Prof. Hanspeter Pfister Kalyan Krishna Sunkavalli

Models of Visual Appearance for Analyzing and Editing Images and Videos

Abstract

The visual appearance of an image is a complex function of factors such as scene geometry, material reflectances and textures, illumination, and the properties of the camera used to capture the image. Understanding how these factors interact to produce an image is a fundamental problem in computer vision and graphics. This dissertation examines two aspects of this problem: models of visual appearance that allow us to recover scene properties from images and videos, and tools that allow users to manipulate visual appearance in images and videos in intuitive ways. In particular, we look at these problems in three different applications.

First, we propose techniques for compositing images that differ significantly in their appearance. Our framework transfers appearance between images by manipulating the different levels of a multi-scale decomposition of the image. This allows users to create realistic composites with minimal interaction in a number of different scenarios. We also discuss techniques for compositing and replacing facial performances in videos.

Second, we look at the problem of creating high-quality still images from low-quality video clips. Traditional multi-image enhancement techniques accomplish this by inverting the camera's imaging process. Our system incorporates feature weights into these image models to create results that have better resolution, noise, and blur characteristics, and summarize the activity in the video.

Finally, we analyze variations in scene appearance caused by changes in lighting. We develop a model for outdoor scene appearance that allows us to recover radiometric and geometric information about the scene from images. We apply this model to a variety of visual tasks, including color-constancy, background subtraction, shadow detection, scene reconstruction, and camera geo-location. We also show that the appearance of a Lambertian scene can be modeled as a combination of distinct three-dimensional illumination subspaces — a result that leads to novel bounds on scene appearance, and a robust uncalibrated photometric stereo method.

Contents

1 Introduction **1**

1.1 Representations for Visual Appearance 5

1.2 Outline of Dissertation 9

2 Multi-scale Representations for Image Appearance **11**

2.1 Introduction 12

2.2 Related Work 13

2.3 Overview 16

2.4 Smooth Histogram Matching 20

2.5 Structure and Noise Matching 23

2.6 Pyramid compositing 24

2.7 Results and Discussion 28

2.8 Summary 35

3 Editing Faces in Videos **38**

3.1 Introduction 39

3.2 Related Work 42

3.3 Overview 44

3.4 Face Tracking . . . 46
3.5 Spatial and Temporal Alignment . . . 50
3.6 Blending . . . 53
3.7 Results and Discussion . . . 59
3.8 Summary . . . 65

4 Enhancing Image Quality using Video Clips **66**
4.1 Introduction . . . 67
4.2 Related Work . . . 69
4.3 Importance-based Image Enhancement . . . 71
4.4 Creating Video Snapshots . . . 74
4.5 Results . . . 80
4.6 Summary . . . 86

5 Appearance Changes in Outdoor Scenes **90**
5.1 Introduction . . . 91
5.2 Related Work . . . 92
5.3 A Color Model for Outdoor Image Sequences . . . 97
5.4 Implications for Machine Vision . . . 105
5.5 Summary . . . 110

6 Shadows and Scene Appearance **112**
6.1 Introduction . . . 113
6.2 Related Work . . . 114
6.3 Visibility Subspaces . . . 116
6.4 Estimating Visibility Subspaces . . . 119
6.5 Subspaces to Surface Normals . . . 121

6.6 Results 123
6.7 Summary 126

7 Summary and Future Directions **132**

7.1 Summary 132
7.2 Future Directions 134

References **137**

Figures

1.0.1 Photographs of Memorial Hall. . . . 3
1.1.1 Modeling visual appearance . . . 9
2.1.1 Image compositing. . . . 13
2.2.1 An overview of the Multi-scale Image Harmonization framework. . . . 16
2.3.1 An image compositing example. . . . 18
2.4.1 Smooth histogram matching. . . . 20
2.6.1 Quadtree solver. . . . 27
2.7.1 Style transfer. . . . 29
2.7.2 Matching texture. . . . 30
2.7.3 Adding and removing noise. . . . 31
2.7.4 Matching blur. . . . 32
2.7.5 Compositing with mixed boundary conditions. . . . 34
2.7.6 Limitations. . . . 35
2.7.7 Matching contrast and noise. . . . 37
3.1.1 Video face replacement. . . . 39
3.2.1 An overview of our method. . . . 44
3.4.1 User interface for tracking. . . . 47

3.5.1 Video retiming. . . . 51
3.6.1 Seam computation for blending. . . . 54
3.7.1 Multi-take video montage. . . . 58
3.7.2 Dubbing using face replacement. . . . 59
3.7.3 Face replacement. . . . 61
3.7.4 Face retargeting. . . . 62
3.7.5 Failure cases. . . . 63
4.1.1 Comparisons of image enhancement. . . . 68
4.3.1 Weighted multi-image enhancement. . . . 73
4.4.1 Sparsifying the feature weights. . . . 78
4.5.1 Video snapshots with saliency weights. . . . 81
4.5.2 Video snapshots for motion blur. . . . 82
4.5.3 Video snapshots for defocus blur. . . . 82
4.5.4 Video snapshots with motion. . . . 83
4.5.5 Video snapshots with saliency weights. . . . 84
4.5.6 Video snapshots with temporal effects. . . . 85
4.5.7 Video snapshots with motion. . . . 86
4.5.8 Video snapshots with saliency weights. . . . 87
4.5.9 Video snapshots with saliency weights. . . . 87
4.5.10 Comparisons with multi-image super-resolution. . . . 88
5.3.1 Color and shadow initialization. . . . 103
5.3.2 Reconstructions from our model. . . . 104
5.4.1 Color constancy using our model. . . . 105
5.4.2 Simple foreground detection using per-pixel thresholds in color space. . . . 106
5.4.3 Partial scene reconstruction and camera geo-location. . . . 109
5.4.4 Shadow detection. . . . 109

6.6.1 Surface reconstruction for the spheres synthetic dataset. 124
6.6.2 Surface reconstruction for the spheres and plane synthetic dataset. 125
6.7.1 Surface reconstruction for the frog dataset. 129
6.7.2 Surface reconstruction for the scholar dataset. 130
6.7.3 Visibility subspace estimation and normal recovery. 131
7.2.1 Directions for future work. 135

Tables

4.4.1 Summary of results. 80

5.3.1 RMS reconstruction errors. 104

5.4.1 Camera geo-location results. 108

Acknowledgments

I am extremely grateful to my advisor, Hanspeter Pfister, whose guidance made this dissertation possible. Six years ago, Hanspeter took a chance on me as an intern at Mitsubishi Electric Research Labs, and I owe much of what I know about being a researcher and a professional to his mentoring.

This dissertation is the result of collaborations with a number of researchers — Michael Cohen, Kevin Dale, Kimo Johnson, Neel Joshi, Sing Bing Kang, Wojciech Matusik, Fabiano Romeiro, and Todd Zickler. I would like to thank all of them for their contributions to both this work and to my growth as an academic. I would especially like to thank Todd, who has been a great teacher and an invaluable resource for discussing half-baked ideas with, Wojciech, whose creativity inspired many of the ideas in this work, Kimo, who seemed to have the answer to every problem we encountered, and Kevin, who has accompanied me on my Ph.D. journey from start to finish.

I would also like to thank all my wonderful colleagues who made graduate school so enjoyable. The VCG group was a constant source of inspiring discussions, interesting stories, and exciting adventures. Thanks also to Moritz Baecher, Karthik Dantu, Sanjeev Koppal, and Ritwik Kumar for always keeping things from getting too serious. Moritz, my labmate for five years, deserves a special mention for putting up with all my quirks.

Thanks to my parents, Padma and Rammohan, for encouraging me to follow my dreams, and always challenging me to be the best I can be. Thanks also to my brother, Shashank, for his unconditional love and support. Finally, thanks to my wife and personal cheering squad, Nupur, whose company makes everything in my life better.

To Tatagaru, for your unwavering belief in me.

1

Introduction

Every day hundreds of people visit Memorial Hall at Harvard University and take photographs like the ones in Fig. 1.0.1. By themselves, each of these images represents a single example of what Memorial Hall looks like from one point in space and at one instant in time. However, an internet search for photos of Memorial Hall will yield hundreds of such images, shot from every conceivable position and angle, under varying weather and lighting conditions, and at different times during the day and night. Today, with the rapid proliferation of cameras (especially cell-phone cameras), and the emergence of photo-sharing websites (Facebook, Instagram, Flickr), it is easier than ever to capture, store, and share images. For example, around 250 million

photos are being uploaded to Facebook everyday[1].

While capturing images is easy, analyzing their appearance is still a very difficult problem and, editing and enhancing them often requires time and expertise. Fig. 1.0.1 illustrates the challenges inherent to understanding images. Even though they depict the same building, these photographs are significantly different from each other. While the range of appearance in these photos is remarkable, it is not entirely surprising. Each of these images is a complex function of the illumination at the instant the photograph was taken, the geometry and material properties of Memorial Hall, and the location and settings of the camera that was used to capture the photograph; varying any one of these factors could result in dramatic changes in appearance.

While it is known that illumination, geometry, materials, and cameras interact in intricate ways to create images (see Fig. 1.1.1), it is clear from Fig. 1.0.1 that there a number of visual cues embedded in these photographs. For example, it is easy to see that one of these photographs was taken during daytime (because the illumination looks "white"), two were captured close to dawn or dusk (because of the orange-red hues corresponding to sunrise and sunset), and the other two were taken under overcast skies. Without knowing anything specific about Memorial Hall *a priori*, we can get a sense for the shape of the building and estimate the positions of the cameras that captured these images. From the textures in these images, we might be able to recognize that Memorial Hall is made up of brick, sandstone, and slate, and has windows made of stained glass. We might even be able to deduce that some of these cameras have white-balance settings that are biased towards blue, while others are skewed towards red.

These observations raise the question: how are we are able to untangle all this information when the image formation process is so complicated? The answer to this question is two-fold. First, even though the factors involved in the image formation process are complex, they vary in very structured ways. For example, low-dimensional models have been proposed for natural illumination [140], real-world surface reflectances [122], and camera pipelines [35]. Thus, one way

[1] http://www.sec.gov/Archives/edgar/data/1326801/000119312512034517/d287954ds1.htm

Figure 1.0.1: *Photographs of Memorial Hall at Harvard University, captured from different views, at different points in times, show dramatic changes in visual appearance. Photo credits: Flickr users* `wallyg`, `Emily Taliaferro Prince`, `arcticpenguin`, *and* `somewhereglad1ybeyond`.*)*

to make the problem of image understanding more tractable is to use models of appearance that explicitly leverage this coherence. Second, while inferring scene properties from a single image is ill-posed, capturing video sequences where only a few of these factors vary makes the problem better constrained. This property has been leveraged in work on scene reconstruction [77, 182], intrinsic image decompositions [180], and multi-image super-resolution and deblurring [14]. Given the ease with which images and video clips can be captured today, this is another effective way of making appearance modeling feasible.

Driven by these two insights, this dissertation explores models of visual appearance (for both single images and videos) that make image understanding tractable. In particular, our work answers two questions:

1. **Can we develop models for visual appearance that allow us to analyze images and recover different scene properties?** The representations we propose explicitly model the structure in image data and leverage it to recover scene characteristics. Furthermore, we show that our models enable a number of applications such as texture analysis, illumination recovery, and scene reconstruction.

2. **Can we build tools that allow users to edit and enhance their photographs and videos in easy, intuitive ways?** Traditional image and video editing tools require users to manipulate pixel values — a process that is tedious, and often not intuitive. Instead, by modeling visual appearance appropriately, our work enables high-level editing operations, such as texture manipulation, face replacement, and video summarization.

Modeling visual appearance is a fundamental problem in computer vision and computer graphics with applications ranging from object recognition and scene understanding, to rendering and image editing. In this dissertation, we focus on three applications:

Image and video compositing: Compositing images and videos is a tedious process that often requires both time and expertise with editing tools. This is especially true when the images and videos to be merged are captured in different conditions and have differing appearances. In this dissertation, we propose tools that automatically match the appearance of images and videos, making the creation of photo-realistic composites an easy and intuitive process.

Image enhancement: Images and videos captured by low-quality cameras often suffer from artifacts such as noise, blur, and compression. Improving the quality of these images is a difficult problem with a long history in computer vision and computer graphics. In this work, we present a novel image enhancement framework that leverages the data captured in the multiple frames of a video clip to create high-quality still images.

Scene understanding: In the third set of applications, we propose tools that can analyze images and videos to extract scene properties such as material reflectances and surface geometry from them. In particular, we look at the problem of analyzing variations in the appearance of both outdoor and indoor scenes caused due to changes in illumination. By modeling the illumination and the scene appropriately, we show that we can enable a number of applications such as scene reconstruction, color constancy, background estimation, and camera geolocation.

Visual appearance has been studied extensively in the literature because of its central role in

computer vision and graphics. Since our work builds on some of these ideas, the remainder of this chapter discusses representations for appearance that have been proposed in the past, and places our work in the context of this previous research. We end this chapter with an outline for the rest of this dissertation.

1.1 Representations for Visual Appearance

In this section, we review work related to this dissertation and, in particular, focus on representations that enable image understanding tasks. While we discuss these representations in very general terms here, each chapter includes a thorough discussion of relevant work.

We categorize models for appearance, based on how strongly they constrain visual appearance, into: weak cues based on the statistical properties of images, intermediate image-based representations (like intrinsic images), and scene-based representations that completely model geometry, reflectance, and illumination. Each of the models we propose and use in this dissertation falls in one of these categories.

1.1.1 Statistical representations

Natural image statistics: It is well-known that natural images lie in a very restricted subspace of the set of all possible images [148], indicating that natural images have strong correlations between pixels. In particular, previous work has shown that the gradients of natural images form heavy-tailed distributions [154], a property that has been applied to a variety of vision problems including denoising [146], super-resolution [166], and deblurring [62, 108].

Illumination: Similar statistical models have been proposed for natural illumination [52] and have been applied to problems in reflectometry [145].

Textures: Textures are often convolved with filters of varying scales and orientations and charac-

terized using the statistical distributions of the resulting filter responses. While such techniques are particularly suited for stochastic textures [79], they have also been adapted to analyze and synthesize structured textures [139].

In Chapter 2, we use similar statistical models to represent single images. In particular, we propose techniques to transfer the appearance between images by manipulating their multi-scale filter decompositions. This allows us to easily create photo-realistic composites from disparate images.

1.1.2 Image-based representations

While statistical models have been shown to be useful for a number of vision tasks, they place only weak constraints on visual appearance. At the other extreme, we could completely model all the components of the image formation process, but that is often very difficult to achieve. Instead, images are often analyzed in terms of scene properties that are defined in the 2-d image space. Such decompositions are general enough to represent a number of visual phenomena, but specific enough that they can be tractably estimated from images.

Intrinsic images: The notion of intrinsic images was introduced by Barrow and Tenenbaum [12], who proposed decomposing an image into the intrinsic characteristics of a scene, including illumination, reflectance, and surface geometry. Since then, a number of related representations have been proposed. One such decomposition involves separating a single grayscale image into the product of per-pixel components for illumination (shading) and surface reflectance (albedo) [165]. Finlayson et al. [63, 65] propose an alternative, color-based decomposition that recovers a reflectance component that is independent of both shading and source color.

The problem of deriving intrinsic images from a single image is highly under-constrained; it can however be simplified by using multiple images of a single scene under varying illumination. Weiss [180] uses a maximum-likelihood framework to estimate a single reflectance image and

multiple illumination images from grayscale time-lapse video. Matsushita et al. [120] generalize this framework by deriving time-varying reflectance and illumination images from similar data sets.

Intrinsic image decompositions allow a variety of image understanding and editing tasks, including, illumination-invariant material segmentation [192], and image recoloring [34]. In Chapter 5, we present a model for the appearance of outdoor scenes that decomposes images into per-pixel estimates of reflectance and geometry, and global estimates of outdoor illumination. We use this model for a number of visual tasks including color constancy, background subtraction, and geolocation.

Imaging models: Camera imaging models are often described in the 2-d image space. Such models are particularly useful for vision applications because they do not require an understanding of the 3-d geometry of the scene, and often lead to tractable algorithms for visual inference. For example, image enhancement techniques like super-resolution (and deblurring and denoising) describe low-resolution images as a 2-d warping and blurring of the unknown high-resolution latent image [86]. In Chapter 3, we compute image-based features such as sharpness and saliency and incorporate them into similar imaging models to recover a high-quality image from a low-quality video.

1.1.3 Scene modeling

In principle, the most complete way to model scene appearance is to explicitly measure all its characteristics, including geometry, material properties, and illumination (see Fig. 1.1.1). This approach is often used in computer graphics to model scenes and re-render them under varying viewing and lighting conditions. Here we review common representations for each of these characteristics.

Reflectances: Surface reflectances are often represented using the 4-d Bidirectional Reflectance

Distribution Function (BRDF), which describes how light is reflected off a surface as a function of the incoming and outgoing lighting directions [127]. BRDFs can be represented by parametric models (such as Lambertian, Cook-Torrance, or Oren-Nayar models) or by using data-driven models [122]. More recently, analytical models that are derived from real BRDF data are being increasingly used for vision and graphics problems [126, 145]. In Chapters 5 and 6 we use the Lambertian model to represent surface reflectance.

Illumination: While physics-based models exist for some forms of illumination (for example, natural illumination [132]), illumination conditions are generally represented using environment maps, i.e., explicit measurements of incoming light from all directions in the scene [49]. Both representations have been used successfully to render scenes in computer graphics [49, 140], and for inverse rendering in vision and graphics [48, 105, 121]. In Chapter 5, we present a novel representation that accounts for both the angular and spectral variation of outdoor illumination. We show that using this model to analyze outdoor time-lapse image sequences allows us to recover scene reflectance and geometry.

Geometry: Surface geometry has a profound effect on image appearance in the form of both local shading effects, and well as non-local shadowing effects. Surface geometry is usually represented using depth (in the form of 3-d meshes or height fields) or surface normals. In some cases, such as with human faces, geometry is highly constrained and can be approximated using low-dimensional models [177]. In Chapter 2, we use models of face geometry to track, and edit facial performances in videos. In Chapters 5 and 6, we analyze variations in scene appearance resulting from changes in illumination to recover geometry in the form of surface normals.

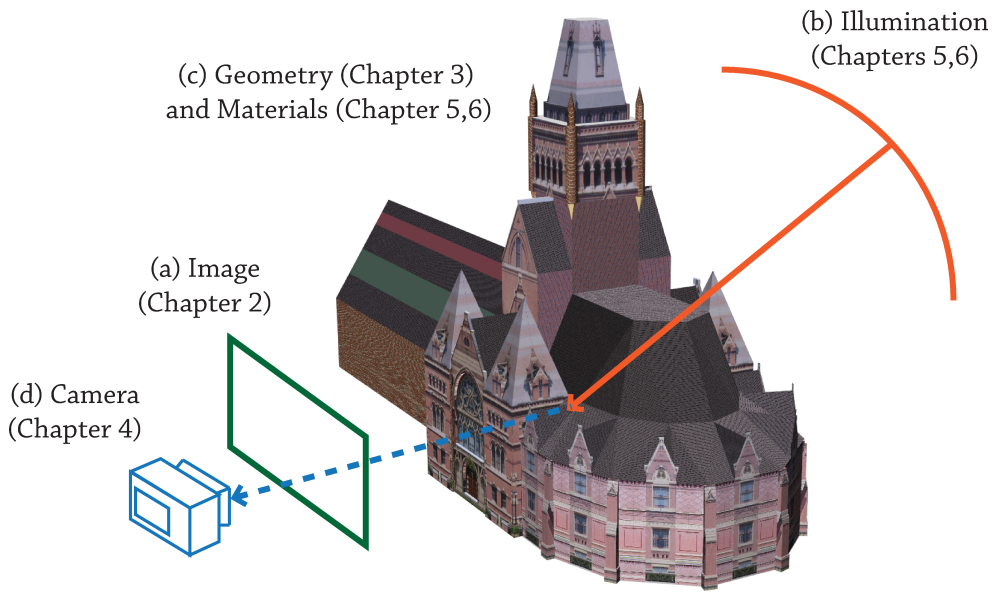


Figure 1.1.1: *Modeling visual appearance. Each image (a) shown in Fig. 1.0.1 is the result of illumination from the sun and the sky (b) interacting with the geometry and material properties of the scene (c), and being captured by the camera. In this dissertation, we propose image analysis and editing tools that model one or more of these factors. Memorial hall model credit: Paul B. Cote.*

1.2 Outline of Dissertation

This dissertation explores a number of models for visual appearance and demonstrates their usefulness for image analysis, image editing, and scene understanding. The models we explore range from statistical representations to full-fledged scene models. In each case, we model a different aspect of the image formation process as illustrated in Fig. 1.1.1.

In Chapter 2, our goal is to create highly photo-realistic composites from images that differ significantly in their appearance. We model the *appearance of a single image* using the statistical properties of its pixel correlations. We use this to transfer appearance between images and create photo-realistic composites in easy and intuitive ways.

In Chapter 3, we extend ideas from Chapter 2 to videos. We use a multi-linear model for face *geometry* to track and replace facial performances in videos.

In Chapter 4, we use *camera* imaging models to describe the frames of a video clip. We combine multiple low-quality video frames to create a single high-quality video snapshot that has better

resolution, noise, and blur characteristics, and even captures the motion in the video.

In Chapter 5, we analyze the appearance of outdoor scenes captured under time-varying *natural illumination*. By modeling the lighting, we show that we can recover scene albedo and geometry, and use this for tasks such as color constancy, background subtraction, and camera geolocation.

In Chapter 6, we analyze the appearance of non-convex Lambertian scenes under changing directional *illumination*. We analyze the effect of shadows on scene appearance, and propose a robust uncalibrated photometric algorithm that can recover high-quality surface *geometry*.

We discuss future steps for each these applications in the individual chapters. In Chapter 7, we summarize the contributions of this dissertation and propose new avenues for research.

Parts of the research presented in this dissertation have appeared in the following publications:

1. Sunkavalli, K., Johnson, M.K., Matusik, W., Pfister, H.: Multi-scale Image Harmonization. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 29(4), 125:1–10 (Jul 2010).

2. Dale, K, Sunkavalli, K., Johnson, M.K., Vlasic, D., Matusik, W., Pfister, H.: Video Face Replacement. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia) 30(6), 130:1–10 (Dec 2011).

3. Sunkavalli, K., Joshi, N., Kang, S.B., Cohen, M.F., Pfister, H.: Video Snapshots: Creating High-Quality Images from Video Clips. IEEE Transactions on Visualization and Computer Graphics, (to appear).

4. Sunkavalli, K., Romeiro, F., Matusik, W., Zickler, T., Pfister, H.: What do color changes reveal about an outdoor scene? Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1–8 (2008).

5. Sunkavalli, K., Zickler, T., Pfister, H.: Visibility Subspaces: Uncalibrated Photometric Stereo in the Presence of Shadows. Proceedings of the European Conference on Computer (ECCV) pp. 251–264 (2008).

2

Multi-scale Representations for Image Appearance

In this chapter, we explore representations for the appearance of a single image. In particular, we focus on models that capture low-level image characteristics such as global and local contrast, texture, noise, and blur. We show that the statistics of a multi-scale decomposition of an image are particularly suited to this task. Based on this insight, we develop a technique for transferring visual appearance across images and use it to create photo-realistic composites from disparate images.

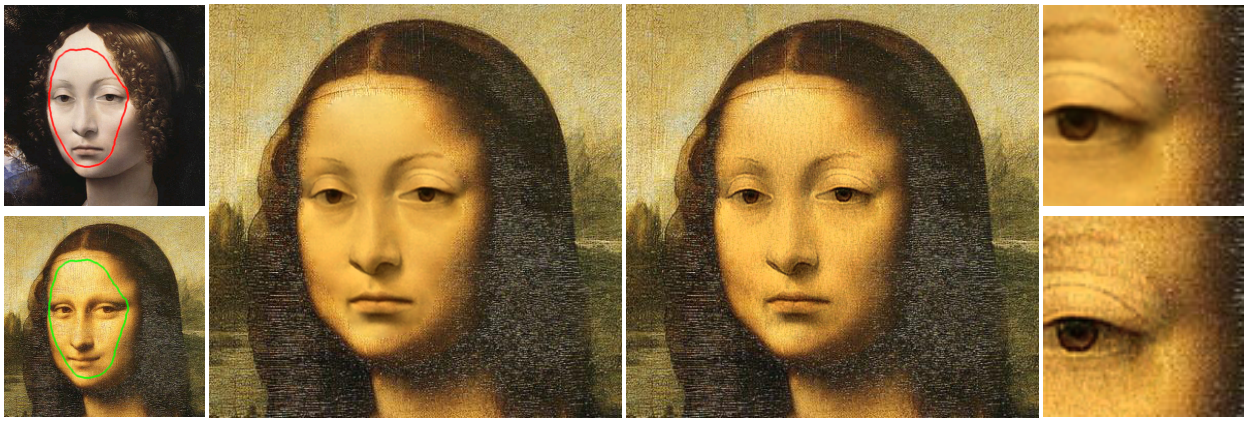
2.1 Introduction

Combining regions of multiple photographs or videos into a seamless composite is a fundamental problem in many vision and graphics applications, such as image compositing, mosaicing, scene completion, and texture synthesis. In order to produce realistic composites, it is important to ensure that the boundaries between the images being combined appear as seamless and natural as possible. This can be achieved through alpha matting, where pixel values are combined using a user-specified alpha matte, or through gradient-domain compositing techniques, which reconstruct pixel intensities from merged gradient vector fields.

While necessary, seamless boundaries are not always sufficient for creating realistic composites. Often the images being combined come from diverse sources and are shot by different cameras under different conditions. This is illustrated in Fig. 2.1.1(a), where the user segments a novel face (top), and inserts it into another image (bottom). Gradient domain compositing (Fig. 2.1.1(b)) creates seamless boundaries in the composite. But because the two images are from different sources with different appearance, the two regions of the composite look inconsistent, detracting from the realism of the composite.

