cohen, R. Alex Colburn, and Steven Drucker. Image stacks. Technical report, Microsoft Research, 2003. MSR-TR-2003-40.

[41] Marc Davis, Michael Smith, John Canny, Nathan Good, Simon King, and Rajkumar Janakiraman. Towards context-aware face recognition. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, pages 483–486, New York, NY, USA, 2005. ACM.

[42] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of ACM SIGGRAPH 2000*, July 2000.

[43] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley

Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 145–156, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[44] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 369–378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[45] Dave Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 1995.

[46] Frédo Durand and Richard Szeliski. Guest editors' introduction: Computational photography. *IEEE Computer Graphics and Applications*, 27(2):21–22, 2007.

[47] Frdo Durand. The art and science of depiction.

[48] Frdo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 21(3), 2002.

[49] Marc Ebner. Color constancy using local color shifts. In *European Conf. on Computer Vision*, 2004.

[50] Marc Ebner. Color constancy based on local space average color. *Machine Vision and Applications Journal*, 20(5), 2009.

[51] Elmar Eisemann and Frédo Durand. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.*, 23(3), 2004.

[52] O. Elbs. *Neuro-esthetics: mapological foundations and applications (map 2003)*. M Press, 2005.

[53] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 27(3), 2008.

[54] Raanan Fattal. Edge-avoiding wavelets and their applications. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), 2009.

[55] Raanan Fattal, Maneesh Agrawala, and Szymon Rusinkiewicz. Multiscale shape and detail enhancement from multi-light image collections. In *ACM SIGGRAPH 2007 papers*, 2007.

[56] Raanan Fattal, Maneesh Agrawala, and Szymon Rusinkiewicz. Multiscale shape and detail enhancement from multi-light image collections. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), 2007.

[57] G. D. Finlayson, S. D. Hordley, and I. Tastl. Gamut constrained illuminant estimation. *International Journal of Computer Vision*, 67(1), 2006.

[58] J. Fiss, B. Curless, and R. Szeliski. Refocusing plenoptic images using depth-adaptive splatting. In *Computational Photography (ICCP), 2014 IEEE International Conference on*, pages 1–9, May 2014.

[59] Roland W Fleming and Heinrich H Bülthoff. Low-level image cues in the perception of translucent materials. *ACM Transactions on Applied Perception*, 2(3), 2005.

[60] Roland W. Fleming, Ron O. Dror, and Edward H. Adelson. Real-world illumination and the perception of surface reflectance properties. *Journal of Vision*, 3(5), 2003.

[61] Roland W. Fleming, Christiane Wiebel, and Karl Gegenfurtner. Perceptual qualities and material classes. *Journal of Vision*, 13(8):9, 2013.

[62] Eduardo S. L. Gastal and Manuel M. Oliveira. Domain transform for edge-aware image and video processing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 30(3), 2011.

[63] Eduardo S. L. Gastal and Manuel M. Oliveira. Adaptive manifolds for real-time high-dimensional filtering. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 31(4), 2012.

[64] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[65] A. Gijsenij, R. Lu, and T. Gevers. Color constancy for multiple light sources. *IEEE Trans. on Image Processing*, 2011.

[66] L. Glondu, L. Muguercia, M. Marchal, C. Bosch, H. Rushmeier, G. Dumont,

and G. Drettakis. Example-based fractured appearance. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering)*, 31(4), 2012.

[67] Aleksey Golovinskiy, Wojciech Matusik, Hanspeter Pfister, Szymon Rusinkiewicz, and Thomas Funkhouser. A statistical model for synthesis of detailed facial geometry. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3), 2006.

[68] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 43–54, New York, NY, USA, 1996. ACM.

[69] Joachim Guanzon and Marden Blake. Video of computational design workflow (https://http://vimeo.com/30363913), 2011.

[70] Johannes Hanika, Holger Dammertz, and Hendrik P. A. Lensch. Edge-optimized à-trous wavelets for local contrast enhancement with robust denoising. *Computer Graphics Forum*, 30(7), 2011.

[71] Mark Harris, Shubhabrata Sengupta, and John D. Owens. Parallel prefix sum (scan) with CUDA. In Hubert Nguyen, editor, *GPU Gems 3*, chapter 39, pages 851–876. Addison Wesley, August 2007.

[72] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *Proceedings of European Conference on Computer Vision*, 2010.