Currently, users fix these inconsistencies manually, and it takes even professional artists hours of work to produce highly realistic composites. In this work, we address this problem by building tools to automatically *harmonize* images before compositing them (Fig. 2.1.1(c)). By building methods to automatically correct inconsistencies in images with minimal user interaction, this work takes the burden of compensating for inconsistencies away from the user and makes compositing effortless and user-friendly.

The main contribution of this work is a unified framework that harmonizes aspects of appearance, such as contrast, texture, noise, and blur. This is guided by the insight that a multi-resolution pyramid representation for images is useful for both transferring different aspects of visual appearance between images and compositing them. We show that we can transfer appearance by



(a) Source / Target (b) Seamless Cloning (c) Harmonization (d) Close-ups

Figure 2.1.1: *Image compositing. In traditional image compositing, a user applies geometric transformations to a source image (a, top) and inserts it into a target image (a, bottom). Tools such as the Photoshop Healing Brush use gradient domain compositing to ensure that the composite is seamless (b) but the inconsistencies between the two images, make the result look unrealistic: the inserted face is much smoother than the rest of the image. Our method "harmonizes" the images before blending them, producing a composite that is seamless and realistic (c). The close-up images (d) compare traditional gradient-domain blending (top) to the harmonized result (bottom).*

manipulating the different levels of the pyramid of the source and target images so that their histograms match. We also present a novel method to reconstruct the composite from the modified pyramids in conjunction with boundary constraints based on matting as well as gradient-domain compositing. To our knowledge, this is the first work that explicitly addresses the problem of harmonizing images during compositing.

This work does not deal with inconsistencies in viewpoint, lighting, or shadows. We assume that the images are geometrically aligned and have compatible viewpoint and vanishing points.

2.2 Related Work

2.2.1 Alpha matting

The simplest way to fuse images is to combine their absolute pixel values. This is often accomplished through alpha matting [137], where the colors of the images are linearly interpolated using

weights specified by the alpha matte. Recent work in this area has focused on making the matte creation as easy as possible [159, 178], but has not corrected for appearance differences.

2.2.2 Gradient-domain compositing

Often two images need to be merged *seamlessly*, i.e., the boundary between them should be imperceptible. Gradient-domain techniques accomplish this by combining image gradients (instead of absolute pixel values) and solving for the composite that would best produce the fused gradient field. These techniques were introduced to the imaging community by Pérez et al. [131] and have since become the standard for seamless compositing [4, 109] and a part of editing tools such as Photoshop [72]. Perez et al. also propose variations of seamless cloning (such as mixing the source and target gradients) to handle differences in texture, but these solutions work only on very specific images. More recently, Farbman et al. [60] showed that the solution to the Poisson linear system could be approximated using a novel interpolation scheme. This work did not consider issues related to harmonization of the source images, but did show that large image regions could be cloned at interactive rates. In general, our method extends gradient-domain techniques by reconstructing images from a much larger set of filter outputs and integrates harmonization into the compositing framework.

2.2.3 Transfer of Visual Appearance

Most of the work on transferring visual appearance focuses on matching color distributions between images [103, 136, 143]. Cohen-Or et al. [43] presented ways to transform images such that their color palettes are perceptually harmonic. Closely related to our work, is the work of Bae et al. [9] on transferring tonal balance and level of detail from one image to another. They use a non-linear bilateral filter to decompose the images into two scales and match the histograms of these scales to match the style of the images. We show that we can achieve similar effects with linear

filters and do this in the context of image compositing. Chen et al. [39] present an interactive tool for separating the noise from an image; this noise can then be transferred to other images. In contrast, our approach automatically matches noise, contrast and blur using a single framework.

2.2.4 Multi-scale methods

Our work is inspired by Burt and Adelson's seminal work [33] on using multi-scale representations such as Laplacian pyramids [32] to composite images. The statistics of each level of an image pyramid are known to be correlated with different aspects of visual appearance and pyramid based representations have been widely used for many problems in vision and graphics including texture analysis and synthesis, object recognition and image retrieval, and transferring visual appearance. In all these works, images are decomposed into multi-scale pyramids and the different levels of the pyramids are then analyzed or manipulated to achieve the desired objective. A classic example of this approach is the work of Heeger and Bergen [79] who use pyramids for texture synthesis, and show that histogram matching the subband coefficients of a noise pyramid to those of a given texture can be used to generate synthetic stochastic textures.

A known problem with pyramids constructed using linear filters, is that applying nonlinear operations (such as tone-mapping and histogram matching) on the subband coefficients of images with structure often results in artifacts such as haloing along strong edges. As a result, recent work on multi-scale methods uses nonlinear edge-preserving filters like the bilateral filter [169] to construct the pyramids [9, 59, 61] and avoid haloing. In contrast to this, Li et al. [112] show that linear multi-scale decompositions used in conjunction with carefully controlled, smooth nonlinear operations (in their case, compressive transforms for high dynamic range tone mapping) do not lead to haloing artifacts.

Our work builds on previous uses of linear image pyramids in three ways. Firstly, we harmonize the appearance of the source and target images by histogram-matching the pyramid coefficients of the target to those of the source. Doing this naively could lead to artifacts but we show how

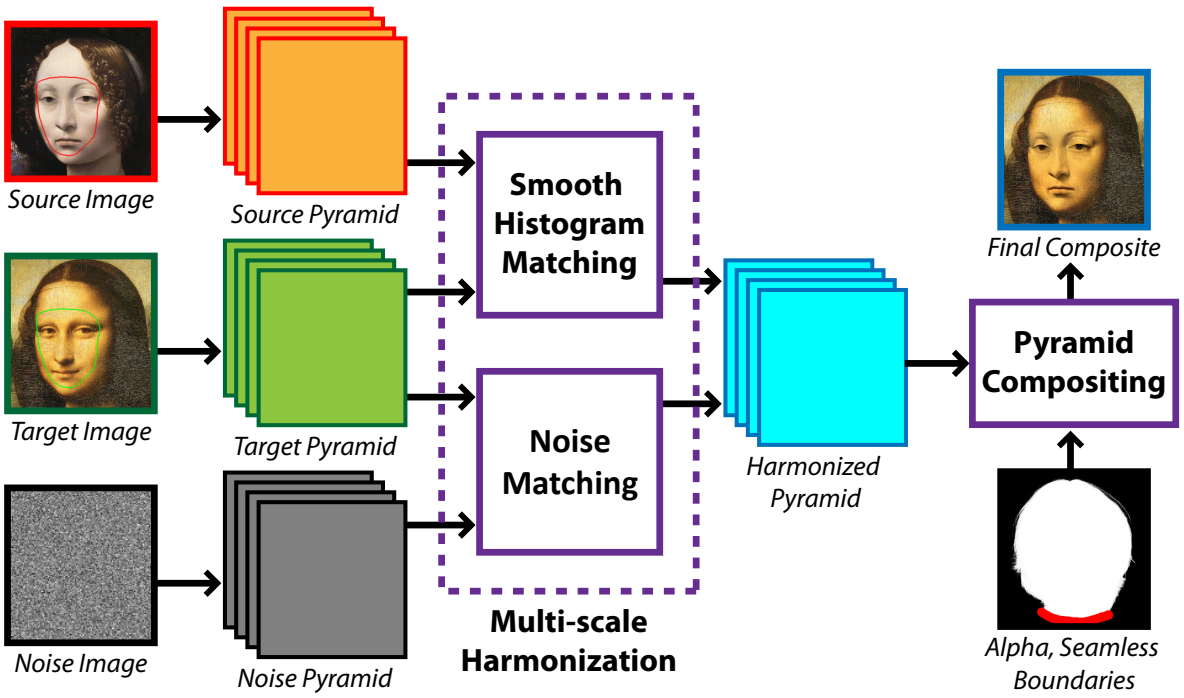


Figure 2.2.1: *An overview of the Multi-scale Image Harmonization framework. The input source and target images, and a uniform random noise image are decomposed into pyramids. Using a smooth histogram matching technique, the source and noise pyramids are iteratively shaped so that they match the target pyramid. This produces a harmonized pyramid from which the final composite is reconstructed by incorporating seamless and/or matte-based boundary conditions.*

regularizing the histogram transfer can minimize these artifacts. Secondly, we inject noise into the harmonization step and show how it can be shaped to handle differences in the noise and texture patterns between images. Finally, we introduce a novel way of computing the final composite from the histogram-matched pyramid coefficients by solving a linear system of equations while satisfying both seamless and matte based boundary conditions.

2.3 Overview

We assume that the user has a source image I^s with an object, or region, that they would like to insert into a target image I^t. The object in the source image may have different visual characteristics from objects in the target image, and our goal is to harmonize these characteristics to create

a more compelling composite.

At a high level, we begin by building pyramids from the source and target images. We also synthesize a uniform random noise image and build a pyramid from the noise image. Next, we modify the source and noise pyramids to match the target pyramid – a process that harmonizes the images. Finally, we reconstruct the composite from the harmonized source and noise pyramids taking into account the appropriate boundary conditions (both alpha and seamless boundaries). An overview of this process is shown in Fig. 2.2.1. In this section, we provide an overview of our framework and in the sections that follow, we discuss each component in detail.

Our compositing framework uses a multi-resolution pyramid representation for all images. The pyramid is constructed by filtering each image with a set of n linear filters, f_1 to f_n; we use Haar filters. For a source image I^s and target image I^t, the subbands are:

$$\begin{aligned} B_i^s &= f_i \star I^s \\ B_i^t &= f_i \star I^t \,. \end{aligned} \tag{2.1}$$

A standard separable n-level pyramid has three subbands at every level in addition to a lowpass residue subband for a total of $3n+1$ subbands. Each level of the pyramid representation is created by filtering an image with three filters of the same scale. The statistics of pyramid subbands are known to be closely related to image appearance – a property that has been exploited in work on texture synthesis [79, 139]. This makes the pyramid an ideal representation for us, and we harmonize the images by transforming the source subbands in a way that matches their statistics to those of the target subbands.

The main tool for modifying the source subbands in order that their statistics are similar to the target subbands is histogram matching [79]. The harmonized subbands coefficients B_i^h can be computed as

$$B_i^h = \text{histmatch}(B_i^s, B_i^t), \tag{2.2}$$

(a) Inputs (Source / Target) | (b) Naive histogram matching | (c) Smooth histogram matching | (d) Smooth histogram and noise matching | (e) Pyramid compositing with seamless boundaries

Figure 2.3.1: *An image compositing example. The user clones a flat photograph (a,top), onto a high-contrast and textured image (a,bottom). Using naive histogram matching to modify the target subbands produces a result with blotches and haloing near strong edges (b). Using smooth histogram matching improves the result but the noise does not match the target image (c). Injecting noise into the harmonization resolves this (d). Finally, reconstructing the composite from the harmonized pyramid by enforcing seamless boundary conditions produces a highly realistic result (e). Photo credit: Flickr user Steve Wampler/*`Steve Wampler`* (a,top) and Starstock / Photoshot (a,bottom).*

where histmatch() denotes the transfer function that matches the histogram of B_i^s to that of B_i^t.

While the simple operation in Eqn. 2.2 is a powerful tool for matching the appearance of images, there are two fundamental problems with it. First, naive histogram matching is a nonlinear operation that distorts the shape of the subbands, and images reconstructed from these modified subbands often suffer from artifacts such as haloing along strong edges and the amplification of noise and blocking artifacts. For example, Fig. 2.3.1 shows different approaches to transferring the appearance of an older high-contrast and textured photograph to a newer flat and smooth photograph. Fig. 2.3.1(b) is the result of direct histogram matching – the gradients in the original source image have been over-sharpened and there are haloing artifacts near strong edges. Our smooth histogram matching technique – described in Sec. 2.4 – minimizes these artifacts by ensuring that the histogram matching process does not distort the shape of the subbands sub-

stantially (Fig. 2.3.1(c)).

The second problem with a direct application of Eqn. 2.2 relates to image noise. Natural images often have noise due to the camera, such as sensor and ISO noise, or due to compression, such as JPEG quantization noise. In addition, the target images might have textures that are missing in the source images. If the noise and texture patterns in the source and target images differ significantly, histogram matching the subbands alone will not harmonize them. To better model these differences, we introduce a noise term to our harmonization framework. In other words, we assume that the harmonized subbands we want to estimate are given by a sum of the *structure* subbands B_i^h and *noise* subbands N_i^h, i.e.,

$$T_i^h = B_i^h + N_i^h. \tag{2.3}$$

Our intuition is that the structure components B_i^h can be estimated by shaping the source subbands to match the target subbands, while the noise components N^h can be estimated by shaping a noise image to match only the noise in the target subbands. Our harmonization step — covered in detail in Sec. 2.5 — does this iteratively to produce a set of harmonized subband coefficients that exhibit the properties we desire in the source image, including the appropriate contrast, texture, noise and blur (Fig. 2.3.1(d)).

The final harmonized image can be reconstructed from the modified pyramid coefficients T_i^h by collapsing the pyramid, i.e., applying synthesis filters (the inverse of the filters applied in Eqn. 2.1) and summing the results. There are fast and efficient algorithms to do this without explicitly solving the linear system of equations corresponding to Eqn. 2.1. However, to composite regions of the source image into the target image, we need to ensure that boundaries are appropriately handled and simply collapsing the pyramid will not satisfy the desired boundary constraints. Instead, for image compositing, we reconstruct the final composite I^h by solving a linear system of equations:

(a) Source image (b) Naive histogram matching (c) Regularized gain (d) Regularized gain and blending (e) Bae et al. [9]

Figure 2.4.1: *Smooth histogram matching. We would like to give the source image, the tulip photograph from Bae et al. (a), the appearance of Ansel Adams'* Clearing Winter Storm *(see Bae et al. [2006] Fig. 2(a)). Using naive histogram matching produces a result with haloing (b). Regularizing the gain removes these artifacts (c), but some of strong edges have been over-amplified. Blending in the source at these edges removes these problems producing a result (d) with the tones from the model image. The technique of Bae et al. [2006] (e) exaggerates these effects for a more stylized result.*

$$\mathbf{F}I^h = T^h - c\,, \tag{2.4}$$

where the matrix $\mathbf{F}$ contains the filters used to construct the pyramid, the vector T^h contains the harmonized subband coefficients, and the vector c specifies boundary constraints. In Sec. 2.6 we discuss how we set up this linear system and how c can be used to specify both seamless and alpha matting boundary constraints. While this linear system can be very large even for small images, we show how it can solved quickly and accurately using a quadtree subdivision.

2.4 Smooth Histogram Matching

As shown in Figs. 2.3.1 and 2.4.1, applying histogram matching naively on subband coefficients leads to haloing and the amplification of artifacts. Instead, we model histogram matching as a gain control that boosts or reduces subband coefficients depending on their magnitudes, and regularize it to avoid artifacts.

We first match the histograms of the source subbands B_i^s to the histograms of the target subbands B_i^t using Eqn. 2.2. To ensure that we modify the subband coefficient magnitudes without changing their sign, we apply the histogram matching on the absolute values of the coefficients and retain the sign. Matching the histograms produces the modified subbands B_i^{hist}.

The effect of the histogram matching can be modeled as a multiplicative gain that, in logarithmic units, is given as:

$$g_i(|B_i^s|) = \log(|B_i^{hist}|) - \log(|B_i^s|) \; . \tag{2.5}$$

A positive gain indicates an increase in the coefficient magnitude, i.e., the histogram matching enhanced detail in the source image, whereas a negative gain represents a decrease in the coefficient magnitude, i.e., the histogram matching dampened the detail. Up to this point, multiplying the source subband coefficients B_i^s by the gain function $\exp(g_i(|B_i^s|))$ recovers the histogram matched subbands B_i^{hist} perfectly.

In practice, three techniques help mitigate visible artifacts introduced by manipulating subband coefficients. The first is to use undecimated, or oversampled, pyramids; i.e., the subbands of the pyramid are not downsampled after filtering and are the same size as the original image [112]. While pyramids based on any set of linear filters could be used to construct the pyramids, we use oversampled Haar pyramids [75] because of their ease of implementation.

The second method to minimize artifacts is to avoid large values in the gain function and we do this by controlling the maximum gain applied:

$$\hat{G}_i = \exp\left(\frac{\delta_k}{\|g_i\|_\infty} g_i\right) . \tag{2.6}$$

Here δ_k indicates the maximum allowed gain for the subbands at level k and $\|g_i\|_\infty$ denotes the maximum value of g_i. δ_k controls the distortion that will be allowed in the subbands and is set to 1.5.

Finally, the third method to minimize artifacts is to ensure that the gain is spatially smooth

and does not distort the shape of the subbands excessively. As in Li et al. [112], we do not apply the computed gain map directly to the subband coefficients. Instead, at every level of the pyramid k, we compute an activity map that represents local coefficient magnitude by pooling all the rectified subbands (i.e., absolute values of the subband coefficients) at that level and blurring with a Gaussian:

$$\begin{aligned} A_k^s &= N(\sigma) \star \sum_{i \in \text{lev}(k)} |B_i^s| \,, \\ A_k^t &= N(\sigma) \star \sum_{i \in \text{lev}(k)} |B_i^t| \,. \end{aligned} \tag{2.7}$$

The parameter σ controls the width of the Gaussian N and it increases by a factor of two between levels with the value at the finest scale set to 4.

Since the activity maps are blurred, they are spatially smooth. Applying the gain function of Eqn. 2.6 to the activity maps thus produces a gain map $\hat{G}(A_k^s)$ that varies smoothly and does not distort the shape of the subbands excessively. The smooth histogram transfer for subband B_i^s is then given by:

$$B_i^h = m_i \hat{G}(A_k^s) \times B_i^s \,, \tag{2.8}$$

where m_i is a scaling factor related to the level of the pyramid and linearly reduces from 1.0 at the finest scale to 0.45 at the coarsest scale. Eqn. 2.8 describes the function that drives all the histogram matching operations we perform on subbands.

Regularizing the gain eliminates most of the artifacts from naive histogram matching. However, repeatedly manipulating pyramid coefficients in each iteration, might over-amplify strong edges in some cases. To avoid this, we compute an aggregate activity map:

$$A_{ag}^s = \sum_{k=1}^{m} A_k^s, \tag{2.9}$$

and convert it into an alpha map that is clamped to 0 at the 85^{th} percentile and 1 at the 95^{th} percentile, and varies linearly in between. We use this alpha map to blend the harmonized pyramid B^h with the original pyramid B^s. Since the activity maps are highest near strong edges, the blending removes over-amplified edges from the harmonized pyramid (Fig. 2.4.1d).

2.5 Structure and Noise Matching

As mentioned in Sec. 2.3, a composite will fail to look realistic if the noise pattern of the source image does not match the background in the target. We also found that histogram matching cannot successfully create noise to match a target image if the source image is too clean. To better match noise in the composited region, we inject noise into the harmonization process.

Let T_i^s represent the sum of the source subband and the corresponding noise subband, $T_i^s = B_i^s + N_i^s$. Similarly the harmonized subbands we wish to estimate T_i^h are also a sum of structure components and noise components. Following Eqn. 2.8, we construct a gain map $\hat{G}_b$ by matching the histogram of the summed source subbands to the target image.

For the noise subband, we construct a gain map, $\hat{G}_n$, designed specifically to shape the noise. We high-pass filter the target image to isolate the noise image I^n and construct a target noise pyramid N^t. This noise will also contain components of the image structure and cannot be used directly. Instead we assume that the noise components are more prominent in low-activity regions of the target image and we identify these by thresholding the target aggregate activity map as:

$$\Omega = A_{ag}^t < \text{percentile}(A_{ag}^t, \beta). \tag{2.10}$$

A_{ag}^t is computed by applying Eqn. 2.9 to the target image, and β is a user-specified parameter that enables us to differentiate between structure and the noise in the target image. We construct the gain map $\hat{G}_n$ using the process described in Sec. 2.4 by histogram matching the subbands N_i^s to the target noise pyramid subbands N_i^t, but restricted to the low-activity regions.

To summarize, the subband gain map $\hat{G}_b$ is computed by histogram matching the summed subband T_i^s to the target subband B_i^t using the entire compositing region. The noise gain map $\hat{G}_n$ is computed by histogram-matching the subbands N_i^s to the target noise pyramid subbands N_i^t while restricting the pixels to the low-activity region Ω. The structure and noise subbands are then updated as in Eqn. 2.8:

$$B_i^h = \hat{G}_b(A_i^s)\, B_i^s \tag{2.11}$$

$$N_i^h = \hat{G}_n(|N_i^s|)\, N_i^s\,. \tag{2.12}$$

After applying the gains, we collapse the source and noise pyramids to produce the corresponding images and repeat the entire harmonization loop for a fixed number of iterations (set to 5). We refer to this combination of smooth histogram and noise matching as harmonization.

After the final iteration, the harmonized pyramid T^h is given by:

$$T_i^h = B_i^h + N_i^h\,. \tag{2.13}$$

By collapsing this pyramid, we can reconstruct the final output image. If the goal is to composite the harmonized source and target images, we also need to impose the appropriate boundary conditions on the reconstruction. In the next section we describe how we achieve this.

2.6 Pyramid compositing

In the absence of any boundary conditions, the image corresponding to the harmonized subbands T^h is the solution to a linear system that comprises n separate linear systems, each corresponding

to one subband in the harmonized pyramid:

$$\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} I^h = \begin{bmatrix} T_1^h \\ T_2^h \\ \vdots \\ T_n^h \end{bmatrix}, \tag{2.14}$$

where f_i are the filters used to construct the pyramid, T_i^h are the harmonized subbands, and the vector I^h is the final composite.

Alpha matting and gradient-based compositing (also known as seamless cloning) are the two common ways of producing plausible boundaries in composites. While most compositing methods can handle one or the other – Drag and Drop Pasting [93] is a notable exception – in many cases, we would like to have both kinds of boundaries (see Fig. 2.7.5).

In alpha matting the composite is created by blending the foreground image with the background image (in our case the target image I^t) using the alpha matte α_m:

$$I^h = \alpha_m I^f + (1 - \alpha_m) I^t. \tag{2.15}$$

Since harmonized pyramid T^h represents the ideal subband coefficients that we would like our final composite to have, applying the pyramid filters to the composite should reproduce these coefficients, thus $T_i^h = f_i \star I^h$.

Combining this with Eqn. 2.15 gives us the relation

$$\alpha_m f_i \star I^f = T_i^h - (1 - \alpha_m) f_i \star I^t. \tag{2.16}$$

Combining the matting equation with Eqn. 2.14 gives us the relation:

$$\begin{bmatrix} \alpha_m f_1 \\ \alpha_m f_2 \\ \vdots \\ \alpha_m f_n \end{bmatrix} I^f = \begin{bmatrix} T_1^h - (1-\alpha_m) f_1 \star I^t \\ T_2^h - (1-\alpha_m) f_2 \star I^t \\ \vdots \\ T_n^h - (1-\alpha_m) f_n \star I^t \end{bmatrix}. \tag{2.17}$$

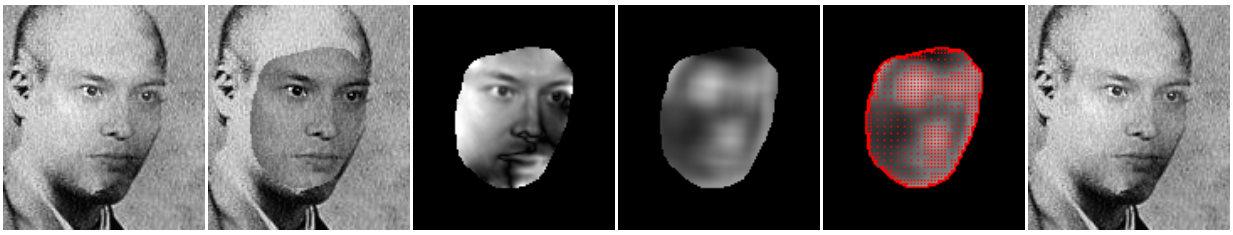
Since both the matte values and the target image are known, we can solve for I^f and compute the final composite I^h by substituting I^f in Eqn. 2.15.

We can incorporate seamless boundaries in Eqn. 2.17 by using the binary compositing mask as the alpha matte. Also, while imposing seamless boundary conditions, we drop the equations corresponding to the coarsest lowpass subband, from Eqn. 2.17. This is similar to gradient domain techniques, where the composite is reconstructed solely from the (highpass) gradients.

To solve Eqn. 2.17 accurately, the subband coefficients T^h need to be consistent with the boundary conditions that we wish to impose. To ensure this, we combine the given alpha matte and seamless region into a single mask that is used to matte the source and target images to create a new image that is now used as the source image. The source subband coefficients T_i^s are computed by decomposing this image, and the harmonization as described in Sec. 2.5 is applied on them. Since the source pyramid is constructed on an image with the correct boundary conditions, the harmonized subband coefficients at the edges will encode these boundary conditions.

2.6.1 Quadtree solver

The size of the linear system we wish to solve in Eqn. 2.17 is quadratic in the number of pixels in the composited region. As a result, as the size of the region increases, solving Eqn. (2.17) directly becomes prohibitively expensive. While this is true even of most gradient based techniques, this effect is amplified in our case because of the larger number of filters we employ.



(a) Least squares (b) Pyramid (c) Difference (d) Activity map (e) Quadtree variables (f) Quadtree solution

Figure 2.6.1: *Quadtree solver. Solving the full system (a) for this example takes* 245 *secs, while collapsing the pyramid (b) takes only* 0.01 *secs. The difference between the two (scaled for visualization) (c) is smooth but contains some image structure. Subdividing the activity map (d) produces a reduced system (e) (the red points correspond to the nodes of the quadtree). Solving this reduced system takes only* 4.875 *secs. and produces a result (f) that is visually indistinguishable from (a).*

Since we have chosen pyramid filters, we can reconstruct an image from the subband coefficients by collapsing the pyramid, i.e., applying synthesis filters to the subbands and summing the result. This pyramid solution I^h_{pyr}, while fast to compute, does not satisfy the boundary constraints. The full least-squares solution I^h_{lsq}, on the other hand, satisfies the boundary constraints, but is slow to compute. The difference between these images, I^h_d, results from satisfying the boundary constraints. As can be seen in Fig. 2.6.1(c), it is smoother than both I^h_{pyr} and I^h_{lsq} and can therefore be well approximated by an upsampled lower resolution image. Thus, instead of solving the full system in Eqn. (2.17), we solve a much smaller system for this difference, upsample it, and add it to the pyramid solution to produce an approximation that is visually identical but much faster to compute. This is similar to Agarwala et al. [3], where, in the context of gradient domain compositing, the difference between a simple color composite and its associated gradient domain composite is efficiently solved for on an adaptively subdivided domain.

The accuracy of this approximation depends on how well the subdivision scheme samples the true difference image. In our case I^h_d still has some of the structure of the original image (Fig. 2.6.1(c)). Therefore, we modify the quadtree subdivision scheme of Agarwala [3] to allocate pixels to regions of high subband coefficient activity as described by the sum of the source activity maps computed in Eqn. 2.8, $\sum_{k=1}^{m} A^s_k$. Starting with the entire compositing region, we recursively

subdivide every block of pixels into four quadrants as long as the sum of the activity in that block is greater than a threshold (set to 4). An example quadtree decomposition is shown in Fig. 2.6.1(e). Note how by basing the quadtree decomposition on the activity map, we are able to sample the difference image fairly well. We solve for I_d^h only at the pixels at the corners of the final quadtree decomposition and the pixels along seam boundaries. At all other pixels we bilinearly interpolate these values.

Let I_{qt} denote the reduced representation for the difference image and let $\mathbf{S}$ denote the interpolation matrix that upsamples I_{qt} to the full size difference image I_d^h:

$$I_d^h = \mathbf{S} I_{qt} \; . \tag{2.18}$$

We rewrite the linear system in Eqn. 2.17 as:

$$\begin{aligned} \mathbf{F}(I_{pyr}^h + I_d) &= \mathbf{T}^h - c \\ (\mathbf{S}^T\mathbf{F}^T\mathbf{F}\mathbf{S})I_d &= \mathbf{S}^T\mathbf{F}^T(\mathbf{T}^h - c - \mathbf{F}I_{pyr}^h) \; . \end{aligned} \tag{2.19}$$

This reduced linear system has reduced memory and time requirements and as shown in Fig. 2.6.1, can be solved efficiently without any differences in visual quality.

2.7 Results and Discussion

Except for Fig. 2.7.4, all the results shown in this chapter were created using a 3-level pyramid. The one parameter in our system that is useful to control the final composite is the noise percentile β in Eqn. 2.10. The noise percentile enables us to distinguish between structure and noise and needs to be set according to how noisy the target image is. We used a value of 25% for all the results except for Figs. 2.7.2 and 2.7.6 where we used 50%.

The run-times for our unoptimized Matlab implementation depend on the size of regions be-

(a) Source Image (b) Our result (c) Bae et al. [9]

Figure 2.7.1: *Style transfer. Using our harmonization framework to transfer the photographic look of Ansel Adams'* Clearing Winter Storm *to the source images (a) produces results (b) with similar effects to the system described by Bae et al. [9] (c).*

ing composited and varied from 15 seconds for the result in Fig. 2.7.7a ($\approx$ 5500 pixels in the composited region) to 12 minutes for the example in Fig. 2.7.5 ($\approx$ 185500 pixels in the composited region). In most cases, almost 85% of the time is spent on solving the reduced version of the linear system in Eqn. 2.17. We used the *CSparse* library [47] to solve the linear system. Recent work on fast sparse solvers [123, 162] and approximate solutions [60] leads us to believe that an optimized implementation of our system can drastically reduce computation times.

Style transfer: With smooth histogram matching on subbands, our harmonization framework is able to achieve effects similar to the style transfer technique described by Bae et al. [9]. Their approach uses a two-level decomposition with nonlinear filters and has separate routines that allow it to exaggerate details. While our goal for harmonization is to improve realism rather than create a stylized result, our results in Figs. 2.4.1 and 2.7.1 suggest that some of these effects are possible within a linear pyramid framework.

(a) Source (b) Target (c) Seamless cloning

(d) Harmonization without noise matching (d) Harmonization with noise matching

Figure 2.7.2: *Matching texture. The sand in the source image (a) has a different texture from that in the target image (b) leading to easily perceivable seams in the seamless cloning result (c). Harmonizing the two image matches the two textures so that the resulting composite (e) is more consistent. This example also illustrates how matching the structure without matching the noise produces unsatisfactory results when the two image have strong texture differences (d). Photo credits: Flickr users* `Scarto` *(a), and* `net_efekt` *(b).*

Contrast matching: The source image in Fig. 2.7.7 has very different contrast from the target faces it has been composited into and the seamlessly cloned composite look unrealistic. By harmonizing the images, our method creates more natural composites.

Texture matching: In both Figs. 2.1.1 and 2.7.2, the target image has a textured appearance that the source does not have. This is especially pronounced in Fig. 2.7.2, where the images are of completely different kinds of sand. While gradient domain compositing produces seamless boundaries, the seam is still easily perceived. By shaping the noise we inject into our system to match

(a) Source / Target (b) Inserting source into target (c) Close-up (d) Inserting target into source (e) Close-up

Figure 2.7.3: *In these examples, the source images (a,top) are smooth while the targets (a,bottom) are noisy. When inserting the source into the target, harmonization adds noise to produce a realistic composite (b). Conversely, when the target image is inserted into the source, harmonization removes most of the noise to match the images (d).*

the textures on the images, we are able to produce more compatible results.

Noise matching: In many cases, the noise characteristics of the source and target images are different. Injecting noise into our framework allows us to reproduce the noise characteristics of the target image and produces a more compelling result. This is illustrated in the examples in Figs. 2.7.3, 2.7.5, and 2.7.7.

While the harmonization framework can add noise to a image to match appearance, an interesting case is the problem of inserting a noisy source image into a smooth target region. This is similar to denoising, which is a long-standing problem in image processing. As seen in Fig. 2.7.3, matching the pyramid subbands decreases the noise and produces a better composite. Intuitively, harmonization suppresses the high frequencies of the noisy source image and automatically se-

(a) Source / Target (b) Harmonized result

Figure 2.7.4: *Matching blur. The region marked in red in the original image (a) is copied and pasted onto the regions marked in green. Cloning the pasted region seamlessly will not match the blur of the original image. Matching the blur produces a result (b) that preserves the shallow depth of field of the original photograph. Photo credit: Flickr user* `patterbt`.

lects the bands to remove frequencies from based on the frequencies in the target image. However, harmonization will not be able to remove all the noise, and often, the final result will be slightly blurred compared to the original.

Color: While our framework was described for grayscale images, it can be easily extended to color. It is important to manipulate color channels in a decorrelated color space so as not to create color shifts and we have found that CIELAB works well. We convert the images to CIELAB space and then harmonize and composite each channel separately. In some cases, the user might like to match the color palette of the source and target images and we use the N-dimensional PDF transfer method of Pitié et al. [136] to match the a and b channels of the source image to those of the target before harmonizing them (Figs. 2.1.1, 2.7.3, and 2.7.5).

Blur: Another scenario in compositing is when the user combines two regions with different blur. This is illustrated in Fig. 2.7.4 where the user segments a sharp object and clones it onto a blurred region expecting the inserted object to have the same defocus properties as the source. By harmonizing the inserted object with the defocused objects it is replacing, we are able to produce an image with realistic blur. We used a 4-level pyramid to generate this example because of the large amount of blur.

Mixed boundary constraints: One of the advantages of pyramid compositing is the ability to incorporate boundary conditions for both alpha matting and seamless cloning. This is illustrated by Figs. 2.7.5 and 2.7.6, where the final composite has seamless boundaries in some parts (the road and the sand) and alpha matte based boundaries elsewhere (the car and the hydrant).

Limitations: Like Heeger-Bergen texture synthesis, our noise and texture matching technique makes the assumption that the target noise and texture can be matched by shaping the subbands of the noise image. Such techniques are known to work well on stochastic textures but do not reproduce every texture pattern accurately. In particular, it is known that histogram matching of pyramid subbands cannot be used to create textures that are correlated across scales [139]. Therefore, in some cases there might be differences in the noise between the target and harmonized images. For example, the harmonized image in Fig. 2.1.1 does not capture the small cracks in the painting and the result in Fig. 2.7.6 does not replicate the structure of the sand. In spite of this, harmonization leads to a substantial improvement in the realism of the composite, and in most cases, it is difficult to see the differences without looking at the original target image.

Also, a fundamental assumption of our approach is that matching the statistics of the source and target images will harmonize them. This may not always be the case, especially in situations where the objects being matched are completely different. This is illustrated in Fig. 2.7.6, where matching the images does not produce the right colors and leads to excessive noise on the fore-

(a) Source (b) Target

(c) Harmonized result

Figure 2.7.5: *Compositing with mixed boundary conditions. In this example, the user clones a Porsche (a) into an old photograph of a Ferrari (b). Our result (c) matches the noise on the images, and alpha mattes the car while enforcing seamless boundaries on the road at the bottom. Photo credits: Flickr users* `teliko82` *(a), and* `prorallypix` *(b).*

(a) Source / Target (b) Harmonized result

Figure 2.7.6: *Limitations. A hydrant in snow ((a) top) has been composited into sand ((a) bottom). Harmonization matches the snow to the sand, and compositing with mixed boundary conditions produces seamless boundaries along the sand and matting along the hydrant. However, the texture generated is not able to match the structure of the original sand. Also, because the target image does not have shadows or a hydrant, harmonization is not able to produce realistic shadows and has added excessive noise on the hydrant. Photo credits: Flickr users* `Bob.Fornal` *(a,top) and* `lrargerich` *(a,bottom).*

ground object.

This can be solved by matching the appearance of different parts of the target image in slightly different ways. We are looking at ways of determining this automatically and applying the harmonization while ensuring we do not introduce artifacts.

2.8 Summary

In this chapter, we used a statistical model for image appearance that used the histograms of a multi-scale decomposition to represent different aspects of appearance such as global and local

contrast, texture, noise, and blur. Based on this representation, we have presented a framework that harmonizes the appearance of images before compositing them. By automatically matching different aspects of visual appearance, our technique takes the burden of correcting for them away from the user. We have also presented a novel compositing scheme that allows us to enforce both matte-based and seamless boundaries in the same framework.

There are other aspects of visual appearance that are important to the realism of a composite that our work does not address. The most important of these are shadows and shading. Automatically estimating and correcting the lighting in single images is a difficult vision problem and is an interesting avenue for future work.

The ability to realistically combine multiple images is important in many vision and graphics applications such as image mosaicing and digital photomontage, and we would like to apply our methods in their context too. One particularly interesting scenario is the problem of video object insertion. In the next chapter, we will explore one instance of this problem — replacing facial performances in videos. In particular, we demonstrate a system that uses models for face geometry to track, align, and subsequently, composite faces in videos.

(a) Target images | (b) Seamless cloning | (c) Naive histogram matching | (d) Smooth histogram matching | (e) Smooth histogram and noise matching

Figure 2.7.7: *Matching contrast and noise. Our method adapts the same source image to match target images (a) with different contrast and noise. Gradient domain compositing (b) produces unrealistic results because of the discrepancies between the images being combined. Naive histogram matching (c) results in over-sharpening and haloing artifacts. Smooth histogram matching method (d) removes these artifacts, but the noise is inconsistent. Matching both the structure and the noise removes these inconsistencies and produces photo-realistic results (e). Photo credits: Flickr users* `Okinawa Soba` *(second row),* `zsoltika` *(third row), and* `freeparking` *(fourth and fifth rows).*

3

Editing Faces in Videos

In Chapter 2, we developed a technique to transfer appearance across images and, used it to blend disparate images and create photo-realistic composites. In this chapter we discuss an extension of the ideas in that work, to the problem of video compositing. Video compositing is significantly harder because of the spatial and temporal dynamics inherent to video sequences. For example, in chapter 2, we assumed that the input images were geometrically aligned by the user, but doing this for every frame of a video is very tedious. Instead, in this chapter, we propose techniques to automatically transfer *both* the geometry and appearance of a video to a different sequence. In particular, we focus on the problem of automatic video face

(a) Source (b) Target (c) Aligned (d) Three frames of the blended result

Figure 3.1.1: *Video face replacement. Our method for face replacement requires only single-camera video of the source (a) and target (b) subject, which allows for simple acquisition and reuse of existing footage. We track both performances with a multilinear morphable model then spatially and temporally align the source face to the target footage (c). We then compute an optimal seam for gradient domain compositing that minimizes bleeding and flickering in the final result (d).*

replacement.

The work in this chapter was done in collaboration with other researchers. While the entire work has been reproduced below for the sake of completeness, this dissertation's author's primary technical contributions are described in Sec. 3.6.

3.1 Introduction

Techniques for manipulating and replacing faces in photographs have matured to the point that realistic results can be obtained with minimal user input (e.g., [4, 22, 160]). Face replacement in video, however, poses significant challenges due to the complex facial geometry as well as our perceptual sensitivity to both the static and dynamic elements of faces. As a result, current systems require complex hardware and significant user intervention to achieve a sufficient level of realism (e.g., Alexander et al. [6]).

This chapter presents a method for face replacement in video that achieves high-quality results using a simple acquisition process. Unlike previous work, our approach assumes inexpensive hardware and requires minimal user intervention. Using a single camera and simple illumination, we capture *source* video that will be inserted into a *target* video (Fig. 3.1.1). We track the face in both the source and target videos using a 3-d multilinear model. Then we warp the source video in

both space and time to align it to the target. Finally, we blend the videos by computing an optimal spatio-temporal seam and a novel mesh-centric gradient domain blending technique.

Our system replaces all or part of the face in the target video with that from the source video. Source and target can have the same person or two different subjects. They can contain similar performances or two very different performances. And either the source or the target can be existing (i.e., uncontrolled) footage, as long as the face poses (i.e., rotation and translation) are approximately the same. This leads to a handful of unique and useful scenarios in film and video editing where video face replacement can be applied.

For example, it is common for multiple takes of the same scene to be shot in close succession during a television or movie shoot. While the timing of performances across takes is very similar, subtle variations in the actor's inflection or expression distinguish one take from the other. Instead of choosing the single best take for the final cut, our system can combine, e.g., the mouth performance from one take and the eyes, brow, and expressions from another to produce a *video montage*.

A related scenario is *dubbing*, where the source and target subject are the same, and the source video depicts an actor in a studio recording a foreign language track for the target footage shot on location. The resulting video face replacement can be far superior to the common approach of replacing the audio track only. In contrast to multi-take video montage, the timing of the dubbing source is completely different and the target face is typically fully replaced, although partial replacement of just the mouth performance is possible, too.

Another useful scenario involves *retargeting* existing footage to produce a sequence that combines an existing backdrop with a new face or places an existing actor's facial performance into new footage. Here the new footage is shot using the old footage as an audiovisual guide such that the timing of the performances roughly matches. Our video-based method is particularly suitable in this case because we have no control over the capture of the existing footage.

A final scenario is *replacement*, where the target facial performance is replaced with an arbitrary

source performance by a different subject. This is useful, for example, when replacing a stunt actor's face, captured in a dangerous environment, with the star actor's face, recorded in a safe studio setting. In contrast to retargeting, where the source footage is shot using the target as an audiovisual guide to roughly match the timings, the performance of the source and target can be very different, similar to dubbing but with different subjects.

Furthermore, it is entertaining for amateurs to put faces of friends and family into popular movies or music videos. Indeed, an active community of users on YouTube has formed to share such videos despite the current manual process of creating them (e.g., search for "Obama Dance Off"). Our video face replacement system would certainly benefit these users by dramatically simplifying the currently labor-intensive process of making these videos.

Video face replacement has advantages over replacing the entire body or the head in video. Full body replacement typically requires chroma key compositing (i.e., green screening) or rotoscoping to separate the body from the video. Head replacement is difficult due to the complexities of determining an appropriate matte in regions containing hair. Existing methods for both body and head replacement require expensive equipment, significant manual work, or both [6]. Such methods are not practical in an amateur setting and are also time consuming and challenging for professionals.

Our system does rely on a few assumptions about the input videos. It works best when the illumination in the source and target videos is similar. However, we mitigate this limitation by finding a coherent spatio-temporal seam for blending that minimizes the differences between the source and target videos (Sec. 3.6). Second, we assume that the pose of faces in the source and target videos is $\pm 45^o$ from frontal, otherwise automatic tracking and alignment of the faces will fail (Sec. 3.4). This assumption could be waived by employing user assistance during tracking.

The main contribution of this work is a new system for video face replacement that does not require expensive equipment or significant user intervention. We developed a novel spatio-temporal seam finding technique that works on meshes for optimal coherent blending results.

We demonstrate the applicability of our approach on a number of examples in four scenarios: video montage (Fig. 3.7.1), dubbing (Fig. 3.7.2), retargeting (Figs. 3.1.1 and 3.7.4), and replacement (Fig. 3.7.3). We present results of a user study on Mechanical Turk that demonstrates that our system is sufficient for plausible face replacement and difficult to distinguish from real footage (Sec. 3.7).

3.2 Related Work

Face replacement in images and video has been considered in a variety of scenarios, including animation, expression transfer, and online privacy. However, the direct video-to-video face transfer presented in this chapter has been relatively unexplored. We briefly describe previous work on face replacement and compare these approaches to our system.

3.2.1 Editing faces in images

Face editing and replacement in images has been a subject of an extensive research. For example, the method by Blanz et al. [24] fits a morphable model to faces in both the source and target images and renders the source face with the parameters estimated from the target image. The well-known photomontage [4] and instant cloning systems [60] allow for replacing faces in photographs using seamless blending [131]. Bitouk et al. [22] describe a system for automatic face swapping using a large database of faces. They use this system to conceal the identity of the face in the target image. Face images have been also used as priors to enhance face attractiveness using global face warping [110] or to adjust tone, sharpness, and lighting of faces [96]. The system of Sunkavalli et al. [160] models the texture, noise, contrast and blur of the target face to improve the appearance of the composite. More recently, Yang et al. [187] use optical flow to replace face expressions between two photographs. The flow is derived from 3-d morphable models that are fit to the source and target photos. It is not clear whether any of these methods could achieve

temporally coherent results when applied to a video sequence.

3.2.2 Face replacement in video using 3-d models

The traditional way to replace faces in video is to acquire a 3-d face model of the actor, to animate the face, and to relight, render, and composite the animated model into the source footage. The 3-d face model of the actor can be captured using marker-based [21, 76, 181], structured light [111, 119, 179, 191], or passive multi-view stereo approaches [17, 29, 94]. Model-based face replacement can achieve remarkable realism. Notable examples include the recreation of actors for The Matrix Reloaded [25], The Curious Case of Benjamin Button [144], and the Digital Emily project [6]. However, these methods are expensive, and typically require complex hardware and significant user intervention to achieve a sufficient level of realism.

3.2.3 Video-to-video face replacement

Purely image-based methods do not construct a 3-d model of the actor. Bregler et al. [30] and Ezzat et al. [58] replace the mouth region in video to match phonemes of novel audio input using a database of training images of the same actor. Flagg et al. [68] use video-textures to synthesize plausible articulated body motion. Kemelmacher-Shlizerman et al. [100] make use of image collections and videos of celebrities available online and replace face photos in real-time based on expression and pose similarity. However, none of these methods are able to synthesize the subtleties of the facial performance of an actor.

3.2.4 Morphable models for face synthesis

Closely related to our work are image-based face capture methods [23, 50, 56, 135, 177]. These approaches build a morphable 3-d face model from source images without markers or special face scanning equipment. We use the multilinear model by Vlasic at al. [177] that captures identity,

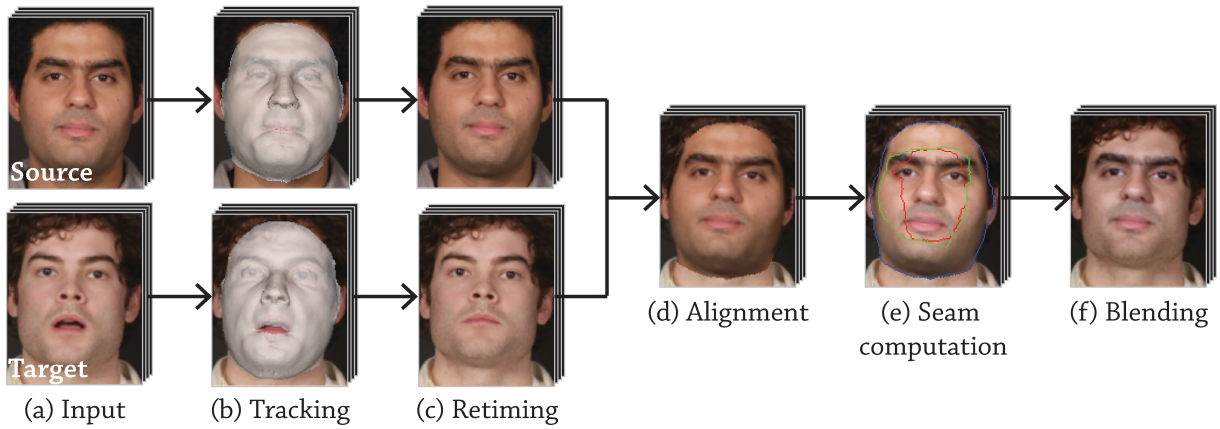


Figure 3.2.1: *An overview of our method. (a) Existing footage or single camera video serves as input source and target videos. (b) Both sequences are tracked and (c) optionally retimed to temporally align the performances. (d) The source face is spatially aligned in the target video. (e) An optimal seam is computed through the target video to minimize blending artifacts, and (f) the final composite is created with gradient-domain blending.*

expression, and visemes in the source and target videos. Existing approaches use the estimated model parameters to generate and drive a detailed 3-d textured face mesh for a target identity, which can be seamlessly rendered back into target footage. In general, these systems assume the source actor's performance, but not their face, is desired in the newly synthesized output video. In contrast, our approach blends the source actor's complete face and performance, with all of its nuances intact, into the target.

3.3 Overview

Figure 3.2.1 shows an overview of our method. In order to replace a source face with a target face, we first model and track facial performances of both source and target with the multilinear method and data of Vlasic et al. [177]. Their method estimates a multilinear model from 3-d face scans of different identities, expressions, and speech articulations (i.e., visemes). It tracks parameters for these attributes and the 3-d pose of the face (given as a rotation, translation, and scale) over a video sequence. At each frame, the pose, the multilinear model, and its parameters

can be used to generate a 3-d mesh that matches the geometry of the subject's face. A sufficient approximate fit is obtainable even for new faces that are not present in the original dataset. We reprocessed the original training data from Vlasic et al. covering 16 identities $\times$ 5 expressions $\times$ 5 visemes–a total of 400 face scans–placing them into correspondence with a face mesh that extends beyond the jaw and chin regions (Sec. 3.7).

In some scenarios it is important that the timing of the facial performance matches precisely in the source and the target footage. However, it might be very tedious to match these timings exactly as demonstrated by the numerous takes that are typically necessary to obtain compelling voiceovers (e.g., when re-recording a dialog for a film.) Instead, we only require a coarse synchronization between source and target videos and automatically retime the footage to generate a precise match for the replacement.

After tracking and retiming, we blend the source performance into the target video to produce the final result. This blending makes use of gradient-domain compositing to merge the source actor's face into the target video. While gradient domain compositing can produce realistic seamless results, the quality of the composite is often tied to the seam along which the blend is computed. Using an arbitrary seam is known to lead to bleeding artifacts. To minimize these artifacts we automatically compute an optimal spatio-temporal seam through the source and target that minimizes the difference across the seam on the face mesh and ensure that the regions being combined are compatible. In the second stage we use this seam to merge the gradients and recover the final composite video. For the results shown in this chapter, each of which is about 10 seconds, processing requires about 20 minutes.

3.4 Face Tracking

3.4.1 Input

Footage for all examples, except those that reuse existing footage, was captured with a Canon T2i camera with 85 mm and 50 mm lenses at 30 frames per second. In-lab sequences were lit with 300 W studio lights placed on the left and right and in front of the subject, softened by umbrella reflectors. When appropriate, we used the target video as an audio-visual guide during capture of the source (or vice versa) to approximately match timing. All such examples in this chapter were captured in 1-2 takes. For pose, actors were simply instructed to face the camera; natural head motion is accounted for with tracking.

3.4.2 Tracking

To track a face across a sequence of frames, the method of Vlasic et al. [177] computes the pose and attribute parameters of the multilinear face model that best explain the optical flow between adjacent frames in the sequence. The multilinear face model $\mathscr{M}$, an N-mode tensor with a total of $3K \times D_2 \times \ldots \times D_N$ elements (where K is the number of vertices in a single face mesh), is obtained via N-mode singular value decomposition (N-mode SVD) from the N-mode data tensor containing the vertex positions of the original scan data (the Cartesian product over expression, viseme, and identity).