[73] J. Hedgecoe. *New Manual of Photography*. Dorling Kindersley, 2009.

[74] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *Proc. of ACM SIGGRAPH*, 1995.

[75] Eugene Hsu, Tom Mertens, Sylvain Paris, Shai Avidan, and Frédo Durand. Light mixture estimation for spatially varying white balance. *ACM Trans. on Graphics*, 27(3), 2008.

[76] R.W.G. Hunt. *The Reproduction of Colour*. Fountain Press, 1987.

[77] F. Hunter, P. Fuqua, and S. Biver. *Light Science and Magic 4/e*. Elsevier Science, 2011.

[78] R.E. Jacobson. *The manual of photography: photographic and digital imaging*. Media Manual Series. Focal Press, 2000.

[79] Tilke Judd, Frédo Durand, and Edward Adelson. Apparent ridges for line drawing. *ACM Transactions on Graphics*, 26(3), 2007.

[80] Levent Karacan, Erkut Erdem, and Aykut Erdem. Structure preserving image smoothing via region covariances. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 32(6), 2013.

[81] Michael P Kelley. Video of computational design workflow (https://www.youtube.com/watch?v=J-exuHchmSk), 2011.

[82] Michael P Kelley. Private communication with professional photographer, 2012.

[83] Erum Arif Khan, Erik Reinhard, Roland W Fleming, and Heinrich H Bülthoff. Image-based material editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3), 2006.

[84] Juno Kim, Phillip Marlow, and Barton L. Anderson. The perception of gloss depends on highlight congruence with surface shading. *Journal of Vision*, 11(9):4, 2011.

[85] Juno Kim, Phillip Marlow, and Barton L. Anderson. The perception of gloss depends on highlight congruence with surface shading. *Journal of Vision*, 11(9), 2011.

[86] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. *ACM Transactions on Graphics*, 26(3), 2007.

[87] Philipp Krahenbuhl and Vladlen Koltun. Learning to propose objects. June 2015.

[88] Edmund Y . Lam and George S . K . Fung. Automatic white balancing in digital photography. In *Single-Sensor Imaging Methods and Applications for Digital Cameras*, pages 267–294. CRC Press 2008, 2008.

[89] Edwin H. Land, John, and J. Mccann. Lightness and retinex theory. *Journal of the Optical Society of America*, pages 1–11, 1971.

[90] P. Lee and Ying Wu. Nonlocal matting. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 2193–2200, Washington, DC, USA, 2011. IEEE Computer Society.

[91] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[92] Anat Levin, Rob Fergus, Frédo Durand, and William T. Freeman. Image and depth from a conventional camera with a coded aperture. In *ACM SIGGRAPH 2007 papers*. ACM, 2007.

[93] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 31–42, New York, NY, USA, 1996. ACM.

[94] Chia-Kai Liang and Ravi Ramamoorthi. A light transport framework for lenslet light field cameras. *ACM Trans. Graph.*, 34(2):16:1–16:19, March 2015.

[95] Zicheng Liao, Neel Joshi, and Hugues Hoppe. Automated video looping with progressive dynamism. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*, 32(4), 2013.

[96] Haiting Lin, Seon Joo Kim, S. Susstrunk, and M.S. Brown. Revisiting radiometric calibration for color computer vision. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 129–136, Nov 2011.

[97] Dani Lischinski, Zeev Farbman, Matt Uyttendaele, and Richard Szeliski. Interactive local adjustment of tonal values. *ACM Trans. on Graphics*, 25(3), 2006.

[98] Ce Liu, Lavanya Sharan, Edward H. Adelson, and Ruth Rosenholtz. Exploring features in a bayesian framework for material recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[99] Song Liu, Liang-Tien Chia, and Deepu Rajan. Attention region selection with information from professional digital camera. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, pages 391–394, New York, NY, USA, 2005. ACM.

[100] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. pages 674–679, 1981.

[101] Satya P. Mallick, Todd Zickler, Peter N. Belhumeur, and David J. Kriegman. Specularity removal in images and videos: a pde approach. In *Proceedings of the 9th European conference on Computer Vision - Volume Part I*, ECCV'06. Springer-Verlag, 2006.

[102] Satya P. Mallick, Todd Zickler, Peter N. Belhumeur, and David J. Kriegman. Specularity removal in images and videos: A pde approach. In *In Proc. of ECCV*, pages 550–563, 2006.