With the multilinear model in hand, the original face data can be interpolated or extrapolated to generate a new face as

$$\mathbf{f} = \mathscr{M} \times_2 \mathbf{w}_2^\top \times_3 \mathbf{w}_3^\top \times_4 \mathbf{w}_4^\top, \tag{3.1}$$

where mode 1 corresponds to vertex positions in the 4-mode model, $\mathbf{w}_i$ is a $D_i \times 1$ column vector of parameters for the attribute corresponding to the i^{th} mode (i.e., one of expression, viseme, or identity), $\mathbf{f}$ is a $3K$-element column vector of new vertex positions, and the $\times_n$ operator is the

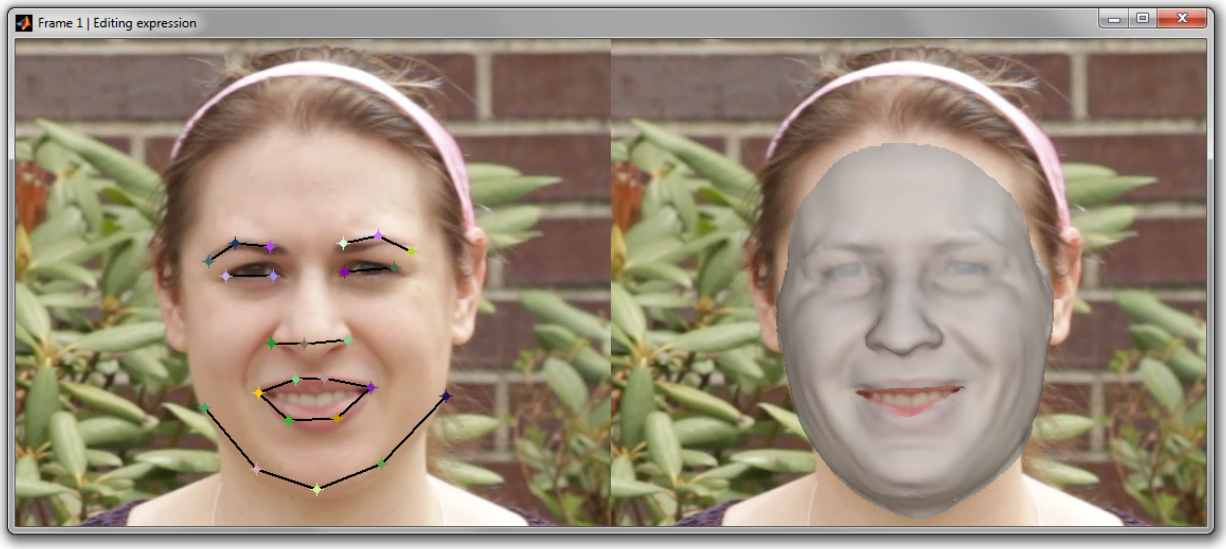


Figure 3.4.1: *User interface for tracking. To refine the initialization or correct tracking at a specific key frame, the user can adjust a few markers on the face to adjust pose, expression, or viseme.*

mode-n product, defined between a tensor and a matrix. We refer the reader to Vlasic et al. [177] for more details.

3.4.3 Initialization

Since tracking is based on optical flow, initialization is critical, as errors in the initialization will be propagated throughout the sequence. Moreover, tracking can go astray on troublesome frames, e.g., due to motion blur, extreme pose change, high frequency lighting, or occlusions. Therefore, we also provide a simple user interface that can ensure good initialization and can correct tracking for troublesome frames.

The interface allows the user to adjust positions of markers on the eyes, eyebrows, nose, mouth, and jawline, from which the best-fit pose and model parameters are computed. The user can alternate between adjusting pose and each attribute individually; typically, 1 iteration of each is sufficient for good initialization (Fig. 3.4.1).

We start by automatically detecting the face [176]. Next, we localize facial features [57] (e.g.,

the corners of the mouth, eyes, and nose) in the first frame of a sequence. Then, we compute the initial pose that best aligns the detected features with the corresponding source features in the face mesh. This initial face mesh is generated from the multilinear model using a user-specified set of initial attributes corresponding to the most appropriate expression, viseme, and identity.

Holding all but one attribute's parameters fixed, we can project the multilinear model $\mathcal{M}$ onto the subspace corresponding to the remaining attribute, e.g., for the third attribute:

$$\mathbf{A}_3 = \mathcal{M} \times_2 \mathbf{w}_2^\top \times_4 \mathbf{w}_4^\top, \tag{3.2}$$

for the $3K \times D_3$ matrix $\mathbf{A}_3$. Given $\mathbf{A}_i$ and a column vector $\mathbf{g}$ of target vertex positions, we can compute parameters for the i^{th} attribute that best fit the target geometry as

$$\underset{\mathbf{w}_i}{\operatorname{argmin}} \|\mathbf{g} - \mathbf{A}_i \mathbf{w}_i\|^2. \tag{3.3}$$

The least squares solution to Eqn. 3.3 is given as

$$\mathbf{w}_i = (\mathbf{A}_i^\top \mathbf{A}_i)^{-1} \mathbf{A}_i^\top \mathbf{g}. \tag{3.4}$$

To fit parameters for the i^{th} attribute to image space markers, we take the subset of the multilinear model corresponding to the (x, y) coordinates of mesh vertices that should align to the markers and apply Eqn. 3.4, populating $\mathbf{g}$ with marker positions, transformed to the coordinate frame of the model via an inverse pose transformation.

While multilinear tracking does well at tracking expression and viseme, which vary from frame to frame, we found that identity, which is computed over the full sequence and held constant, was not. Even after multiple iterations of tracking, each of which updates identity parameters, those parameters changed very little from their initial values. This caused significant problems when tracking with a full face model, where it is critical that the mesh covers the subject's entire face,

and only their face (no background) over the entire sequence. Therefore it is important to have an accurate initialization of identity.

We employ the FaceGen Modeller [156] in order to obtain a better initialization of the identity parameters. FaceGen generates a 3-d mesh based on a frontal face image and, optionally, a profile image. The input images can be extracted from the original video sequences or downloaded from the Internet when reusing existing footage. The input images need to depict the subject with a closed-mouth neutral expression. FaceGen requires minimal user input to specify about 10 markers per image. All meshes created by FaceGen are themselves in correspondence. Therefore, we can register the FaceGen mesh with the multilinear model using the same template-fitting procedure [177] we used to register the original scan data. We then fit the multilinear model to the registered FaceGen mesh using Procrustes alignments to our current best-fit mesh and using Eqs. 3.3 and 3.4 to solve for the best-fit identity parameters. In this optimization we only use about 1 percent of the original mesh vertices. The process typically converges in 10 iterations.

3.4.4 Key framing

We can use the same interface (Fig. 3.4.1) for adjusting pose and attribute parameters at specific key frames where automatic tracking fails. First, we track the subsequences between each pair of user-adjusted key frames in both the forward and reverse directions and linearly interpolate the two results. We then perform additional tracking iterations on the full sequence to refine pose and parameter estimates across key frame boundaries. Note that none of the results shown in the chapter required key framing.

3.5 Spatial and Temporal Alignment

3.5.1 Spatial alignment

From an image sequence I, where $I(x,t)$ denotes the value at pixel position x in frame t, tracking produces a sequence of attribute parameters and pose transformations. For each frame t, $\mathbf{f}(t)$ is the column vector of vertex positions computed from attribute parameters at time t using Eqn. 3.1, and $\mathbf{f}_i(t)$, the i^{th} vertex at time t. Per-frame pose consists of a scale s, 3×3 rotation matrix $\mathbf{R}$, and a translation vector $\mathbf{t}$ that together transform the face meshes into their tracked positions in image space coordinates. Subscripts S and T denote source and target, respectively.

To align the source face in the target frame, we use the face geometry from the source sequence and pose from the target sequence. That is, for frame t, the aligned position of the i^{th} source vertex position is given as

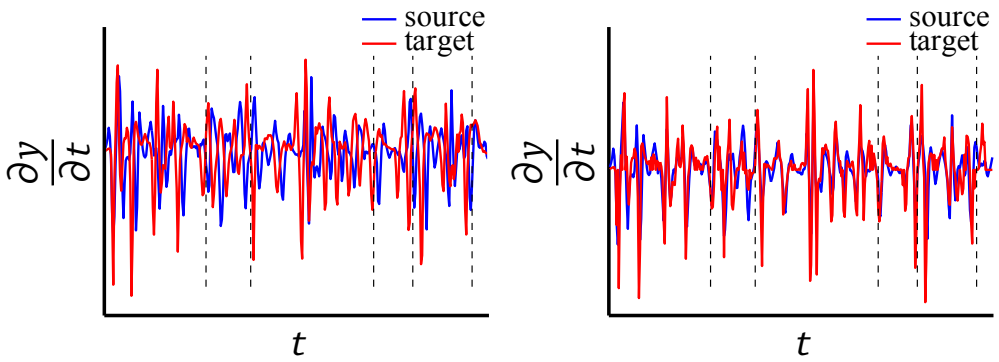
$$\mathbf{f}'_{i,S}(t) = s_T(t)\mathbf{R}_T(t)\mathbf{f}_{i,S}(t) + \mathbf{t}_T(t) \tag{3.5}$$

We also take texture from the source image I_S; texture coordinates are computed similarly to Eqn. 3.5 using instead both source geometry and source pose.

While we track the full face mesh in both source and target sequences, the user may choose to replace only part of the target face, for example, in the multi-take video montage result in Fig. 3.7.1. In this case, the user either selects from a predefined set of masks – eyes, eyes and nose, or mouth – or paints an arbitrary mask on the face. In these cases, $\mathbf{f}'_S$ represents only those vertices within the user-specified mask.

3.5.2 Retiming

We retime the footage using Dynamic Time Warping (DTW) [141]. DTW is a dynamic programming algorithm that seeks a monotonic mapping between two sequences that minimizes the to-



(a) Lip motion before retiming (b) Lip motion after retiming

(c) Target frames before retiming

(d) Source frames

(e) Target frames after retiming

Figure 3.5.1: *Video retiming. Motion of the center vertex of the lower lip for source and target before retiming (a) and after (b). Corresponding cropped frames from the target before retiming (c), the source (d), and the target after retiming (e).*

tal cost of pairwise mappings. The output of DTW provides a reordering of one sequence to best match the other. Here we define pairwise cost between source and target frames according to the motion of the mouth in each frame. We found that computing cost based on motion instead of absolute position was more robust across differences in mouth shape and articulation in different subjects.

Specifically, for the loop of vertices along the interior of the upper and lower lip, we compare the average minimum Euclidean distance between the first partial derivatives with respect to time. Comparing velocity of mouth vertices for this step, as opposed to position, ensures robustness to differences in mouth shape between source and target. We compute these partial derivatives using first order differencing on the original vertex positions without transforming to image space. Let $\mathbf{m}_{i,S}(t_1)$ and $\mathbf{m}_{j,T}(t_2)$ be the partial derivatives for the i^{th} vertex in the source mouth at time t_1 and the j^{th} vertex in the target mouth at time t_2 , respectively. Then the cost of mapping source frame t_1 to target frame t_2 for DTW is

$$\sum_i \min_j ||\mathbf{m}_{i,S}(t_1) - \mathbf{m}_{j,T}(t_2)|| + \min_j ||\mathbf{m}_{j,S}(t_1) - \mathbf{m}_{i,T}(t_2)||. \tag{3.6}$$

DTW does not consider temporal continuity. The resulting mapping may include 'stairstepping', where a given frame is repeated multiple times, followed by a non-consecutive frame, which appears unnatural in the retimed video. We smooth the mapping with a low-pass filter and round the result to the nearest integer frame. This maintains sufficient synchronization while removing discontinuities. While there are more sophisticated methods that can directly enforce continuity e.g., Hidden Markov Models (HMMs), as well as those for temporal resampling, we found this approach to be fast and well-suited to our input data, where timing is already fairly close.

Since the timing of the original source and target videos is already close, the mapping can be applied from source to target and vice versa (for example, to maintain important motion in the background of the target or to capture the subtle timing of the source actor's performance.) For

simplicity, in the following sections $\mathbf{f}_S(t)$ and $\mathbf{f}_T(t)$, as well as their corresponding texture coordinates and texture data, refer to the retimed sequences when retiming is employed and to the original sequences when it is not. Fig. 3.5.1 highlights the result of retiming inputs with dialog with DTW.

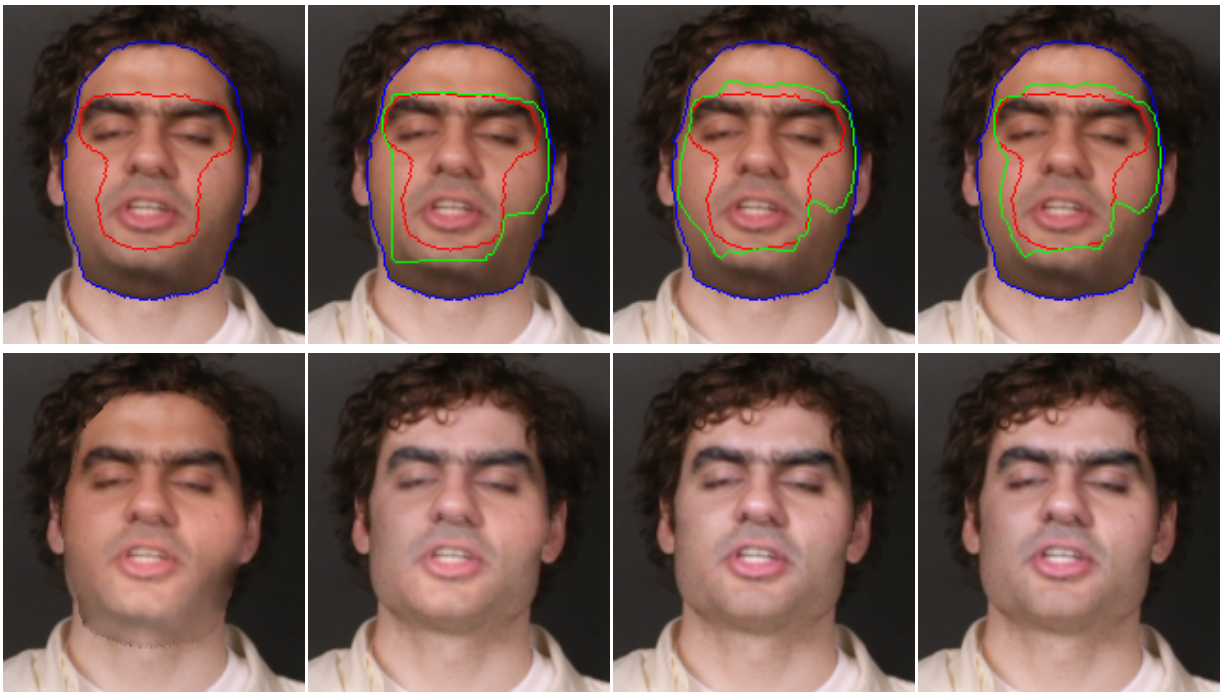
3.6 Blending

3.6.1 Optimal seam finding

Having aligned the source face texture to the target face, we would like to create a truly photo-realistic composite by blending the two together. While this can be accomplishing using gradient-domain fusion [131], we need to specify the region from the aligned video that needs to be blended into the target video, or alternatively, the *seam* that demarcates the region in the composite that comes from the target video from the region that comes from the aligned video. While the edge of face mesh could be used as the seam, in many cases it cuts across features in the video leading to artifacts such as bleeding (see Fig. 3.6.1). In addition, this seam needs to be specified in every frame of the composite video, making it very tedious for the user to do.

We solve this problem by automatically estimating a seam in space-time that minimizes the differences between the aligned and target videos, thereby avoiding bleeding artifacts. While a similar issue has been addressed in previous work [4, 93, 102], our problem has two important differences. First, the faces we are blending often undergo large (rigid and non-rigid) transformations, and the seam computation needs to be handle this. Second, it is important that the seam be temporally coherent to ensure that the composited region does not change substantially from frame to frame leading to flickering artifacts (see Fig. 3.6.1).

Our algorithm incorporates these requirements in a novel graph-cut framework that estimates the optimal seam *on the face mesh*. For every frame in the video, we compute a closed polygon on



(a) No seam computation | (b) Image-space seam computation | (c) Mesh-space seam computation | (d) Mesh-space seam computation with temporal coherence

Figure 3.6.1: *Seam computation for blending. The face mask boundary (blue), user-specified region to be preserved (red), and the optimal seam (green) are marked in each source frame. (a) Directly blending the source and target produces results with strong bleeding artifacts. (b) Computing a seam in image space improves results substantially but does not vary as pose and expression change. (c) A seam computed on the mesh can track these variations but may lead to flickering artifacts (see accompanying video) without additional constraints. (d) Enforcing temporal coherence minimizes these artifacts.*

the face mesh that separates the source region from the target region; projecting this polygon onto the frame gives us the corresponding *image-space* seam. Estimating the seam in *mesh-space* helps us handle our two requirements. First, when the face deforms in the source and target videos, the face mesh deforms to track it without any changes in its topology. The mesh already accounts for these deformations, making the seam computation invariant to these changes. For example, when a subject talks, the vertices corresponding to his lips remain the same, while their positions change. Thus, a polygon corresponding to these vertices defines a time-varying seam that stays true to the motion of the mouth. Second, estimating the seam on the mesh allows us to enforce temporal constraints that encourage the seam to pass through the same vertices over time. Since the face vertices track the same face features over time this means that same parts of the face are preserved from the source video in every frame.

We formulate the optimal seam computation as a problem of labeling the vertices of the face mesh as belonging to the source or target video. We do this by constructing a graph on the basis of the face mesh and computing the min-cut of this graph. The nodes of this graph correspond to the vertices in the face aligned mesh over time (i.e., $\mathbf{f}_i(t) \forall i, t$). The edges in the graph consist of spatial edges corresponding to the edges in the mesh (i.e., all the edges between a vertex $\mathbf{f}_i(t)$ and its neighbor $\mathbf{f}_j(t)$) as well as temporal edges between corresponding vertices from frame to frame (i.e., between $\mathbf{f}_i(t)$ and $\mathbf{f}_i(t+1)$).

Similar to previous work on graphcut textures [102] and photomontage [4], we want the seam to cut through edges where the differences between the source and target video frames are minimal. This is done by setting the weights on the spatial edges in the graph between neighboring vertices $\mathbf{f}_i(t)$ and $\mathbf{f}_j(t)$ as:

$$\begin{aligned} W_s(\mathbf{f}_i(t), \mathbf{f}_j(t)) &= ||I_S(\mathbf{f}_i(t), t) - I_T(\mathbf{f}_i(t), t)|| \\ &\quad + ||I_S(\mathbf{f}_j(t), t) - I_T(\mathbf{f}_j(t), t)|| \end{aligned} \tag{3.7}$$

When both the source and the target videos have very similar pixel values at vertices $\mathbf{f}_i(t)$ and $\mathbf{f}_j(t)$, the corresponding weight term takes on a very small value. This makes it favorable for a min-cut to cut across this this edge.

We would also like the seam to stay temporally coherent to ensure that the final composite does not flicker. We ensure this by setting the weights for the temporal edges of the graph as follows:

$$\begin{aligned} W_t(\mathbf{f}_i(t), \mathbf{f}_i(t+1)) &= W(\mathbf{f}_i(t+1), \mathbf{f}_i(t)) \\ &= \lambda(||I_S(\mathbf{f}_i(t), t) - I_S(\mathbf{f}_i(t), t+1)||^{-1} \\ &\quad +||I_T(\mathbf{f}_i, t) - I_T\mathbf{f}_i, t+1)||^{-1}), \end{aligned} \tag{3.8}$$

where λ is used to control the influence of the temporal coherence. Unlike the spatial weights, these weights are constructed to have high values when the appearance of the vertices does not change much over time. If the appearance of vertex $\mathbf{f}_i(t)$ does not change over time in either the source or target video, this weight term takes on a large value, thus making it unlikely that the min-cut would pass through this edge, thus ensuring that this vertex has the same label over time. However, if the appearance of the vertex does change (due to the appearance of features such as hair, eyebrows, etc.), the temporal weight drops. This makes the seam temporally coherent while retaining the ability to shift to avoid features that cause large differences in intensity values. In practice, we set λ as the ratio of the sum of the spatial and temporal weights, i.e., $\lambda = \sum_{i,j,t} W_s(\mathbf{f}_i(t), \mathbf{f}_j(t), t) / \sum_{i,j,t} W_t(\mathbf{f}_i(t), \mathbf{f}_i(t+1))$. This ensures that the spatial and temporal terms are weighted approximately equally.

The vertices on the boundary of the face mesh in every frame are labeled as target vertices as they definitely come from the target videos. Similarly, a small set of vertices in the interior of the mesh are labeled as source vertices. This set can be directly specified by the user in one single frame.

Having constructed this graph, we use the alpha-expansion algorithm [28] to label the mesh

vertices as belonging to the either the source or target videos. The construction of the graph ensures that, in every frame, the graph-cut seam forms a closed polygon that separates the target vertices from the source vertices. From these labels we can explicitly compute this closed polygon $\partial P(t) = \{p_0(t), p_1(t), \cdots, p_{m_t}(t)\}$ for every frame. In addition, we also project these labels onto the frames to compute the corresponding image-space mask for compositing.

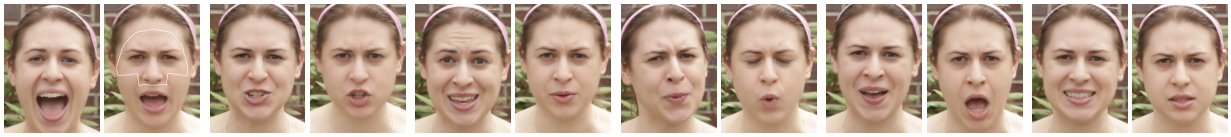
Fig. 3.6.1 shows the results of estimating the seam using our technique on an example video sequence. As can be seen in this example, using the edge of the face mesh as the seam leads to strong bleeding artifacts. Computing an optimal seam ensures that these artifacts don't occur. However, without temporal coherence, the optimal seam "jumps" from frame to frame, leading to flickering in the video. By computing the seam on the mesh using our combination of spatial and temporal weights we are able to produce a realistic composite that stays coherent over time. Please see the accompanying video to observe these effects.

3.6.2 Compositing

Having estimated the optimal seam for compositing, we blend the source and target videos using gradient-domain fusion. We do this using a recently proposed technique that uses mean value coordinates [60] to interpolate the differences between the source and target frames along the boundary. We re-use the face mesh to interpolate these differences. In particular, for every frame of the video, we compute the differences between source and target frames along the seam $\partial P(t)$, and interpolate them at the remaining source vertices using mean value coordinates. These differences are then projected onto the image and added to the source video to compute the final blended composite video.

(a) Source and target frame pairs

(b) Blending result



(c) Source and target frame pairs

(d) Blending result

Figure 3.7.1: *Multi-take video montages. (top) Two handheld takes of the same dialog and (bottom) two handheld takes of poetry recitation. (a,c) Retimed source and target frames (left and right, respectively) with the region to be replaced marked in the first target frame. (b,d) Frames from the blended result that combine the target pose, background, and mouth with the source eyes and expression.*

(a) Source and target frame pairs

(b) Blending result

Figure 3.7.2: *Dubbing using face replacement. (a) Cropped source and target frames (left and right, respectively) from an indoor recording of dialog in English and an outdoor recording in Hindi, respectively. (b) Frames from the blended result. Note how the differences in lighting and mouth/chin position between source and target are seamlessly combined in the result.*

3.7 Results and Discussion

3.7.1 Results

We show results for a number of different subjects, capture conditions, and replacement scenarios. Fig. 3.7.1 shows multi-take video montage examples, both shot outdoors with a handheld camera. Fig. 3.7.2 shows dubbing results of a translation scenario, where the source and target depict the same subject speaking in different languages, with source captured in a studio setting and target captured outdoors. Figs. 3.7.3 shows a replacement result with different source and target subjects and notably different performances. Fig. 3.7.4 shows a retargeting result with different subjects, where the target was used as an audiovisual guide and the source retimed to match the target.

3.7.2 User interaction

Although the majority of our system is automatic, some user interaction is required. This includes placing markers in FaceGen, adjusting markers for tracking initialization, and specifying the initial blending mask. Interaction in FaceGen required 2-3 minutes per subject. Tracking initialization was performed in less than a minute for all videos used in our results; the amount of interaction here depends on the accuracy of the automatic face detection and the degree to which the subject's expression and viseme differ from closed-mouth neutral. Finally, specifying the mask for blending in the first frame of every example took between 30 seconds and 1 minute. For any given result, total interaction time is therefore on the order of a few minutes, which is significantly less than what would be required using existing video compositing methods.

3.7.3 Comparisons

Vlasic et al. [177] use a face tracking and replacement pipeline that is similar to ours. We reprocessed their original scan data [177] to place it into correspondence with a face mesh that covers the full face, including the jaw. This was done for two reasons. First, the original model only covered the interior of the face; this restricted us to scenarios where the timing of the source and target's mouth motion must match exactly. While this is the case for multi-take montage and some dubbing scenarios when the speech is the same in both source and target videos, it presents a problem for other situations when the motion of the target jaw and source mouth do not match. For these situations – changing the language during dubbing or in arbitrary face replacements – a full face model is necessary so that the source's jaw can also be transferred (Fig. 3.7.5 a).

Second, our experience using the original interior-only face model confirmed earlier psychological studies that had concluded that face shape is one of the stronger cues for identity. When source and target subjects differ, replacing the interior of the face was not always sufficient to convey the identity of the source subject, particularly when source and target face shapes differ

(a) Source and target frame pairs

(b) Blended result

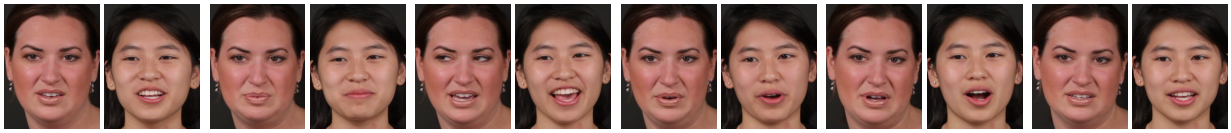
Figure 3.7.3: *Face replacement. (top) Cropped source and target frames (left and right, respectively) showing casual conversation and head motion, with the target shot handheld. (bottom) Frames from the blended result, combining frames from two subjects with notably different expression, speech, pose, and face shape.*

significantly.

In Vlasic et al., face texture can come from either the source or the target, and morphable model parameters can be a mixture of source and target. When the target texture is used, as in their puppetry application, blending the warped texture is relatively easy. However, the expressiveness of the result stems exclusively from the morphable model, which is limited and lacks the detail and nuances of real facial performances in video. On the other hand, taking face texture from the source makes the task of blending far more difficult; as can be seen in Fig. 3.6.1, the naïve blending of source face texture into the target used in Vlasic et al. produces bleeding and flickering artifacts that are mitigated with our seam finding and blending method.

3.7.4 User study

To quantitatively and objectively evaluate our system, we ran a user study using Amazon's Mechanical Turk. Our test set consisted of 24 videos: 10 unmodified videos, 10 videos with replaced



(a) Source and target frame pairs

(b) Blended result

Figure 3.7.4: *Face retargeting. (top) Cropped source (retimed) and target frames (left and right, resp.) from indoor recordings of poetry recitation. (bottom) Frames from the blended result combine the identity of the source with the background and timing of the target.*

faces, and four additional videos designed to verify that the subjects were watching the videos and not simply clicking on random responses. All videos were presented at 640 $\times$ 360 pixels for five seconds and then disappeared from the page to prevent the subject from analyzing the final frame.

The subjects were informed that the video they viewed was either "captured directly by a video camera" or "manipulated by a computer program." They were asked to respond to the statement "This video was captured directly by a video camera" by choosing a response from a five-point Likert scale: strongly agree (5), agree (4), neither agree nor disagree (3), disagree (2), or strongly disagree (1). We collected 40 distinct opinions per video and paid the subjects $0.04 per opinion per video. The additional four videos began with similar footage as the rest but then instructed the subjects to click a specific response, e.g., 'agree', to verify that they were paying attention. Subjects who did not respond as instructed to these videos were discarded from the study. Approximately 20 opinions per video remained after removing these users.

The average response for the face-replaced videos was 4.1, indicating that the subjects believed the videos were captured directly by a camera and were not manipulated by a computer program. The average response for the authentic videos was 4.3, indicating a slightly stronger belief that

(a) (b) (c)

Figure 3.7.5: *Failure cases. (a) Split frame of two nearby frames in a blended result where the model does not cover the full face*[1]*. (b) When the tracking fails, the source content for replacement is distorted, seen here after alignment. (c) Significant differences in lighting between source and target lead to an unrealistic blended result, where the lighting on the right is darker on the source face but not in the target environment.*

the videos were captured by a camera. None of the face-replaced videos had a median score below 4 and three of the videos had a median score of 5. These results indicate that our method can produce convincing videos that look similar to those coming directly from a camera.

3.7.5 Limitations

Our approach is not without limitations (Fig. 3.7.5). Tracking is based on optical flow, which requires that the lighting change slowly over the face. High frequency lighting, such as hard shadows, must be avoided to ensure good tracking. Additionally, the method assumes an orthographic camera; while estimation of parameters of a more sophisticated camera model is possible, we use the simple model and shot our input videos with longer focal lengths that better approximate an orthographic projection. Finally, tracking often degrades beyond the range of poses outside $\pm 45^o$ from frontal. Even with successful tracking, the geometric fit can cause artifacts in the final result. For example, the fit is sometimes insufficient for the large pose differences between source and target. This is particularly noticeable in the nose area when, for example, the head is significantly

tilted downwards, causing the nose to distort slightly.

Pose is also constrained to be sufficiently similar between source and target to prevent occluded regions in the source face from appearing in the pose-transformed target frame. For cases where we have control over source acquisition, the source subject can be captured in a frontal pose as we do here, or in a pose similar to the target, both ensuring no occluded regions. However when existing footage is used as the source, it is necessary to ensure compatible pose between source and target. This issue could be alleviated by automatic or user-assisted inpainting that derives the missing texture from spatially and temporally adjacent pixels in the video sequence.

In all examples shown here, source / target pairs are of the same gender and approximate age and thus of roughly similar proportions. Any difference in face shape can be accounted for by a single global scale to ensure the source face covers the target. For vastly different face shape, e.g., a child and adult, this may not be sufficient. However it is plausible to add a 2-d warping step, similar to that used in [92], that warps the target face and nearby background to match the source before blending.

Lighting must also be similar between source and target. For multi-take montage scenarios, where source and target are typically captured in close succession in the same setting, this condition is trivially met. Likewise, when either the source or target is captured in a studio setting, with full control over the lighting setup, this condition can also be met with the same efforts required for plausible green screening. However such matching can be difficult for novices or may be impossible if the source and target are from existing footage.

Finally, seam finding and blending can fail for difficult inputs. For example, when hair falls along the forehead, there may be no seam that generates a natural blend between source and target. Strong differences in illuminations will lead to bleeding artifacts because it sometimes is not possible for the seam to avoid such regions. Fig. 3.7.5 shows some examples where these limitations are manifested in the final result.

[1] Target frame from www.whitehouse.gov.

3.8 Summary

The shape of a person's face varies substantially with changes in speech, expression, and pose leading to variations in appearance. In this chapter, we have shown that it is possible to capture the subtleties of face appearance by using a multi-linear model to describe variations in face geometry. Based on this representation, we have presented a video face replacement system that requires only single-camera video and minimal user input and is robust under significant differences between source and target. We have shown with a user study that results generated with this method are perceived as realistic. Our method is useful in a variety of situations, including multi-take montage, dubbing, retargeting, and face replacement.

There are a number of extensions of this work that will allow it to be applied to more general scenarios. Videos with large pose variations will require more accurate tracking algorithms as well as inpainting to handle occlusions. Videos with vastly different face shapes would have to be compensated using 2-d background warping. The ability to estimate and correct the illumination in videos, would make this approach applicable to sequences captured under different lighting conditions. Another interesting avenue for future work would be to extend the techniques we discussed in Chapter 2 to video and combine it with the work presented in this chapter. This would allow us to use footage that differs widely in terms of contrast, noise, texture, and blur.

4

Enhancing Image Quality using Video Clips

In the previous chapters, we used models of appearance to analyze and edit images. However, we did not explicitly account for the camera photographing the scene. The properties of the camera have a profound effect on the appearance of the final image. For example, the camera sensor determines the resolution and noise characteristics of the image, the optics and camera (and scene) motion lead to image blur, and exposure and white balance settings on the camera affect the luminance and colors of the image. Analyzing images and inferring these camera properties requires an understanding of the imaging process. This chapter utilizes one such imaging model to enhance the images captured by a low-quality camera.

4.1 Introduction

Often the most important photographic moments are unexpected and difficult to predict—the proud grandfather wanting to capture his grandson's first home run or a delighted mother trying to catch that perfect smile from her daughter. In many such scenarios, the photographer has to stay ready, finger on the trigger, trying to time the shutter release perfectly. Unfortunately, these important moments are often missed, leaving a photographer frustrated with a photograph taken just a bit too early or a touch too late. In other cases, there is no one right instant; the moment can only be captured in a still image by combining multiple instances in time.

In these situations, a good alternative is to take a video to capture the whole action. This is an increasingly available option as practically all cameras and phones today have a video mode. The video provides a temporally dense sampling of the action that ensures not only that the right moment is never missed, but that it can be revisited later on.

Unfortunately, using a video camera in lieu of a still camera comes at a cost. Even high-end video cameras today have a much lower resolution and higher noise levels than still cameras. And since the best camera is the one that you have with you, it is increasingly likely that these short videos are shot on cellphones, smartphones, or iPods with low-quality cameras. Moreover, video clips on these portable devices are compressed aggressively. As a result, a single video frame has a much lower quality than a corresponding photograph shot with a still camera, making it less satisfying to use directly.

In this work, we consider the problem of creating a single high-quality still image—a *snapshot*—from a video clip. The snapshots we produce have higher resolution, lower noise, and less blur than the original video frames. By modeling the camera along with scene motion and saliency, we can produce either a snapshot of a single moment in time where scene motion is suppressed (Fig. 4.1.1(c)), or a snapshot that summarizes the motion of salient objects and actions (Fig. 4.1.1(d)).

(a) Example video frames | (b) Bicubic upsampling | (c) Video snapshot with motion suppression | (d) Video snapshot with motion summarization

Figure 4.1.1: *Comparisons of image enhancement. (a) Four frames from a short clip showing a man jumping from a cliff. Each of these frames has low resolution, high noise and compression, and captures the man at only one time instant. (b) Bicubic upsampling one particular frame of interest. Note that the high frequency texture on the rocks on the left and the trees on the right are lost, and there are blocking artifacts in the water. Our framework leverages the multiple frames in the video to produce a super-resolved, denoised snapshot. We can do this while suppressing the motion of the jumping man (c) to freeze the motion in time, or while summarizing the motion (d) to capture the activity in a single image. Note that in both these results the rocks and trees are sharper, and the blocking artifacts in the water have been removed.*

We assume the input to our system is a short video clip and a user-specified reference frame. We request a user-specified reference frame because picking the most important moment in a video is a subjective activity that depends on the goals, intentions, and preferences of the user. Our algorithm first aligns neighboring frames in the video to the reference frame, and then combines these frames using a Bayesian multi-image enhancement formulation to perform super-resolution, denoising, sharpening, and/or motion summarization.

Previous work either uses *all* of the aligned frames equally to generate a restored image, or selects a single frame for each pixel to create a composition (such as digital photomontage [4]). In contrast, our algorithm combines each image and pixel contribution differently using a set of

importance-based weights. Our primary contribution is a novel importance-based framework that bridges the gap between traditional multi-image super-resolution and multi-image compositing. It can create images where stationary, non-salient parts of a scene are enhanced by combining data from multiple frames, while the salient, moving objects are enhanced using support from a single frame. Furthermore, by computing per-pixel, per-frame weights, we incorporate aspects of *lucky imaging*, where poor-quality frames in the video are not weighted as heavily when computing the resulting snapshot [95].

4.2 Related Work

Image enhancement techniques such as super-resolution and denoising have a long history in image processing and computer vision. Also, recent work on image fusion has looked at the problem of using user-defined preferences to fuse a collection of images into a single photomontage. Our work is related to both these problems, and in this section we briefly review these areas.

4.2.1 Image enhancement

Since the early work of Tsai and Huang [174], image super-resolution has been studied extensively and Park et al. [130] present a comprehensive survey of a number of recent methods. Super-resolution is an inherently ill-posed problem, and early work focused on using multiple low-resolution frames with aliasing to create a high-resolution image. The image formation process is modeled as a warping and subsampling of the high-resolution image, and these techniques explicitly invert this process to solve for a higher-resolution image that is consistent with the warped and blurred low-resolution observations [86]. Often, the parameters of the warping and subsampling are assumed to be known; this requirement can be removed by marginalizing over these parameters in a Bayesian framework [134, 168]. However, these techniques depend on the aliasing in the low-resolution frames, and because cameras often band-limit the high frequen-

cies to minimize aliasing, there is a theoretical limit on the amount of resolution enhancement (approximately an upsampling factor of 2) that these methods can provide [10, 113].

More recent work has generalized super-resolution to scenes with arbitrary motion by using non-local means methods [163] or by using high-quality optical flow methods to estimate per-pixel motion [115]. Parallel to the work on multi-image super-resolution, researchers have also looked at the problem of super-resolving a single image. This problem is less constrained than multi-image super-resolution, and is often dealt with by using dictionaries of images patches [69, 188], or sparse priors [166]. Another way to constrain this problem is to use the fact that image patches often recur (possibly at different scales and orientations), and recent work has used this to spatially super-resolve images [73], and spatio-temporally upsample videos [150].

Our work leverages the information in all the frames of the video clip to create a super-resolved video snapshot. Similar to classic multi-frame super-resolution [86], we estimate the snapshot by modeling the warping and subsampling, and explicitly inverting them. However, unlike most work on super-resolution where all the pixels in the video clip are treated in the same way, we introduce the notion of importance-based weights that encode the influence each pixel has on the final snapshot. This allows us to perform a number of other operations in the multi-image super-resolution framework.

Like super-resolution, image denoising is a well studied problem in image processing, and we refer the reader to Chatterjee and Milanfar [38] for a survey of recent work. Early work in image denoising made use of the sparsity of coefficients when transformed into the wavelet domain [138, 155]; here large wavelet coefficients were assumed to correspond to image structure and were retained, while small coefficients were removed. Edge-preserving filters [133, 169] have also been used to smooth noise out while retaining image structure. Priors based on natural image statistics have been incorporated in image denoising [146]. More recently, researchers have looked at making use of image sparsity in the spatial domain for image denoising. This has led to a class of algorithms where an image is modeled as consisting of a small set of patches. The K-

SVD algorithm [5] learns an over-complete dictionary for image patches that can then be used for denoising [54]. In non-local means methods [31], patches across the image are aggregated, using weights based on their similarity, to smooth noise out. While all these techniques were proposed for single images, they have been used subsequently for video clips. Many video denoising techniques use motion estimation to align spatial neighborhoods. Once aligned, these frames can be merged using weights based on a spatio-temporal bilateral filter [20] or denoised using a temporal extension of non-local means techniques [114].

Like other video denoising techniques, we combine multiple frames to denoise video clips and create a video snapshot. However, we use a combination of weights based on sharpness, saliency, motion accuracy, etc. that allows us to incorporate a number of other effects into the denoised snapshot.

4.2.2 Image fusion

Agarwala et al. [4] propose a system that combines multiple images to create a single *photomontage*. In their system, users define objectives – locally by using strokes, or globally by specifying attributes to be used – that are used to decide which image each pixel in the photomontage is copied from. Similarly, "Salient Stills" [167] create a single image by fusing multiple images using different global criteria. While our goal is similar to this class of techniques, our work differs from them in its ability to automatically combine image-enhancement as well as photomontage-style image fusion in the same unifying framework.

4.3 Importance-based Image Enhancement

Given multiple video frames and one user-selected reference frame, our goal is to generate a clean, enhanced version of the reference frame. We adopt an image formation model that maps the restored image to the original frames that are deemed "degraded". This image formation model is

popular in multi-image restoration techniques such as super-resolution (e.g., [86]). The restoration process uses multiple degraded observations to invert this image formation model and estimate the high-quality input. Our framework introduces importance-based weights into this inversion process. While our framework can be easily applied to any linear image formation model, we will discuss it here in the context of multi-image super-resolution.

Given a set of N video frames $L_k, k = 1, 2, \cdots N$ of resolution $h \times w$, multi-image super-resolution seeks to combine the frames to obtain a single high-resolution $sh \times sw$ image H. The standard super-resolution problem [55] assumes a generative image formation model given by:

$$L_k = D_s(P(T_k H)) + \eta, \tag{4.1}$$

where T encodes the camera motion, P denotes the camera's anti-aliasing filter, D_s is a decimation by factor s, and η is the observation noise.

D_s, P, and T are all linear operators and can be combined into a single operation $M_k(\cdot) = D_s(P(T_k(\cdot)))$. Under the assumption of zero-mean Gaussian noise, i.e., $\eta \sim N(0, \sigma_\eta^2)$, this reduces to solving for H by minimizing the following energy function:

$$E_d = \sum_{k=1}^{N} ||(L_k - M_k H)||^2 / \sigma_\eta^2. \tag{4.2}$$

While multi-image super-resolution is better conditioned than single-image super-resolution, errors in alignment, saturation, noise, etc. can make solving Eqn. 4.2 ill-posed. This is often handled by regularizing the solutions with a prior. By using a sparse prior on the distribution of image gradients that is based on natural image statistics [108], the total energy to minimize has the form:

$$E_t = \sum_{k=1}^{N} ||L_k - M_k H||^2 / \sigma_\eta^2 + \lambda(\nabla H)^{0.8}. \tag{4.3}$$

Eqn. 4.3 represents the standard multi-image super-resolution problem. The high-resolution

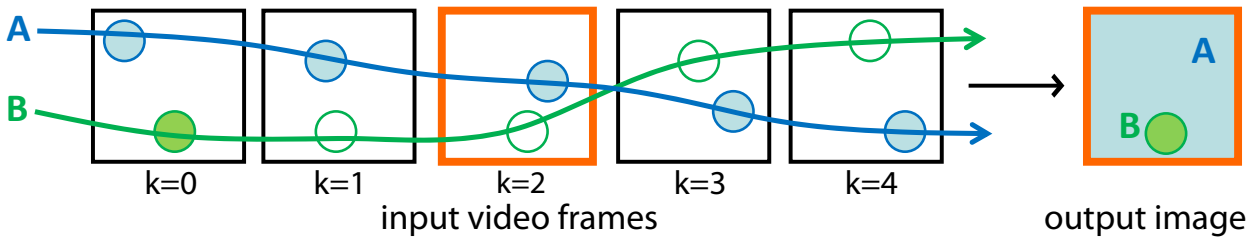


Figure 4.3.1: *Weighted multi-image enhancement. Manipulating the weights in Eqn. 4.4 allows us to handle multi-image enhancement operations while preserving salient objects. The weights for blue patches A in all the frames are equal (i.e., $W^0 = W^1 = W^2 = W^3 = W^4$), and the output patch A is a linear combination of all the input patches A^k as in Eqn. 4.2. The weights for the green patches B are non-zero only in frame 0 (i.e., $W^0 = 1, W^1 = W^2 = W^3 = W^4 = 0$), and the output patch B is copied as is from it.*

image H can be solved for using iterative re-weighted least squares (IRLS) [124].

In this formulation, every output pixel $H(x_h, y_h)$ is a linear combination of *all* the aligned input pixels $L_k(x_l, y_l), k = \{1, 2, \cdots N\}$. In many scenarios this is ideal; for example, the noise in the low-resolution frames is most suppressed when all frames are combined. However, in some cases, some frames (or some regions of frames) are inherently more important than others (e.g., a smiling face or a moving object), and it is usually desirable to preserve them in the final result.

This idea is the basis of image fusion algorithms such as digital photomontage [4], where every output pixel $H(x_h, y_h)$ is set to exactly one of the corresponding input pixels $L_k(x_l, y_l)$. The choice of which pixel is picked is decided by user-specified objectives. In contrast to multi-image enhancement, this approach preserves important regions, but at the cost of retaining the resolution and noise of the input frames.

Our goal is to combine aspects of these two approaches – multi-image super-resolution and image fusion – into a single framework that combines multiple low-importance pixels while preserving important pixels as they are. To bridge this gap we introduce the notion of *importance-based weights* into the restoration equation:

$$E_t = \sum_{k=1}^{N} ||W_k \{L_k - M_k H\}||^2 / \sigma_\eta^2 + \lambda (\nabla H)^{0.8}. \tag{4.4}$$

$W_k(x, y)$ encodes the importance of each (low-resolution) input pixel $L_k(x_l, y_l)$, and decides how they are combined to produce the (high-resolution) output pixels $H(x_h, y_h)$ that they are aligned with. The incorporation of these weights allows us to generalize Eqn. 4.3 in many different ways. For instance, by using equal weights, i.e., $W_k(x_l, y_l) = 1 \forall k$, Eqn. 4.4 reduces to the original multi-image super-resolution problem of Eqn. 4.3. On the other hand, using sparse weights, i.e., $W_k(x_l, y_l) \in \{1, 0\}, \sum_k W_k(x_l, y_l) = 1$, Eqn. 4.4 reduces to the digital photomontage framework. More importantly, since the weights are defined per-pixel, we can combine both of these scenarios in the same image, as illustrated in Fig. 4.3.1. By setting the weights appropriately, some parts of the output image can be enhanced by combining multiple frames, while the others can be preserved from an individual frame.

While the importance-based enhancement of videos has been discussed in terms of super-resolution in Eqn. 4.4, it can be easily generalized beyond this operation. Many imaging operations, including filtering, denoising, deblurring, stitching, and compositing can be expressed as a linear processing of the input video pixels, and for the appropriate choice of operator M_k, have the same form as Eqn. 4.4.

4.4 Creating Video Snapshots

Based on these ideas we now discuss how to create snapshots from a video clip. We assume that the camera motion in the video is well-approximated by an affine transform. Given an input video clip and the user-specified reference frame, we detect interest points [118] in the video frames, and estimate an affine motion model using RANSAC [66]. We assign the weights for each frame based on three different spatial features – motion confidence, local sharpness, and temporal saliency – and time. Finally, we combine the different importance weights, and use them to solve Eqn. 4.4 for the output snapshot.

4.4.1 Motion confidence

Motion estimation is a challenging problem, and even state-of-the-art algorithms make errors while handling general scenes with arbitrary camera motion. To ensure that these errors do not lead to artifacts in the snapshots, we use weights based on the re-projection error of the estimated motion. To make this motion confidence measure robust to noise and compression artifacts, we first blur the frames using a low-pass Gaussian filter with $\sigma = 1.0$ to create the smoothed frames L'_{ref} and L'_k. We then warp the filtered reference frame L'_{ref} to the k^{th} frame using the estimated motion T_k^{-1} and assign the motion confidence as:

$$W_k^m = N(T_k^{-1}(L'_{ref}) - L'_k; 0, \sigma_m^2), \tag{4.5}$$

where $\sigma_m = 0.01$. Filtering the images ensures that the differences between pixels of the blurred images correspond to the spatially-weighted differences between neighborhoods of pixels in the original images.

4.4.2 Local sharpness

Motion blur (due to camera or scene motion) and defocus blur (due to an out-of-focus camera) often degrade the quality of a video. While creating a snapshot, we avoid pixels that are blurred by using the local sharpness measured at every pixel as weights. Our local sharpness measure estimates the high-frequency content in the neighborhood of a pixel, and is computed as a difference of Gaussians of each input frame:

$$W_k^{ls} = |L_k - G_\sigma \otimes L_k|, \tag{4.6}$$

where G_σ is Gaussian filter with standard deviation 3.

4.4.3 Temporal saliency

To preserve object motion in the video, we use a temporal saliency measure that detects and preserves salient regions in the scene. Many measures have been proposed for both spatial [88] and spatio-temporal saliency [87]. We use a simpler variation of the "flicker conspicuity" measure used by Itti and Baldi [87]. Our method estimates temporal saliency as the deviation of the video frames from an estimated background model. We first align all the video frames and median filter them to remove moving objects and create a background model for the video. We assign saliency weights to the input pixels based on how much they deviate from this background model. To ensure that this measure detects moving objects while staying robust to noise, compression artifacts, and small frame misalignments, we first blur the median image and the video frames using a low-pass Gaussian filter (with standard deviation set to 2.0) to create the smoothed frames L'_k and L'_{median}. The saliency weights are then set as:

$$W_k^{sal} = 1 - N(T_k^{-1}(L'_{median}) - L'_k; 0, \sigma_{sal}^2), \tag{4.7}$$

where $\sigma_{sal} = 0.03$. Note that because we use deviations from the median image to detect salient objects, all stationary (and even very slow-moving objects) will not register as being salient, and will be retained as part of the background in the final snapshot.

While saliency can be used to capture moving people and objects, and summarize actions in snapshots, sometimes a user might want to create snapshots where the moving parts have been *removed*, i.e., a "clean-plate" image. For example, while filming a building, the pedestrians photographed walking back and forth in front of it are often undesirable elements that the photographer might want to remove. To be able to do this in our framework, we use the notion of anti-saliency which is defined as:

$$W_k^{isal} = 1 - W^{sal}. \tag{4.8}$$

This formulation gives higher weights to stationary parts of the scene while removing transient objects.

4.4.4 Time

Artists and scientists often use tools such as shear, blur, and action lines [46] to create the perception of movement in static images. We manipulate the saliency weights estimated from Eqn. 4.7 using time to create perceptual cues about the motion of the salient objects in the snapshot. In particular, we use three different weighting schemes:

1. *Sampling*. Saliency weights are retained at periodic frames and set to 0 at all other frames, i.e., $W_k^{samp} = W_k^{sal}\delta(k - ik_0)$. In video clips where the object motion is very small, this makes sure that the snapshot is not cluttered.

2. *Linear Ramp*. Saliency weights are scaled linearly from the first frame to the last, i.e., $W_k^{ramp} = kW_k^{sal}$. Gradually accentuating the salient object over time creates cues for the direction of motion.

3. *Overlaying*. When regions identified as salient in different frames overlap spatially, only the latter of the regions is retained and all the others are removed, i.e.,

$$W_k^{over}(x,y) = 0, \ if \ \ W_l^{sal}(x,y) \ > \ \beta \qquad (4.9)$$
$$\forall l = \{k+1, \cdots, N\}$$

   This creates the impression of motion in the direction of time. Alternatively, we can reverse this to create the impression of motion against time by setting the weights as:

$$W_k^{rev-over}(x,y) = 0, \ if \ \ W_l^{sal}(x,y) \ > \ \beta \qquad (4.10)$$
$$\forall\, l = \{1, \cdots, k-1\}$$

(a) Example video frames

(b) $r_{saliency} = 0$ (c) $r_{saliency} = 1$ (d) $r_{saliency} = 6$

Figure 4.4.1: *Sparsifying the feature weights. Exponentiating the feature weights makes them sparse, resulting in some pixels in the output snapshot being reconstructed from very few frames. This is illustrated on this video (35 frames, 960×540 resolution) of a bicyclist (a). The saliency measure picks out the moving bicyclist. (b) When the exponent for the saliency measure is 0, the weights are uniform, all the frames are combined, and the bicyclist is blurred out. (d) As the exponent is increased to 6, the saliency weights become sparse, and the bicyclist is reconstructed from single frames. The non-salient regions of the image are not affected by this, and continue to be estimated from all the frames. Credit: Vimeo user* markusarulius.