[103] RafałMantiuk, Kil Joong Kim, Allan G. Rempel, and Wolfgang Heidrich. HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 30(4), 2011.

[104] RafałMantiuk, Karol Myszkowski, and Hans-Peter Seidel. A perceptual framework for contrast processing of high dynamic range images. *ACM Transactions on Applied Perception*, 2006.

[105] Phillip J. Marlow and Barton L. Anderson. Generative constraints on image cues for perceived gloss. *Journal of Vision*, 13(14):2, 2013.

[106] PhillipJ. Marlow, Juno Kim, and BartonL. Anderson. The perception and misperception of specular surface reflectance. *Current Biology*, 22(20):1909 – 1913, 2012.

[107] Tom Mertens, Jan Kautz, Jiawen Chen, Philippe Bekaert, and Frdo Durand. Texture transfer using geometry correlation. In *Proc. of Eurographics Symposium on Rendering*, 2006.

[108] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*. IEEE Computer Society, 2007.

[109] Isamu Motoyoshi, Shin'ya Nishida, Lavanya Sharan, and Edward H. Adelson. Image statistics and the perception of surface qualities. *Nature*, 2007.

[110] Suzanne Nalbantian. Neuroaesthetics: neuroscientific theory and illustration from the arts. *Interdisciplinary Science Reviews*, 33(4):357–368.

[111] Sudha Natarajan. *Euclidean Distance Transform and Its Applications*. AV Akademikerverlag GmbH & Co. KG., 2010.

[112] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. Technical report, 2005.

[113] Addy Ngan, Frédo Durand, and Wojciech Matusik. Image-driven navigation of analytical BRDF models. In *Proceedings of the Eurographics Symposium on Rendering*, 2006.

[114] Y. Ostrovsky, P. Cavanagh, and P. Sinha. Perceiving illumination inconsistencies in scenes. *Perception*, 34, 2005.

[115] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. *International Journal of Computer Vision*, 81(1), 2009.

[116] Sylvain Paris, Samuel W. Hasinoff, and Jan Kautz. Local Laplacian filters: Edge-aware image processing with a Laplacian pyramid. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 30(4), 2011.

[117] Genevieve Patterson, Chen Xu, Hang Su, and James Hays. The sun attribute database: Beyond categories for deeper scene understanding. *Int. J. Comput. Vision*, 108(1-2):59–81, May 2014.

[118] Fabio Pellacini. Envylight: an interface for editing natural illumination. *ACM Trans. Graph.*, 29, July 2010.

[119] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. In *ACM SIGGRAPH 2004 Papers*, 2004.

[120] Javier Portilla and Eero P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *Int. Journal Computer Vision*, 40(1):49–70, October 2000.

[121] Vilayanur S. Ramachandran and William Hirstein. The science of art: A neurological theory of aesthetic experience. *Journal of Consciousness Studies*, 6(6-7):15–41, 1999.

[122] Ramesh Raskar, Kar-Han Tan, Rogerio Feris, Jingyi Yu, and Matthew Turk.

Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, New York, NY, USA, 2004. ACM.

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

[124] C. Riess, E. Eibenberger, and E. Angelopoulou. Illuminant color estimation for real-world mixed-illuminant scenes. In *IEEE Color and Photometry in Computer Vision Workshop*, 2011.

[125] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, August 2004.

[126] Szymon Rusinkiewicz, Michael Burns, and Doug DeCarlo. Exaggerated shading for depicting shape and detail. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3), July 2006.

[127] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. 3-d depth reconstruction from a single still image. *International Journal of Computer Vision (IJCV*, 76:2007, 2007.

[128] Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. Video textures. 2000.

[129] Chris Schoeneman, Julie Dorsey, Brian Smits, James Arvo, and Donald Greenberg. Painting with light. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, New York, NY, USA, 1993. ACM.

[130] D.H.H.M.D.J.F.E.U.P.N.H.M. School and B.B.P.T.N.W.M.D.D.S.W.L.L.C. Mind. *Brain and Visual Perception : The Story of a 25-Year Collaboration: The Story of a 25-Year Collaboration*. Oxford University Press, USA, 2004.

[131] L. Sharan, C. Liu, R. Rosenholtz, and E. H. Adelson. Recognizing materials using perceptually inspired features. *International Journal of Computer Vision*, 2013.