4.4.5 Combining feature weights

To combine the weights computed on each feature, we normalize them to the $[0, 1]$ range, scale and exponentiate them, and finally sum them:

$$W'_k = \sum_f \alpha_f (W_k^f)^{r_f} + \epsilon, \tag{4.11}$$

where ϵ is a small number (set to 0.001) that ensures that every input pixel is given a non-zero weight. By varying the exponent r_f in Eqn. 4.11, we can smoothly transition between uniform ($r_f = 0$) and sparse weights ($r_f \to \infty$). This allows us to unify multi-image enhancement and photomontage in a single framework. The effect of manipulating this exponent is illustrated in Fig. 4.4.1. The salient regions of each frame all have high weights, while all other regions have uniformly low weights. When the saliency weights are raised to exponent zero, all the frames are combined to denoise the video; however, this blurs the salient regions out. As the exponent is

increased, the difference in the weights of the salient and non-salient regions is accentuated until they are copied directly from the input video into the output snapshot. Meanwhile, regions of the video that are never salient and have uniformly low weights (ϵ in Eqn. 4.11) continue to be reconstructed by combining multiple frames. In practice, we found that $r_f = 6$ worked well for our examples.

4.4.6 Normalizing weights

To ensure that the error at each output snapshot pixel is weighted equally in the total energy, we normalize the weights. This is done by first warping the weights by the motion estimated on the video frames, normalizing them, and then unwarping them:

$$W_k = (T_k)^{-1} \left\{ T_k(W'_k) / \sum_{k=1}^{N} T_k(W'_k) \right\}. \tag{4.12}$$

4.4.7 Image Prior

In traditional image enhancement, every pixel in the output image is a linear combination of approximately the same number of input image pixels. As a result, in most cases, the prior used in Eqn. 4.3 is spatially constant. However, in our case, the application of the spatially-varying weights changes the support of each output pixel. To take this into account, we use a spatially-varying image prior. We identify the number of input pixels that are aligned with, and contribute to the reconstruction of each output snapshot pixel; in practice, we test for this by thresholding the weights W_k by $0.1/K$, i.e., 10% of the value that a uniform weight would take. We scale the prior term by the inverse of the number of input pixels that contribute to each snapshot pixel. Incorporating this spatially varying prior into our framework leads to a graceful transition between very little regularization at pixels with large data support, and more regularization at pixels with small or no data support.

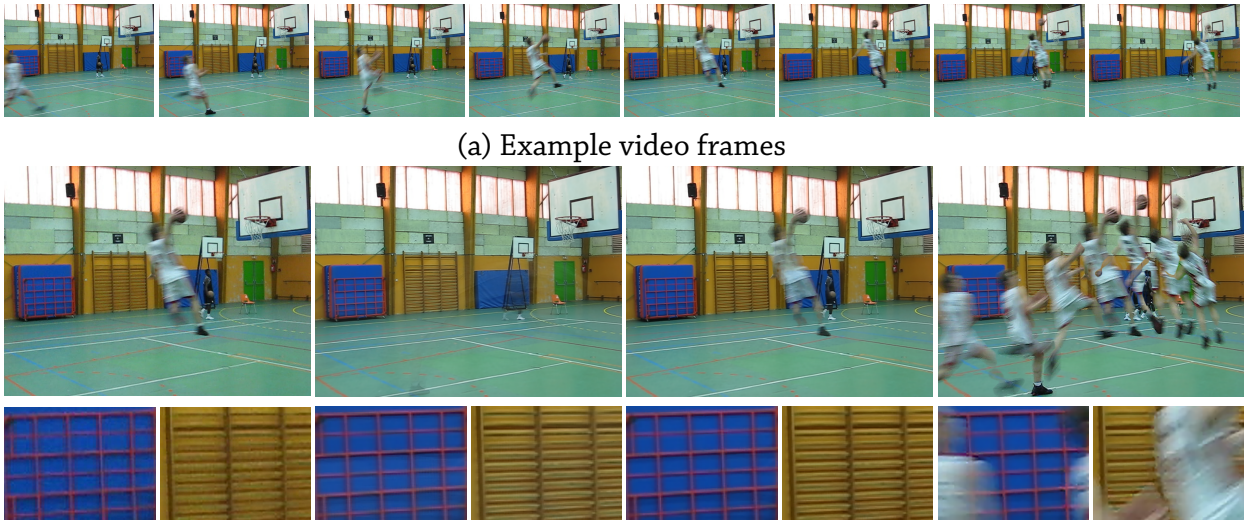
Enhancements / Videos	Super-resolution	Noise reduction	Sharpening	Motion suppression	Salient object summary	Temporal effects
jump (Fig. 4.1.1)	X	X	X	X	X	
ditchjump (Fig. 4.4.1)				X	X	X
dunks (Fig. 4.5.1)	X	X	X	X	X	
mural (Fig. 4.5.2)	X	X	X			
focus (Fig. 4.5.3)	X	X	X			
basketball (Fig. 4.5.4)	X	X	X	X	X	
bounce (Fig. 4.5.6)				X	X	X
dive (Fig. 4.5.7)	X	X	X	X	X	
walk (Fig. 4.5.8)	X	X	X	X	X	
calendar (Fig. 4.5.10)	X		X			
foliage (Fig. 4.5.10)	X		X	X		

Table 4.4.1: *A summary of the enhancements we apply to our input videos.*

4.5 Results

We now present the results of enhancing a number of short video clips using our framework. All these videos clips were either captured with low-quality video cameras or downloaded from the video sharing website Vimeo (http://www.vimeo.com). They range in length from 11 frames to 31 frames and have a combination of low-resolution, high camera noise, and compression artifacts. The enhancements and effects we apply to each of them are summarized in Table 4.4.1.

We assume that the motion in the video clip is well modeled by an affine camera model. For each video clip, we estimate the inter-frame motion by fitting an affine model to interest points. The motion and the video frames are then used estimate the importance weights. With the exception of Figs. 4.4.1, 4.5.6, and 4.5.10, all results are produced using a super-resolution factor of 2. The anti-aliasing point spread function (P in Eqn. 4.1) is set to a Gaussian filter with $\sigma = 1.2$ and the noise level (σ_η in Eqn. 4.4) is automatically estimated from the reference frame using the method of Liu et al. [116]. Finally, we put the weights and the estimated motion together to set up the energy function of Eqn. 4.4. We solve for the output video snapshot by minimizing this energy function using conjugate gradients. We perform 5 iterations of IRLS for every result and each IRLS iteration uses 10 iterations of conjugate gradients. The time taken to compute a



(a) Example video frames

(b) Single-image super-resolution (X 2) [188]

(c) Multi-image super-resolution (X 2) [86]

(c) Video snapshot with motion confidence (X 2)

(d) Video snapshot with saliency (X 2)

Figure 4.5.1: *Video snapshots with saliency weights. (a) This clip of a basketball player dunking (25 frames, 640 x 480 resolution) suffers from low resolution and high noise. (b) Upsampling the reference frame using the single-image super-resolution [188] produces a noisy result. (c) By combining multiple frames, multi-frame super-resolution [86] produces a result with more detail and low noise, but blurs out the player completely. (d) Using the motion confidence as weights preserves the high-resolution, low-noise background and captures the player. (e) Using saliency weights and temporal-overlaying summarizes the player's movement while retaining the high-quality background. Credit: Vimeo user* A.S. Saint Pantaléon Basket.

snapshot is almost completely dominated by the time spent in minimizing Eqn. 4.4; this depends approximately linearly on the resolution of the output snapshot and the number of input frames being used. Our unoptimized C++ solver takes anywhere from 6 minutes on our smallest example (Fig. 4.5.1) to 15 minutes on our largest example (Fig. 4.5.4) on an i7 2.67 GHz PC.

The quality of results from super-resolution closely depends on the accuracy of the motion estimation. This is especially true of videos with complex camera motion and moving objects in the scene. By using weights based on motion confidence we ensure that only pixels where the motion estimates are reliable are used. Because they are computed with respect to the reference frame, motion confidence weights also help in suppressing moving objects in the video, while moving objects in the reference frame are preserved in their position. The results of using motion

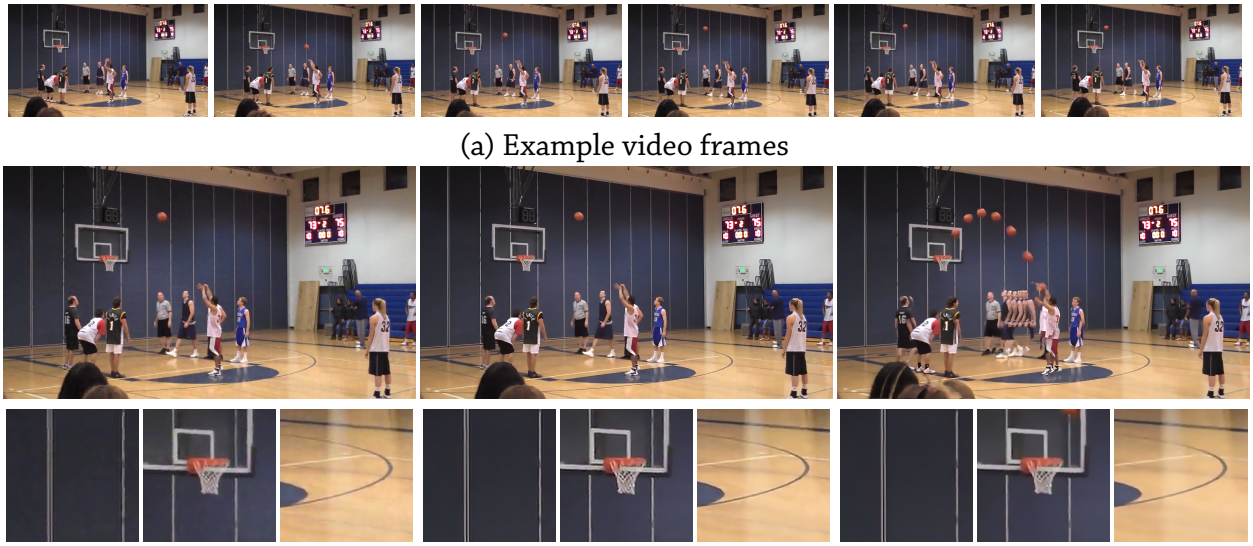
(a) Example video frames (b) Multi-image super-resolution (X 2) [86] (c) Video snapshot with sharpness weights (X 2)

Figure 4.5.2: *Video snapshots for motion blur. Often when photographing a scene with a moving camera, some of the frames, possibly even the desired frames captured, are motion blurred. (a) This is illustrated on this video clip of a mural captured with a hand-held video camera (21 frames, 640 x 360 resolution), where the reference frame has the best composition of the scene, but is motion blurred. (b) Most of the frames in this video clip are blurred and combining all of them to super-resolve the reference frame [86] results in a blurry image. (c) Using the local sharpness weights in our framework ensures that pixels from only the sharp frames are propagated to the reference frame, resulting in a sharp snapshot.*

(a) Example video frames (b) Multi-image super-resolution (X 2) [86] (c) Video snapshot with sharpness weights (X 2)

Figure 4.5.3: *Video snapshots for defocus blur. (a) In this video clip (21 frames, 640 x 360 resolution), shot with a handheld video camera, the focal plane is being moved from the back to the front to create an unstabilized focal stack. (b) Naive multi-image super-resolution [86] combines both sharp and blurry frames, and produces a result that is only marginally sharper because it does not model the defocus blur in the video properly. (c) Our result uses local sharpness weights to identify and combine the sharpest pixels in the input video clip to produce an all-in-focus super-resolved snapshot.*

confidence in our framework are illustrated in Figs. 4.1.1, 4.5.1, 4.5.4, 4.5.5, 4.5.7, and 4.5.8.



(a) Example video frames

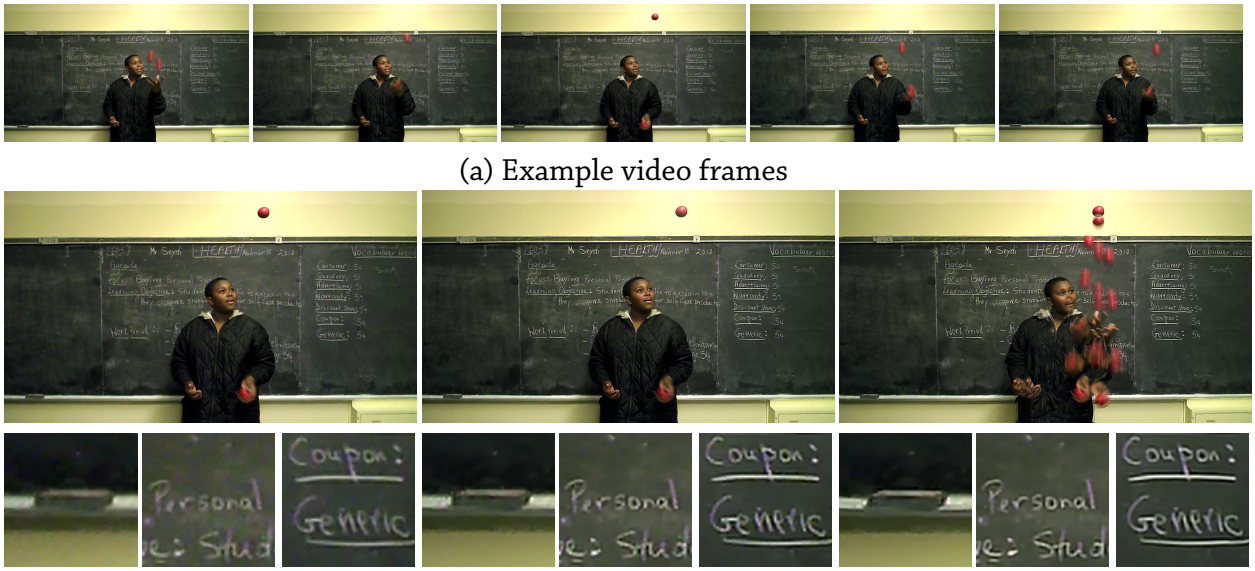
(b) Single-image super-resolution (X 2) [188]

(c) Video snapshot (X 2) with motion confidence

(d) Video snapshot (X 2) with saliency and sampling

Figure 4.5.4: *Video snapshots with motion. (a) This input video clip (31 frames, 640 x 360 resolution) has noise and compression artifacts. (b) Single-image super-resolution [188] super-resolves the reference frame but is unable to remove the noise and blocking artifacts. (c) Using motion confidence weights produces a low-noise 1280 x 720 snapshot with the moving players and the basketball preserved in place. (d) By using saliency weights with time-sampling we can retain the high-quality background from (c) while capturing the motion of the players and the basketball. Credit: Vimeo user* Charles Skoda.

Blur caused by camera shake or the wrong focal settings is one of the most common problems with photographs. While the short exposure time of video clips alleviates the effect of camera motion to an extent, it is not unusual to capture a video sequence and to later find out that intermittent frames are blurred. Estimating the blur kernel (which is spatially-varying in most cases) and deconvolving the image is a very difficult vision problem. Instead, we use local sharpness weights to automatically identify and reconstruct the output snapshot from only the sharpest pixels in the video clip. This approach also has the advantage that it handles variation in scene texture gracefully; smooth, low-texture regions will have uniformly low sharpness values and can be estimated from many frames, while textured regions and strong edges are reconstructed from only the sharpest pixels. Local sharpness weights can be used to create the sharpest possible snapshot in the case of motion blur (Fig. 4.5.2), as well as an all-in-focus image from a clip with varying



(a) Example video frames

(b) Single-image super-resolution (X 2) [188]

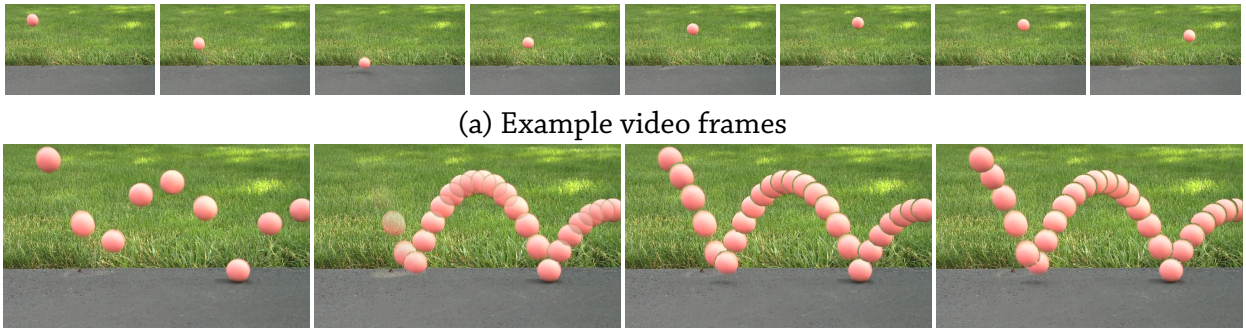
(c) Video snapshot (X 2) with motion confidence

(d) Video snapshot (X 2) with saliency and time-sampling

Figure 4.5.5: *Video snapshots with saliency weights. (a) This juggling video (24 frames, 640 x 480 resolution) has low resolution and high noise. (b) Single-image super-resolution [188] improves the resolution marginally and cannot handle the noise. Our method improves the resolution significantly (note the letters on the blackboard), while also denoising the image (note the duster in the bottom right). We do this while either (c) capturing the moment in the reference frame, or (d) depicting the motion of the ball and the hands. Credit: Vimeo user* BCCP Video.

defocus blur (Fig. 4.5.3).

Motion is often a critical component of video sequences, and the depiction of motion in static images has a long history in artistic and scientific visualization. However, most work on image enhancement avoids the issue of moving objects in a video. By using saliency weights in our framework, we are able to combine multiple frames and create a high-resolution, low-noise, sharp background while retaining the salient moving objects from individual frames. This results in high-quality still images that summarize the entire video clip in a single static snapshot (Figs. 4.1.1, 4.4.1, 4.5.1, 4.5.4, 4.5.5, 4.5.6, 4.5.7, 4.5.8, and 4.5.9). We can also use saliency in conjunction with time-based weighting to create different depictions of motion (Fig. 4.5.6). Finally, we can also use anti-saliency weights to completely remove transient elements of the video clip



(a) Example video frames

(b) Video snapshot with saliency and time-sampling

(c) Video snapshot with saliency and linear time weights

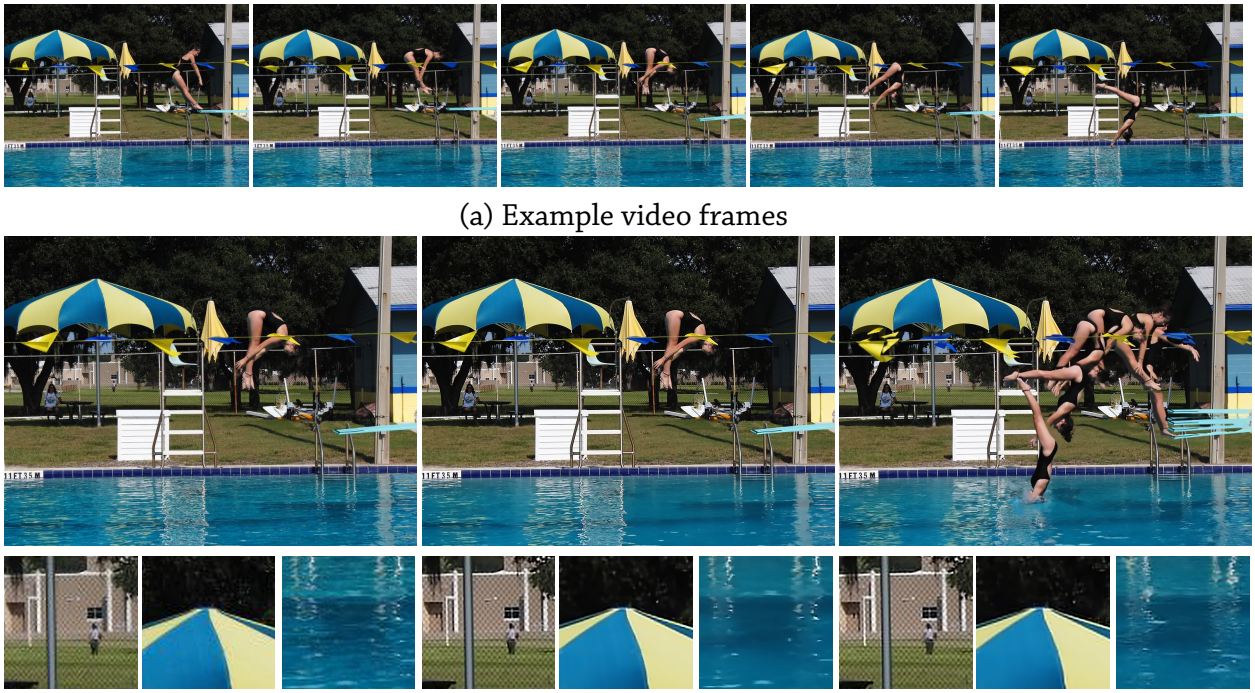
(d) Video snapshot with saliency and time overlaying

(e) Video snapshot with saliency and reverse overlaying

Figure 4.5.6: *Video snapshots with temporal effects. Our framework can create video snapshots with time-based effects. (a) In this clip of a bouncing ball (*11 *frames,* 960 × 540 *resolution), the input frames can be combined with (b) time-sampling weights to discretely sample some of the frames, with (c) temporal weights that increase linearly to emphasize the direction of motion, or with (d,e) weights that overlay each instant of the ball on top of the previous or next instances. Note that these weights only affect the ball.*

and produce high-quality snapshots of just the background (Fig. 4.5.8, and 4.5.9).

We have compared the quality of our results against single-image super-resolution and multi-image super-resolution. For single-image super-resolution, we compare against the work of Yang et al. [188], which uses a learned sparse dictionary of image patches to super-resolve images. As is expected, leveraging multiple frames almost always produces higher quality results than using a single image. For multi-image enhancement, we compare against the standard super-resolution technique of Irani and Peleg [86] that models the image formation process in a way that is similar to ours, and can be thought of as the standard approach to multi-image super-resolution without the use of our importance-based weights. By weighting the important pixels in the video appropriately, our framework produces snapshots with the same or better quality as standard multi-image super-resolution. We also compare our technique to a recent state-of-the-art video super-resolution method proposed by Liu and Sun [115]. This technique iteratively solves for the underlying motion, blur kernel and noise level while using a sparse image prior as well as priors



(a) Example video frames

(b) Single-image super-resolution (X 2) [188]

(c) Video snapshot (X 2) with motion confidence

(d) Video snapshot (X 2) with saliency, time-sampling and overlaying

Figure 4.5.7: *Video snapshots with motion. (a) This video clip of a diving girl (28 frames, 640 x 480 resolution) has noise and compression artifacts. (b) Single-image super-resolution [188] marginally improves the resolution but can not handle the noise and blocking artifacts. Our framework combines multiple images to upsample and denoise the reference frame. We do this while either (c) suppressing the motion, or while (d) summarizing the entire dive in the snapshot. Credit: Vimeo user* DHS Swim & Dive.

on the motion and kernel. Fig. 4.5.10 shows the results of this comparison for two datasets from their work. As can be seen from the results, when our assumption of approximately affine camera motion is met, our technique produces results that are qualitatively similar to those of Liu and Sun. In addition, our technique gives the user the freedom to go beyond basic enhancement, and depict interesting events and actions in the final snapshot.

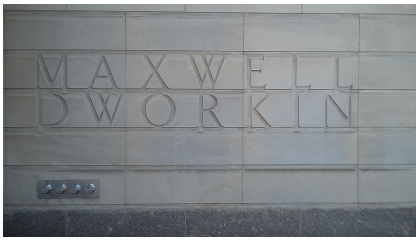
4.6 Summary

The camera used to photograph a scene is one of the factors that determines the appearance of the resulting image. In this chapter, we have presented an image formation model that explicitly rep-

(a) Example video frames

(b) Video snapshot (X 2) with motion confidence

(c) Video snapshot (X 2) with saliency and sampling



(d) Video snapshot (X 2) with anti-saliency

Figure 4.5.8: *Video snapshots with saliency weights. (a) In this video clip of a man walking (13 frames, 640 x 360 resolution), we can produce a high-quality image of the background while, (b) using motion-confidence weights to preserve the man is as in the original frame, (c) using saliency-based weights to summarize his motion, or (d) using anti-saliency weights to remove the man completely.*

(a) Example video frames

(b) Video snapshot (X 2) with saliency

(c) Video snapshot (X 2) with saliency, sampling, and overlaying

(d) Video snapshot (X 2) with anti-saliency

Figure 4.5.9: *Video snapshots with saliency weights. (a) This is video clip of traffic at a busy roundabout (20 frames, 640 x 360 resolution). (b) Using saliency weights produces a snapshot that captures all the vehicles in the video. However, because of the number of moving objects in the scene, this result looks crowded. (c) By using saliency with time-based effects, we can reduce this clutter. (d) We can also create a "clean-plate" snapshot of just the background by using anti-saliency weights. Credit: Vimeo user* Vietnam720.

(a) Input video frames | (b) Bicubic upsampling (X 4) | (c) Liu and Sun [115] (X 4) | (c) Video snapshot (X 4)

Figure 4.5.10: *Comparisons with multi-image super-resolution. When the camera motion is approximately affine, our technique produces results that are qualitatively similar to a state-of-the-art video super-resolution technique [115]. The camera zooms out and translates in the* `calendar` *sequence (top) and both techniques resolve the details. The* `foliage` *video (bottom) has a panning camera and scene motion. Using motion confidence weights suppresses this scene motion to produce a high-quality snapshot.*

resents the motion, blur, and noise characteristics of the camera. By inverting this model, we have shown that we can generate sharp, high-quality snapshots from lower resolution, lower quality videos. Our framework computes per-pixel weights based on temporal saliency, alignment, and local image statistics, and uses them to fuse aligned video frames. Our approach is flexible and can perform super-resolution, noise reduction, sharpening, and spatio-temporal summarization by changing only a few parameters. We believe this is a big step forward in increasing the ease with which users can create high-quality still photographs from short video clips. The importance of this work increases as the cost and effort of capturing video continues to decrease due to the availability of inexpensive, and portable consumer devices.

Our results suggest several areas for future work. While our approximation of camera motion using an affine transformation worked well for our video clips, motion estimation in complex videos is still a challenging task. As the alignment quality degrades, fewer samples can be aligned and averaged, reducing our method's ability to enhance image quality. We are investigating hierar-

chical motion estimation algorithms, e.g., Kang et al. [98], to address this issue. We are also investigating extensions of our importance-based weighting schemes to image enhancement methods that do not require explicit motion estimation [31, 150]. Extremely poor quality videos pose a challenge to our system because very high noise levels and compression artifacts corrupt both the alignment as well as the importance measures.

In addition to the weights discussed in this work, there are other weights that would be interesting to use in our framework, such as resampling / distortion weights [95, 167]. Using feature detection methods, one could also automatically find weights that indicate the presence of faces, smiles, and open / closed eyes. Our framework is general and allows any type of importance weights and user-defined combinations thereof to be used to create compelling video snapshots. Our importance-based enhancement can also be generalized to any application that involves a linear processing of video pixels. In the future we would like to investigate applications such as image stitching and compositing. It would also be interesting to perform some of our processing in the gradient domain; certain enhancements, e.g., removing blocking artifacts in compressed videos, could benefit from the seamless edits that are possible with gradient domain methods.

Lastly, our final snapshots are based on a user-specified reference-frame. This could be replaced by an algorithm that automatically selects "good" reference frames (e.g., Fiss et al. [67]) based on factors such as image quality and scene semantics.

5

Appearance Changes in Outdoor Scenes

The previous chapters have studied the role of factors such as geometry and the camera in the image formation process. In the next two chapters, we analyze the effect of changing illumination on image appearance; in this chapter, we focus on outdoor scenes captured under natural illumination. In an extended image sequence of an outdoor scene, one observes changes in appearance induced by variations in the illumination. We propose a model for these temporal color changes and explore its use for the analysis of outdoor scenes from time-lapse video data. We show that the time-varying changes in direct sunlight and ambient skylight can be recovered with this model, and that an image sequence can be decomposed into

two corresponding components. The decomposition provides access to both radiometric and geometric information about a scene, and we demonstrate how this can be exploited for a variety of visual tasks, including color-constancy, background subtraction, shadow detection, scene reconstruction, and camera geo-location.

5.1 Introduction

The importance of video-based scene analysis is growing rapidly in response to the proliferation of webcams and surveillance cameras being shared world-wide. Most of these cameras remain static with respect to the scene they observe, and when this is the case, their acquired videos contain tremendous temporal structure that can be used for many visual tasks. Compression, video summarization, background subtraction, camera geo-location, and video editing are but a few applications that have recently prospered from this type of analysis.

While temporal patterns in webcam data have received significant attention, the same cannot be said of color patterns. Many webcams observe outdoor scenes, and as a result, the sequences they acquire are directly affected by changes in the spectral content of daylight. Variations in daylight induce color changes in video data, and these changes are correlated with the time of day, atmospheric conditions, weather, and camera geo-location and geo-orientation. Thus, one would expect the colorimetric patterns of outdoor webcam data to be an important source of scene information.

In this chapter, we present a model for outdoor image sequences that accounts for this time-varying color information, and exploits the spectral structure of daylight. We explicitly represent the distinct time-varying colors of ambient daylight and direct sunlight, and in doing so, we show how an image sequence can be decomposed into two corresponding components. The decomposition provides access to a wealth of scene information, which can be divided into two categories:

1. *Per-pixel illuminant color and material properties:* Temporal variations in illuminant color are

recovered separately at each scene point along with a color albedo. This provides a time-varying background model that handles cast shadows in a natural way. It also provides a trivial method for obtaining color-constant measurements of foreground objects, which is a challenging problem otherwise.

2. *Scene and camera geometry:* The model provides partial information regarding the orientation of scene surfaces relative to the moving sun. By combining this information with standard geometric constraints we can predict shadow directions, recover scene geometry, and locate and orient the camera in a celestial coordinate system.

5.2 Related Work

Recovering information about a scene, such as surface reflectances, geometry, and illumination from photographs of the scene is a long-standing problem in vision and graphics. Since this is under-constrained in many cases, one way of simplifying the problem is to vary a single parameter in the image formation process and capture multiple images. As stated earlier, in our case, the images are captured under (passively) varying illumination. This configuration of the acquisition process is similar to many other vision problems such as intrinsic image decompositions, photometric stereo, etc. In this section we will review a number of these problems and techniques that are relevant to our work. Since there are a number of applications that our work allows, we will also discuss previous work in these fields.

5.2.1 Outdoor scene modeling

Because of the practical difficulties in measuring the surface reflectances and geometry of outdoor scenes, most such work has been restricted to small objects and indoor scenes. One of the early attempts to explicitly model and render outdoor scenes is the work of Nimeroff et al. [128]. They render scenes under natural illumination by combining basis images which are pre-rendered using

measured geometry and reflectances and a set of basis illuminations. Yu and Malik [189] and Debevec et al. [48] have used measurements of the incident illumination, surface materials, and a 3-d model of the scene geometry to create photo-realistic images for arbitrary viewpoints and lighting conditions.

5.2.2 Outdoor time-lapse data modeling

Time-lapse sequences of outdoor scenes have been studied extensively in recent work. Matusik et al. [121] use time-lapse data to compute the reflectance field (or light transport) of a scene for a fixed viewpoint. This estimated light transport combines the effects of reflectance and shadows and can then be used to re-render outdoor scenes with very realistic relighting results. Most closely related to our work is that of Sunkavalli et al. [161], who propose a method for decomposing a color outdoor image sequence into components due to skylight illumination and sunlight illumination. Each of these two components is further factored into components due to reflectance and illumination that are optimized for compression and intuitive video editing. While this is related to our work, our motivation is quite different, and hence, so is our model. We employ a more physically accurate model that uses general linear color transforms—as opposed to the diagonal transforms their model reduces—and we make explicit assumptions about scene reflectance. This allows us to handle more general weather conditions and to recover explicit scene information such as illuminant colors, sun direction, camera position, etc. Jacobs and colleagues [90] collect a large database of outdoor time-lapse webcams and analyze the temporal patterns in the data. They also use these temporal patterns to geolocate these webcams [91]. Time-lapse data has also been used for radiometric [101] and geometric camera calibration [105] geometrically calibrate the camera by using the appearance of the sky. Lalonde et al. [104] demonstrate techniques to recover environment maps for outdoor illumination from time-lapse sequences and use them to transfer appearance and illumination. Time-lapse data has also been used to recover surface geometry by using photometric stereo techniques [1, 149, 152], or by leveraging cloudy weather [89, 107].

5.2.3 Color constancy

The goal of a computational color constancy algorithm is to extract an illuminant-invariant representation of an observed surface. Given a trichromatic (RGB) observation $\mathbf{I}^E$ acquired under unknown illuminant E, the aim is to predict the observation $\mathbf{I}^{E_o}$ that would occur under a canonical illuminant E_o. One can distinguish most color constancy algorithms along three different lines: the type of transform used for illuminant changes; the method used to estimate the transform for a given image; and whether the illuminant is homogeneous or varies throughout the scene.

Almost all existing methods model illuminant changes using 3×3 linear transforms $\mathbf{I}^{E_o} = \mathbf{M}^{E \to E_o} \mathbf{I}^E$ that are restricted to being diagonal or 'generalized diagonal' [64]. This restriction is important because it reduces the estimation problem from finding nine parameters of a general linear transform to finding only three diagonal entries. Restricting transforms to be diagonal or generalized diagonal (or even linear in the first place), implies joint restrictions on the sensors being employed, and the sets of illuminants and materials being observed [40]. General linear transforms are the least restrictive—and hence the most accurate—of the three. They are rarely used in practice, however, because robust methods for estimating nine parameters from an image do not yet exist. One of the contributions of our work is to show that by exploiting the colorimetric structure of outdoor images we can overcome this limitation and achieve reliable color constancy with general linear transforms.

Most color constancy algorithms also restrict their attention to scenes with a single illuminant. The task of deriving illumination-independent images of a scene under mixed lighting is addressed by Barnard et al. [11] and later by Ebner [53]. These two approaches can be considered local approaches, since they derive illumination for each scene element (e.g., a pixel). In our context, however, outdoor images are captured under a mixture of two different light sources: direct sunlight and ambient skylight. Moreover, both the spectral content and the intensities of these two light sources change over the course of the day. Nonetheless, we show that we can recover the

normalizing (general linear) transform parameters for any mixture of these two illuminants, and that we can do so independently for each pixel in each frame of an image sequence (see Fig. 5.4.1).

5.2.4 Daylight Illumination

Since it is the most important natural source of radiant energy, the spectral content of daylight has received much attention. In the 1960's and 1970's, many researchers conducted measurements of the spectral power distribution (SPD) of daylight in different countries, and came to the conclusion that all the different forms of daylight spectra have chromaticities that lie close to the 1931 CIE Planckian locus. The most cited of these studies is the work of Judd et al. [97], who show that most daylight SPDs are can be accurately estimated by a linear combination of three basis SPDs. This work forms the basis for the CIE daylight recommendations [185]. It is common to parametrize the daylight locus in terms of correlated color temperature (CCT), which corresponds to the temperature at which a blackbody radiator would emit radiation with the same spectra. The CCTs of ambient skylight and direct sunlight are generally distinct, and each varies with weather, location, time of day, and time of year [185]. The CIE illuminant D65 (with a CCT of 6500K) is the commonly used standard for daylight illumination at noon. Hernández-Andrés [81] measured the SPDs of daylight at one location for two years, and showed that using three bases as recommended by the CIE produces reconstructed SPDs that are colorimetrically indistinguishable from the measured SPDs.

In addition to its spectral content, the luminance distribution of daylight has been studied extensively. Daylight illumination consists of direct radiance from the sun and radiance that is scattered from the sky. Since the sun is close to a point light source, it can be modeled easily. The sky, on the other hand, has a more complex luminance distribution. The CIE luminance formula [42] is an analytical sky model for clear skies that has been used in computer graphics. Perez et al. [132] developed a five-parameter model for the luminance distribution of the sky that generalizes to all weather conditions. Each parameter has a specific physical effect on the sky distribution, making

this model more accurate than the CIE model. More recently, Preetham et al. [140] simplified the Perez model for fast rendering of outdoor scenes.

5.2.5 Camera location and orientation

Estimating the geographic location and orientation of a camera from a time-stamped image sequence has rarely been considered. Cozman and Krotkov [45] extract sun altitudes from images and use them to estimate camera latitude and longitude (geo-location), and Trebi-Ollennu et al. [172] describe a system for planetary rovers that estimates camera orientation in a celestial coordinate system (geo-orientation). Both systems assume that the sun is visible in the images. Recently, Jacobs et al. [91] presented a method for geo-location based on correlating images with satellite data, but geo-orientation was not considered. In our work, we recover the position of the sun indirectly by observing its photometric effect on the scene. This provides both geo-location and geo-orientation without the need for satellite data and without requiring the sun to be in the camera's field of view (see Fig. 5.4.3).

5.2.6 Background subtraction and foreground detection

The core of most methods for background subtraction is the maintenance of a time-varying probability model for the intensity at each pixel. Foreground objects are then detected as low-probability observations (e.g., [171]). These methods can be difficult to apply to time-lapse data, for which the time between captured frames is on the order of minutes or more. In these cases, the 'background' can change dramatically between frames as clouds pass overhead and shadows change, and these intensity variations are difficult to distinguish from those caused by foreground objects. Our work suggests that the structure of daylight can be exploited to overcome this problem and obtain a reliable background model from time-lapse data. By modeling the colors and intensities of both direct sunlight and ambient skylight over time, we can effectively

predict how each scene point would appear under any mixture of these two illuminants in any given frame. Not only does this provide a means to detect foreground objects, but it also ensures that we do not return false-positive detections on the shadows that they cast (see Fig. 5.4.2).

5.3 A Color Model for Outdoor Image Sequences

Since it is the most important natural source of radiant energy, the spectral content of daylight has received significant attention [185]. A variety of studies have shown that daylight spectra—including those of direct sunlight, ambient skylight, and combinations of the two—form a one-dimensional sub-manifold of spectral densities. When represented in chromaticity coordinates they form a 'daylight locus' that lies slightly offset from the Planckian locus of blackbody radiators. From a computational standpoint, it is often more convenient to represent daylight spectra in terms of a linear subspace and studies suggest that subspaces of two (or perhaps three) dimensions are sufficient.

As the spectral content of illumination changes, so does the color of an observed surface point. Restricting our attention to Lambertian surfaces and linear sensors, the trichromatic observation of any surface point under illuminant $E(\lambda)$ can be written as:

$$I_k = \sigma \int C_k(\lambda)\rho(\lambda)E(\lambda)d\lambda, \tag{5.1}$$

where $C_k(\lambda)$ and $\rho(\lambda)$ are the sensor and spectral reflectance terms, respectively, and σ is a geometric scale factor that accounts for the angular distribution of incident radiant flux relative to the orientation of the observed surface patch.

We will use the notation $\mathbf{I}(x,t)$ for a trichromatic (RGB) image sequence parametrized by (linearized) pixel location x and time t. We choose linear transforms as our model for the effects of illuminant changes and, informed by the discussion above, we assume that the subspace contain-

ing daylight spectra is two-dimensional[1]. According to this assumption, the observation of any given material under any daylight spectral density (i.e., at any time of day and under any weather conditions) can be written as [40, 64]:

$$\mathbf{I}(x,t) = \left(\sum_{i=1}^{2} c_i(t)\mathbf{M}_i \right) \rho(x), \tag{5.2}$$

where $\rho(x)$ is an illumination-independent material descriptor, $\mathbf{M}_i$ are fixed 3×3 invertible matrices that span the allowable transforms (and more), and c_i are the coordinates of a particular color transform in this basis. In the next section, we combine this color model with geometry terms to produce a complete model for outdoor image sequences.

5.3.1 Incorporating shading

We assume that the sequence is captured by a fixed camera in an outdoor environment. For the moment, we also assume that the scene is static, that reflectance at scene points is Lambertian, and that the irradiance incident at any scene point is entirely due to light from the sky and the sun (i.e., mutual illumination of scene points is negligible.) Under these assumptions, the sequence can be written as:

$$\mathbf{I}(x,t) = \alpha(x,t) \left(\sum_{i=1}^{2} e_i^{sky}(t)\mathbf{M}_i \right) \rho(x) + \beta(x,t) \left(\sum_{i=1}^{2} e_i^{sun}(t)\mathbf{M}_i \right) \rho(x), \tag{5.3}$$

where $\rho(x)$ is the material descriptor of each surface point (assumed to be of unit norm), the terms in parentheses model the effects of time-varying spectra of ambient skylight and direct sunlight, and $\alpha(x,t)$ and $\beta(x,t)$ encode the effects of albedo and scene geometry. Since the sun is

[1] While some studies suggest that three dimensions are required, we have found that two are sufficient for our datasets.

a directional light source, we can write:

$$\beta(x,t) = V(x,t)a(x)\cos(\omega^{sun}t + \phi(x)), \tag{5.4}$$

where $a(x)$ is the albedo intensity, ω^{sun} is the angular velocity of the sun, $\phi(x)$ is the projection of the surface normal at a scene point onto the plane spanned by the sun directions (the *solar plane*), and $V(x,t) \in [0,1]$ models cast shadows. This last function will be binary-valued on a cloudless day, but it will be real-valued under partly cloudy conditions.

Similarly, the α term represents the surface reflectance integrated against the ambient sky illumination. Analytical forms for this are very difficult to estimate but for our datasets we have found that a low-frequency cosine works well. Therefore, we write this term as:

$$\alpha(x,t) = b(x)\cos(\omega^{sky}t), \tag{5.5}$$

where $b(x)$ combines the intensity $a(x)$ and the *ambient occlusion* which represents the fraction of the hemispherical sky that is visible to each point.

5.3.2 Model fitting

While the model in Eq. 5.3 is non-linear and has a large number of parameters, these parameters are overconstrained by the input data. For a time-lapse image sequence with P pixels and F frames, we have $3PF$ observations but only $PF + 5P + 4F$ degrees of freedom. In order to fit the model to an input sequence, we begin by recovering the color parameters (M_1, M_2, and $\rho(x)$) independent of intensity. This enables an initial decomposition into sun and sky components, which is then refined through a global optimization over the remaining parameters.

5.3.3 Material colors and a transform basis

From Eq. 5.2 it follows that the trichromatic observations $\mathbf{I}(x, \cdot)$ of a single pixel over the course of time will lie in a plane spanned by $\mathbf{M}_1\rho(x)$ and $\mathbf{M}_2\rho(x)$. A good estimate of this plane is found through a principal component analysis (PCA) of $\mathbf{I}(x, \cdot)$. The PCA yields color basis vectors $(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$ corresponding to the three eigenvalues $\sigma_1 \geq \sigma_2 \geq \sigma_3$. The plane we seek has $\mathbf{u}_3$ as its normal vector. Doing this separately at each pixel yields a set of F planes, which induce constraints on the materials and transform basis matrices:

$$\mathbf{u}_3(x)^\top(\mathbf{M}_1\rho(x)) = 0, \quad \mathbf{u}_3(x)^\top(\mathbf{M}_2\rho(x)) = 0. \tag{5.6}$$

These constraints do not uniquely determine the unknown parameters. Arbitrary invertible linear transformations can be inserted between $\mathbf{M}_i$ and $\rho(x)$, for example, and these correspond to changes of bases for the illuminant spectra and material spectral reflectance functions. These changes of bases are of no theoretical importance, but they do have practical implications. In particular, parameter choices for which the angle between $\mathbf{M}_1\rho(x)$ and $\mathbf{M}_2\rho(x)$ is small (for any scene point x) are poor because they will lead to numerical instabilities. A convenient method for choosing 'good' parameters is to find those that minimize the objective function:

$$\mathcal{O}(\mathbf{M}_i, \rho(x)) = \sum_{i=1}^{2} \sum_{x} ||\mathbf{M}_i\rho(x) - \mathbf{u}_i(x)||^2 \tag{5.7}$$

subject to the constraints in Eq. 5.6. Since $\mathbf{u}_1(x)$ and $\mathbf{u}_2(x)$ are orthonormal for all x, this ensures numerical stability in the subsequent analysis, and since $\mathbf{u}_1(x)$ is the dominant color direction at each scene point, it effectively chooses bases for the space of illuminants and spectral reflectances such that $M_1\rho(x)$ is close to the mean color of the sequence.

When the scene contains foreground objects, interreflections, and non-Lambertian surfaces, estimates of the color plane for each pixel (i.e., the normals $\mathbf{u}_3(x)$) can be corrupted by outliers. In

these cases, we have found that enforcing Eq. 5.6 as hard constraints yields poor results. A better approach is to perform an unconstrained minimization of the objective function in Eq. 5.7, which already has a soft version of the constraints 'built in'.

5.3.4 The shadow function

Central to the decomposition into sun and sky components is the estimation of the shadow function $V(x,t)$, which indicates whether the sun is visible to a scene point in a given frame. This function can be recovered by simultaneously exploiting the differences between the color and intensity of sunlight and ambient daylight. For the moment, we assume that $V(x,t)$ is a binary function.

The material vectors $\rho(x)$ and the transform basis $\{\mathbf{M}_1, \mathbf{M}_2\}$ define a color plane for each pixel, and by projecting the observations $\mathbf{I}(x,t)$ onto these planes we obtain the coefficients $\mathbf{c}(x,t) = (c_1(x,t), c_2(x,t))$ of Eq. 5.2. For a given pixel, the coefficients $\mathbf{c}(x,\cdot)$ provide a description of that pixel's color and intensity over time. Due to the differences between sunlight and skylight, the coefficients $\mathbf{c}(x,\cdot)$ will generally form two separate clusters corresponding to the times when the scene point is lit by the sun and, those when it is not (Fig. 5.3.1(d)). We observe that the clusters differ in both intensity (distance from the origin) and color (polar angle). Using the cluster centers $\mathbf{c}^{sky}(x)$ and $\mathbf{c}^{sun}(x)$, we label a pixel as 'in shadow' or 'lit by the sun' on the basis of the distances $d^{sky}_{x,t} = ||\mathbf{c}(x,t) - \mathbf{c}^{sky}(x)||$ and $d^{sun}_{x,t} = ||\mathbf{c}(x,t) - \mathbf{c}^{sun}(x)||$. By applying a two-cluster k-means algorithm, we can define a decision boundary $\mathcal{B}(x)$ for whether the sun is visible to a scene point or not.

While we could use these per-pixel decision boundaries to recover the binary shadow function $V(x,t)$, the results can be significantly improved by exploiting temporal and spatial coherence. To do this, we construct a graph in which each space-time point (x,t) is a node and each node is connected to its six nearest space-time neighbors. We determine $V(x,t)$ as the binary labeling

that minimizes (globally) an energy function:

$$E(L) = \sum_{\mathcal{V}} D_{\mathbf{x},t}(L_{\mathbf{x},t}) + \sum_{\mathcal{E}_\mathbf{x}} S_{x_i x_j} + \sum_{\mathcal{E}_\mathbf{t}} S_{t_i t_j}. \tag{5.8}$$

The unary data terms in the energy function, $D_{x,t}(\cdot)$, measure the position of the coefficients $\mathbf{c}(x,t)$ relative to the decision boundaries $\mathcal{B}(x)$, and for the labels of sky and sun, are given by:

$$D^{sun}_{x,t} = \frac{1}{1+e^{-(d_{x,t})}}, D^{sky}_{x,t} = 1 - D^{sun}_{x,t}, \tag{5.9}$$

where the distance $d_{x,t} = 3d^{sky}_{x,t} - d^{sun}_{x,t}$. The spatial pairwise (smoothness) terms are based on Pott's model as follows:

$$S_{x_i x_j} = \mathcal{N}(\rho(x_i) - \rho(x_j), 0.02). \tag{5.10}$$

In addition, to ensure that the shadows are coherent over time, we use temporal smoothness terms:

$$S_{t_i t_j} = 0.2. \tag{5.11}$$

We recover the binary shadow function $V(x,t)$ by minimizing Eqn. 5.8 using a standard graph cuts algorithm [27]. The recovered binary shadow function $V(x,t)$ can be refined, for example, by updating the per-pixel cluster centers according to this labeling and repeating the graph-cuts procedure. In practice we have found this not to be necessary. Fig. 5.3.1 shows an example of our shadow detection algorithm on a typical pixel.

5.3.5 Remaining parameters

Points that are known to be in shadow determine the angular sky parameter ω^{sky} in Eq. 5.5. This parameter can be estimated robustly using a non-linear least-squares optimization. By subtracting the ambient component from the input sequence, we obtain an approximate 'direct sun' sequence

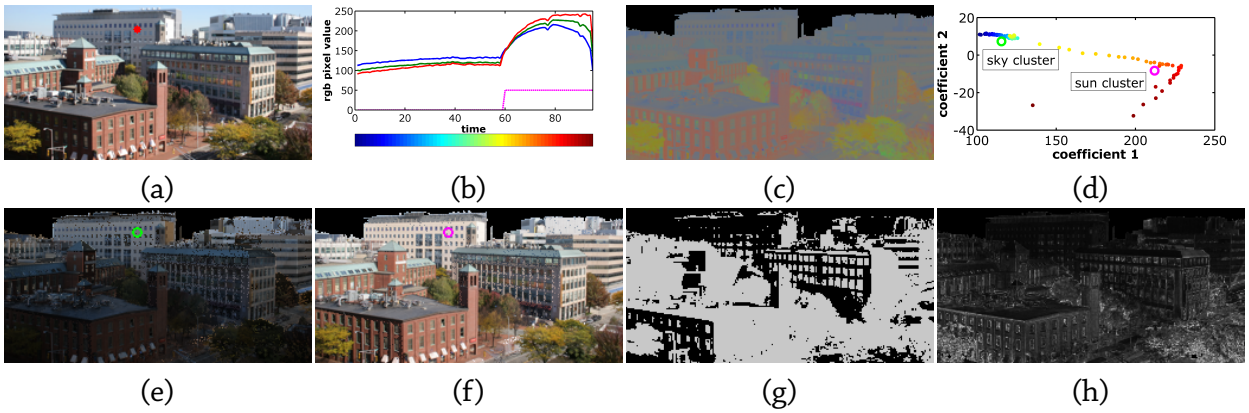


Figure 5.3.1: *Color and shadow initialization. (a) Frame 50 from the original time-lapse sequence. (b) RGB pixel values over time for the pixel indicated in (a. Since daylight spectra is low-dimensional, the time-varying color at each pixel lies in a plane in the color cube. A principal component analysis at each pixel allows us to recover each plane as well as a per-pixel normalized albedo (c). Projecting each pixel onto its dominant plane yields coefficients (d), shown with time coded using color from the colorbar in (b). These coefficients form two clusters that correspond to illumination from only ambient skylight (e) or by direct sunlight (f). Based on these clusters we can estimate a binary shadow function (g) (also shown for a single pixel as the magenta curve in (b)). (h) The ratio of the 3^{rd} to 2^{nd} eigenvalues at each pixel (scaled by 200). This is largest in regions of noise due to motion, foreground clutter etc., where the assumption of two-dimensional color variation for each pixels is violated.*

that can be used to estimate the angular sun velocity ω^{sun} in a similar fashion. Note that we need to consider only a small number of spatial points to recover these parameters.

Referring to Eq. 5.3, the remaining parameters to be estimated are the transform coefficients $e_i^{sky}(t)$ and $e_i^{sun}(t)$, the surface albedos $\rho(x)$, and normal angles $\phi(x)$. Similar to [161] we randomly initialize these parameters and then iteratively update each in order to minimize the RMS difference between the model and the input sequence. The coefficients $e_i^{sky}(t)$ and $e_i^{sun}(t)$ are updated using linear least squares, and the normal angles $\phi(x)$ are updated using a one-dimensional exhaustive search for each pixel.