[132] Lavanya Sharan, Yuanzhen Li, Isamu Motoyoshi, Shin'ya Nishida, and

Edward H. Adelson. Image statistics for surface reflectance perception. *Journal of the Optical Society of America A*, 25(4), 2008.

[133] Jianbing Shen, Xiaoshan Yang, Yunde Jia, and Xuelong Li. Intrinsic images using optimization. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[134] Li Shen, Ping Tan, and Stephen Lin. Intrinsic image decomposition with non-local texture cues. In *IEEE Conf. on Computer Vision and Patten Recognition*, 2008.

[135] YiChang Shih, Sylvain Paris, Connelly Barnes, William T Freeman, and Frédo Durand. Style transfer for headshot portraits. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*, 33(4), 2014.

[136] E. P. Simoncelli and E. H. Adelson. Noise removal via bayesian wavelet coring. In *Proc. of IEEE International Conference on Image Processing*, 1996.

[137] Manohar Srikanth, Kavita Bala, and Frédo Durand. Computational rim illumination with aerial robots. In *Proc. of Workshop on Computational Aesthetics*, 2014.

[138] R.T. Tan, K. Nishino, and K. Ikeuchi. Separating reflection components based on chromaticity and noise analysis. *IEEE Trans. Pattern. Anal. Mach. Intell.*, 2004.

[139] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000.

[140] TissotAd. Example video of lighting design for product videography (https://www.youtube.com/watch?v=bpe88mi0ebq), 2011.

[141] TissotAd. Example video of lighting design for product videography (https://www.youtube.com/watch?v=lliuhzzkqa8), 2014.

[142] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Proc. of IEEE Int. Conf. on Computer Vision*, 1998.

[143] M. Trentacoste, R. Mantiuk, and W. Heidrich. Blur-aware image downsizing. *Computer Graphics Forum (Proc. Eurographics)*, 2011.

[144] M. Trentacoste, R. Mantiuk, W. Heidrich, and F. Dufrot. Unsharp masking, countershading and halos: Enhancements or artifacts? *Computer Graphics Forum (Proc. Eurographics)*, 2012.

[145] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *European Conf. on Comp. Vision*, 2008.

[146] Romain Vergne, Pascal Barla, Roland Fleming, and Xavier Granier. Surface flows for image-based shading design. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 31(3), 2012.

[147] Javier von der Pahlen, Jorge Jimenez, Etienne Danvoye, Paul Debevec, Graham Fyffe, and Oleg Alexander. Digital ira and beyond: Creating real-time photoreal digital actors. In *ACM SIGGRAPH 2014 Courses*, SIGGRAPH '14, pages 1:1–1:384, New York, NY, USA, 2014. ACM.

[148] Quoc Kien Vuong, Se hwan Yun, and Suki Kim. A new auto exposure and auto white-balance algorithm to detect high dynamic range conditions using cmos technology, 2008.

[149] Gregory J. Ward. Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph.*, 26(2):265–272, July 1992.

[150] Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*, 24(3), 2005.

[151] Wikipedia. Digital camera modes — Wikipedia, the free encyclopedia. [Online; accessed 26-May-2015].

[152] Holger Winnemöeller, Ankit Mohan, Jack Tumblin, and Bruce Gooch. Light waving: Estimating light positions from photographs alone. *Computer Graphics Forum*, 24(3), 2005.

[153] Holger Winnemöeller, Ankit Mohan, Jack Tumblin, and Bruce Gooch. Light waving: Estimating light positions from photographs alone. *Computer Graphics Forum (Proc. of Eurographics)*, 24(3), 2005.

[154] Bei Xiao, Bruce Walter, Ioannis Gkioulekas, Todd Zickler, Edward Adelson, and Kavita Bala. Looking against the light: How perception of translucency depends on lighting direction. *Journal of Vision*, 14(3), 2014.

[155] Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. Image smoothing via L0 gradient minimization. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 30(6), 2011.

[156] Li Xu, Qiong Yan, Yang Xia, and Jiaya Jia. Structure extraction from texture via relative total variation. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 31(6), 2012.

[157] S Zeki and L Marini. Three cortical stages of colour processing in the human brain. *Brain*, 121(9):1669–1685, 1998.

[158] Semir Zeki. Statement on neuroesthetics., 2007.