As a final step, the binary shadow function is relaxed by finding the real-valued function $V(x,t) \in [0,1]$ that minimizes the RMS reconstruction error. This is an important step for any scene captured under partly cloudy conditions.

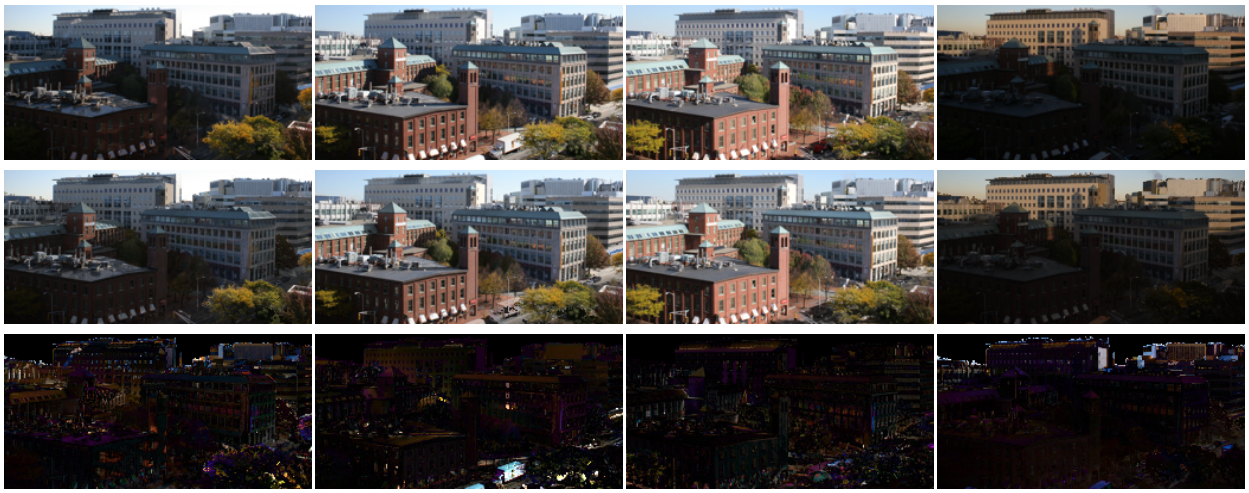


Figure 5.3.2: *Reconstructions from our model. Frames 1, 30, 60 and 95 from the original video (top), the reconstructions from our model (middle), and the absolute error scaled by 3 (bottom).*

Data	# Imgs	Resolution	RMS Error
Sunny square	95	130×260	6.42%
Cloudy square	120	240×360	7.36%

Table 5.3.1: *RMS reconstruction errors.*

5.3.6 Experimental results

Table 5.3.1 and Fig. 5.3.2 show results for two sequences obtained from Sunkavalli et al. [161]. These sequences consist of roughly the same scene in two different weather conditions (sunny and partly cloudy), and each sequence was captured over the course of one day with approximately 250 seconds between frames. The accompanying video shows these sequences in their entirety. It is important to note that the visible portions of the sky in our sequences were not considered in the decomposition; for all the results shown in this chapter, they have been copied from the original data to avoid distracting the reader.

In our results, errors are caused by foreground objects, smoke, interreflections from windows, and saturation of the camera. Another significant source of error is deviation from the assumed Lambertian reflectance model. From examining our data, it seems as though a rough-diffuse model [129] would be more appropriate.

Figure 5.4.1: *Color constancy using our model. Frames 1, 35, 94 and 120 from the original time-lapse data (top), and the corresponding images reconstructed with the sun and sky illuminant colors fixed to those of frame 35 (bottom).*

5.4 Implications for Machine Vision

The appearance of a scene depends on shape and reflectance, the scene illumination (both color and angular distribution), as well as the observer's viewpoint. Any visual task that requires some of this information seeks to recover it in a manner that is insensitive to changes in the others. By explicitly isolating many of these scene factors, our model enables novel approaches to some visual tasks and improves the performance of a number of others. Here we provide examples that relate to both color and geometry.

5.4.1 Color constancy

As mentioned in Sec. 5.2, most (single image) color constancy algorithms restrict their attention to diagonal or generalized diagonal transforms when representing changes in illumination. Even with this restricted model, estimating the transform parameters in uncontrolled environments is hard to do reliably. In contrast, once our model is fit to an image sequence, the task of color constancy becomes trivial. Since we obtain illuminant transform parameters separately for each frame and sun/sky mixing coefficients independently for each pixel, we can obtain illuminant-

Figure 5.4.2: *Simple foreground detection using per-pixel thresholds in color space. Frames 3, 41, 72, 88 from the original sequence with detected foreground pixels marked in red. Shadows cast by foreground objects are correctly ignored. Violations of our model (interreflections, saturated regions, etc.) trigger false positive responses.*

invariant descriptions everywhere simply by manipulating these parameters. Fig. 5.4.1 shows an example in which the color in each frame of the sequence is corrected so that the effective sky and sunlight colors are constant over the course of the day (they are held fixed to the colors observed in frame 35 of the sequence). Clear differences are visible between this and the original sequence, especially near dawn and dusk.

We emphasize that the color corrections are applied to the entire sequence, including the foreground objects. As a result, if one applies a color-based recognition algorithm to the color-corrected sequence instead of the original sequence, one can effectively obtain color-constant recognition with very little computational overhead. In addition, our use of general linear transforms can be expected to provide increased accuracy over what could be obtained using common diagonal or generalized diagonal transforms [40].

5.4.2 Background subtraction

Most background subtraction methods perform poorly when the illumination changes rapidly, for example, on a partly cloudy day. This problem is exacerbated in time-lapse data, where the time between frames is on the order of minutes, and the temporal coherence of foreground objects cannot be exploited. By modeling the entire scene over time, our model provides the means to handle these effects quite naturally. In particular, it immediately suggests two strategies for foreground detection. As noted earlier, the trichromatic observations $\mathbf{I}(x, \cdot)$ lie in the plane spanned

by vectors $\mathbf{M}_1\rho(x)$ and $\mathbf{M}_2\rho(x)$. Thus, one approach to foreground detection is simply to measure the distance between an observation $\mathbf{I}(x,t)$ and its corresponding spanning plane. This approach has the advantage of ignoring shadows that are cast by foreground objects, since cast shadow induce variations *within* the spanning planes. A second approach is to use the complete time-varying reconstruction as a background model and to use simple background subtraction for each frame. Fig. 5.4.2 shows the result of a combination of these two approaches, and shows how one can identify cars on the street without false positive responses to the shadows they cast or to the shadows cast by moving clouds. We do see detection errors in some areas, however, and these correspond to saturated image points, dark foreground objects with low signal-to-noise ratios, and inter-reflections from other buildings. Nonetheless, the detection results presented here suggest that our model will provide a useful input to a more sophisticated detection algorithm.

5.4.3 Scene geometry and camera geo-location

Our model provides direct access to the angular velocity of the sun ω^{sun} as well as the angles $\phi(x)$ in Eq. 5.4, which are one component of the surface normal at each scene point that corresponds to its projection onto the solar plane. This partial scene information can be combined with time-stamp information and common geometric constraints to recover scene geometry as well as the geo-location and geo-orientation of the camera.

Given three scene points x_i that are known to lie on three mutually orthogonal planes (two sides of a building and the ground plane for example), we can represent the normals $n_i = (\theta_i, \phi_i)$ in terms of spherical coordinates in a solar coordinate system (Z-axis is the normal to the solar plane and East is the X-axis). The azimuthal angles ϕ_i are equal to the corresponding $\phi(x_i)$ from our model up to a unknown, global additive constant. If each normal has a unique azimuthal component, our model gives two constraints on n_i in the form of the azimuthal differences $(\phi_{x_1} - \phi_{x_2})$ and $(\phi_{x_2} - \phi_{x_3})$. Combining these with mutual orthogonality constraints, the three normals are determined relative to the solar plane. (The same can be achieved from two orthogonal planes

True value	Estimate
42°21′57″N 71°05′33″W	42°12′58″N 70°05′47″W
t_{peak} = 16:25 UTC date = 10/27/06	t_{peak} = 16:30 UTC date = 10/28/06

Table 5.4.1: Camera geo-location results.

with the same albedo.)

If one of the recovered normals is the ground plane, the angle of the solar plane and, therefore, the peak (or meridian) altitude of the sun is uniquely determined. In addition, the projection of the ground plane normal onto the solar plane provides the azimuthal angle ϕ_{peak} of the sun's peak passage and East corresponds to the direction in the solar plane with azimuthal angle $\phi_{peak} - \pi/2$. Thus, by observing orthogonal planes over the course of a day, we can achieve the functionality of a combined compass and sextant.

Given the date and UTC time-stamps for each frame, we know the UTC time of the sun's peak passage (i.e., its meridian altitude) and can estimate both the latitude and longitude of the observed scene[2]. Likewise, if we know the latitude and longitude of the camera (and the season and year) we can reverse this process and compute the date and a UTC time stamp for the peak frame and propagate time stamps to all frames in the sequence using the time interval. Results of these analyzes for one of our sequences is shown in Table 5.4.1.

The meridian altitude of the sun was found to be 34.3°. Using the UTC time-stamps from the image sequence, this predicts a latitude and longitude that is only 83.7 km from the ground truth position. Alternatively, had we known the true geo-location of the camera, as well as the year and season of acquisition, we would have estimated a UTC time that differs from the true value by only 5 minutes and a date that deviates from the actual one by a day.

Finally, if we know the vanishing lines corresponding to the three scene planes, the camera

[2]The latitude and longitude of a location are uniquely determined by the time of the sun's peak passage, and can be looked up from a nautical almanac such as http://aa.usno.navy.mil/data/docs/AltAz.php

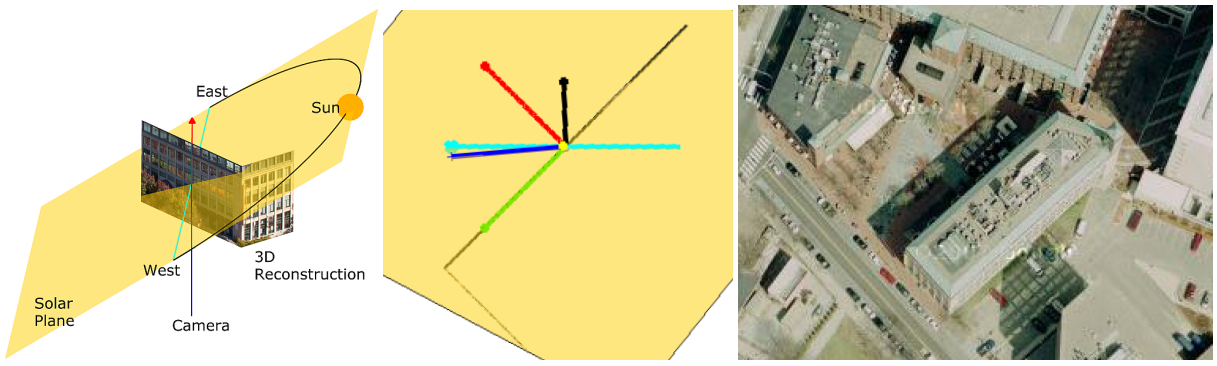


Figure 5.4.3: *Partial scene reconstruction and camera geo-location. By fitting our model to the input image sequence, we recover the orientation of the solar plane relative to the local horizon (a). When combined with time stamps, this determines the latitude and longitude of the camera as well as its orientation in an astronomical coordinate system. We also compare a top view of our reconstruction (with east-west axis in cyan and the walls' normals in red and green) (b) to a satellite image of the real building (with east-west corresponding to right-left) (c).*

Figure 5.4.4: *Shadow detection. The estimated shadow direction is marked in red for four frames from the time-lapse images.*

can be calibrated [77]. This yields the orientation of its optical axis relative to the solar plane, and in a celestial coordinate system. This achieves the functionality of a combined compass and inclinometer. A reconstruction of the scene is shown in Fig. 5.4.3. This includes the recovered solar plane, the orientation of the camera, and two reconstructed planes that are texture-mapped with the input frame that corresponds to the indicated sun direction.

5.4.4 Shadow prediction

Once the solar plane is known, we can determine the sun direction within that plane for each frame of a sequence. This can be used, for example, to predict a time-varying vanishing point on the image plane that corresponds to the direction in which vertical objects will cast shadows onto

the ground plane. If a vertical object (e.g., a person) is known to touch the ground plane at pixel x in a given frame, its shadow will lie on the line segment connecting x to the vanishing point of the shadow direction for that frame. This is demonstrated in Fig. 5.4.4, which shows predicted shadows vectors for some vertical objects that can be used for improved background subtraction.

5.5 Summary

In this chapter, we have proposed a model for outdoor illumination that exploits the colorimetric structure of extended outdoor image sequences. The model explicitly represents the time-varying spectral characteristics of direct sunlight and ambient skylight, and it allows an image sequence to be decomposed into distinct components that can be interpreted physically. The model can be reliably fit to time-lapse sequences in which the sun is visible for at least a fraction of a day; and once it is fit, it can be used for a variety of visual tasks. The examples presented in this chapter include color constancy, background subtraction, scene reconstruction and camera geo-location.

Our model could be improved by incorporating robust estimators into the fitting process, by using a more flexible reflectance model, and by making use of temporal patterns to appropriately handle 'time-varying textures' such as moving water and swaying trees.

There are a number of additional applications to be explored. By segmenting the scene according to albedo $\rho(x)$ and surface normal angle $\phi(x)$, one may be able to use orthogonality constraints to produce a coarse reconstruction of the scene. This type of scene geometry has proven to be a useful source of context information for object recognition. Also, since there is a one-to-one correspondence between coordinates in our illuminant transform space (e_i^{sky}, e_i^{sun}) and complete spectral densities in the daylight locus, it may be possible to use our model to infer information about air quality and other atmospheric conditions.

Finally, the model presented here relies only on images of an outdoor scene captured over the course of *o*ne single day. This limits the amount of scene information we can extract from the

data; for e.g., because of the planar motion of the sun, we are only able to extract one component of the surface normal. It would be interesting to extend our model to data captured over a longer period of time (and the sun's motion is non-planar), to recover high-quality surface normals. This is, in fact, a variant of the general Photometric Stereo problem that attempts to recover surface geometry by analyzing variations in appearance caused by changing illumination. In the next chapter, we will study this problem in detail, and in particular, analyze the effect of shadows on scene appearance.

6

Shadows and Scene Appearance

In Chapter 5, we presented a technique to recover scene properties like surface geometry, by analyzing the variations in the appearance of the scene caused by changes in illumination. While the model we presented was specific to scenes captured under outdoor illumination, similar representations have been explored for more general lighting conditions. In this chapter we consider the appearance of images of Lambertian scenes captured under moving directional lights. In particular, we examine the effect of shadows on scene appearance and propose a novel and robust algorithm for recovering surface geometry from these images.

6.1 Introduction

Photometric stereo is a class of techniques that seek to recover surface geometry from images of a scene captured under varying lighting. One specific instance of these techniques—Lambertian photometric stereo– is specifically derived for the case of Lambertian scenes imaged under directional lighting. In spite of being based on a crude reflectance model, Lambertian photometric is frequently used because it allows surface normal recovery even under uncalibrated lighting. In the ideal case, given a set of images under varying, but unknown, directional lighting, it is possible to recover both a surface normal field and the light source directions up to a three-parameter family of solutions [19, 190].

Like any photometric stereo technique, uncalibrated Lambertian photometric stereo relies on inverting the image formation process. It seeks to explain observations using combinations of light sources, surface normals, and surface albedos; and doing this accurately requires reasoning about the visibility of light sources with respect to each surface point. In Chapter 2, we used heuristics based on pixel intensities and colors to detect shadows. While such simple heuristics suffice in some cases, they are susceptible to error. In fact, this problem is deceptively hard because shadowing is a non-local function of surface geometry, and heuristics for shadow detection, such as simple thresholding, are unreliable in the presence of albedo variations and sparse input images.

In this work, we avoid explicit shadow detection by reasoning about illumination subspaces instead. It is well-known that the set of images of a convex Lambertian surface under directional lighting spans a three-dimensional linear subspace. It is also well-known that attached shadows and cast shadows violate this subspace property, so that the image-span of a scene with shadows can grow to a high dimension. What has not been fully exploited is that these high-dimensional spans have useful structure. We show that the image-span of any Lambertian scene captured under a discrete set of light sources with arbitrary shadowing can be decomposed into a set of

three-dimensional subspaces. We refer to these as *visibility subspaces* because they correspond to sets of surface points that can see a common set of lights.

Given a sequence of uncalibrated photometric stereo images of a Lambertian object, the visibility subspaces can be automatically identified—without knowledge of the lighting directions—using well-known subspace clustering techniques. We show that once these subspaces are identified, the surface is partitioned, the exact set of lights that is visible to each region can be computed, and the surface and light directions can be reconstructed up to the usual global linear ambiguity.

6.2 Related Work

Photometric stereo can produce per-pixel estimates of surface normals and is a common technique for scene reconstruction. Originally developed for Lambertian surfaces and calibrated directional lighting [182], photometric stereo has been generalized to handle uncalibrated directional lights [78], specular and glossy surfaces [71, 85, 125], symmetric reflectance functions [7, 84, 164], reflectance mixtures [83], and uncalibrated environment map lighting [16]. Despite these generalizations, Lambertian photometric stereo remains useful because of its simplicity and allowance for uncalibrated acquisition, as well as being an analytical "stepping stone" for developing more comprehensive techniques.

In order to obtain accurate reconstructions with any photometric stereo technique, Lambertian or not, one must identify shadowed regions in the images. Most approaches for isolating shadows rely on using enough light sources such that every surface point is illuminated by at least two or three of them, and then detecting and discarding intensity measurements having low values. The number of images may be as few as three or four [13, 44, 80] but can also be many more [183, 184]. Since these methods detect shadows by analyzing the intensities at individual pixels, they can be unreliable when a surface has texture with low albedo, and when cast shadows prevent some

surface points from being illuminated by a sufficient number of lights.

An alternative approach is proposed by Chandraker et al. [36]. They estimate which light sources can be seen by each surface point using a Markov random field in which the per-pixel "data term" is based on Lambertian photometric stereo and the "smoothness term" acts to encourage spatial coherence. This approach requires that the light directions are calibrated and known, and like the methods above, relies on reasoning about the intensities at each pixel. Our approach also derives from Lambertian photometric stereo, but unlike [36], does not require the light sources to be calibrated. Moreover, instead of reasoning about per-pixel intensities, it reasons about illumination subspaces.

Our work is also related to the problem of characterizing the structure of the set of a scene's images. There exist bounds on the dimension of the image-span of convex Lambertian scenes under directional lighting [151] and environment map lighting [15, 142], as well as convex scenes with a single arbitrary reflectance function [18] and mixtures of reflectance functions [70]. All of these bounds assume the scene to be convex so that cast shadows are absent. As a by-product of our analysis, we derive a complimentary bound that accommodates cast shadows and is valid for any Lambertian scene illuminated by a finite set of directional lights.

Finally, our work leverages insight from subspace clustering techniques, such as Generalized Principal Component Analysis (GPCA) [175] and Local Subspace Affinity (LSA) [186], that have been developed for motion segmentation. In our case, we perform subspace clustering using RANdom SAmple Consensus (RANSAC) [66, 170, 173]. This is quite different from a previous use of RANSAC in photometric stereo [82], which was aimed at identifying contour generators within an object's visual hull.

6.3 Visibility Subspaces

We begin with background and notation. For a Lambertian surface, the radiance from a surface point with normal $N \in \mathbb{S}^2$ and albedo ρ, illuminated with directional lighting L (i.e., with direction $L/||L|| \in \mathbb{S}^2$, and magnitude $||L||$), is given by $I = \max(0, \rho L^T N)$. In the absence of shadows, we know that $L^T N > 0$, and the image observations at m surface points illuminated by n light sources can be arranged as an $n \times m$ data matrix $\mathbf{I}$ that is the product of the $3 \times n$ lighting matrix $\mathbf{L} = [L_1, L_2, \cdots, L_n]$ and the $3 \times m$ albedo-scaled normals matrix $\mathbf{N} = [\rho_1 N_1, \rho_2 N_2, \cdots, \rho_m N_m]$:

$$\mathbf{I} = \mathbf{L}^T \mathbf{N}. \tag{6.1}$$

$\mathbf{L}$ and $\mathbf{N}$ are at most rank-three, and therefore, so is matrix $\mathbf{I}$ [151, 182].

If the scene is imaged under at least three non-coplanar light sources and these sources are calibrated and known, the surface normals can be estimated from noisy image intensities as $\mathbf{N} = (\mathbf{L}^T)^+ \mathbf{I}$ [182], where $(\cdot)^+$ is the pseudo-inverse operator. If the light sources are not calibrated, we can factor $\mathbf{I}$ using singular value decomposition (SVD) to recover the normals and lights using a rank-three approximation [78]:

$$\mathbf{I} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \quad \hat{\mathbf{L}}^T \triangleq \mathbf{U}_3 \mathbf{D}_3^{\frac{1}{2}}, \quad \hat{\mathbf{N}} \triangleq \mathbf{D}_3^{\frac{1}{2}} \mathbf{V}_3^T. \tag{6.2}$$

This determines the normals up to a linear 3×3 linear ambiguity such that:

$$\mathbf{L}^T = \hat{\mathbf{L}}^T \mathbf{A}, \mathbf{N} = \mathbf{A}^{-1} \hat{\mathbf{N}}. \tag{6.3}$$

for some non-singular matrix $\mathbf{A}$. This ambiguity can be resolved if light source intensities or surface albedos are known [78]. It can also be resolved up to the three-parameter generalized bas-relief ambiguity by enforcing an integrability condition on the normal field [19, 190].

Up to this point we have assumed the absence of cast and attached shadows, or equivalently, that every light source is *visible* to every surface normal. Now suppose that shadows exist, and consider the following toy example. A scene is partitioned into two uniform-visibility regions S_1 and S_2 that project to m_1 and m_2 pixels respectively. The scene is imaged under a set of n light directions that can be grouped into two (potentially) overlapping subsets $\mathbf{L}_1$ and $\mathbf{L}_2$, such that all of the lights $\mathbf{L}_1$ are visible to all points in S_1, and all of the lights $\mathbf{L}_2$ are visible to all points in S_2. Let the number of lights in these overlapping subsets be denoted by n_1 and n_2, and since they might overlap, we have $n_1 + n_2 \geq n$.

Now, the data matrix $\mathbf{I}$ can be permuted so that the first m_1 columns correspond to S_1 and last m_2 columns to S_2, and the first n_1 rows correspond to $\mathbf{L}_1$ and last n_2 rows to $\mathbf{L}_2$ with their shared lights lined up in the middle. Then, the observation matrix can be written as two sub-matrices, and if we denote by $\mathbf{N}_k$ the collection of surface normals in region S_k, the matrix can be factored as:

$$\mathbf{I} = \left[\mathbf{I}_1 \mid \mathbf{I}_2 \right] = \left[\begin{array}{c|c} \mathbf{L}_1^T & 0_{n-n_2}^T \\ 0_{n-n_1}^T & \mathbf{L}_2^T \end{array}\right] \begin{bmatrix} \mathbf{N}_1 & 0_{m_2} \\ 0_{m_1} & \mathbf{N}_2 \end{bmatrix}, \tag{6.4}$$

with 0_x representing a matrix of zeros with size $3 \times x$. The form of this factorization shows that while the row-space of $\mathbf{I}$ spans six dimensions, it actually consists of two rank-three subspaces corresponding to the two disjoint surface regions with different visibilities.

To generalize this to multiple regions with arbitrarily overlapping visibilities (i.e., sets of visible light sources), we define the *visibility vector* of region S_k to be the binary vector $V_k = [v_{k1}, v_{k2}, \cdots, v_{kn}]$, such that $v_{ki} = 1$ if light source L_i is visible to all the points in S_k and $v_{ki} = 0$ otherwise. The light sources visible to region S_k can then be expressed (with a slight change in notation from Eq. 6.4) as

$$\mathbf{L}_k = \mathbf{L} \otimes V_k, \tag{6.5}$$

where $\otimes$ represents the element-wise Hadamard product applied to every row of the lighting ma-

trix. As above, we can then factor the observation matrix for a scene with s distinct visibility regions as:

$$\mathbf{I} = \left[\mathbf{I}_1 \mid \mathbf{I}_2 \mid \cdots \mid \mathbf{I}_s \right] = \left[\mathbf{L}_1^T \mid \mathbf{L}_2^T \mid \cdots \mid \mathbf{L}_s^T \right] \begin{bmatrix} \mathbf{N}_1 & & & \\ & \mathbf{N}_2 & & \\ & & \ddots & \\ & & & \mathbf{N}_s \end{bmatrix}, \tag{6.6}$$

where $\mathbf{N}_k$ is the surface normal matrix corresponding to region S_k.

Thus, the observation matrix is made up of multiple subspaces, and we call these *visibility subspaces* because they correspond to regions in the scene that each have a consistent set of visible lights. Clearly, each subspace is at most rank-three, and the row space of a scene with s visibility subspaces has dimension at most $3s$. This leads us to the following:

Proposition. *The set of all images of a Lambertian scene illuminated by any combination of n directional light sources lies in a linear space with dimension at most* $3 \cdot 2^n$.

Proof: A scene illuminated by n light sources will have at most 2^n regions with distinct visibility configurations. The images of each region span at most a three-dimensional space, so the dimension of the image-span of the entire scene is at most $3 \cdot 2^n$.

This result is complementary to previous work that has established bounds on the dimensionality of scene appearance. Belhumeur and Kriegman [18] showed that the images of a scene with an arbitrary uniform BRDF, and illuminated by distant (environment map) lighting, lie in a linear space whose dimension is bounded by the number of distinct surface normals in the scene. Garg et al. [70] generalized this to spatially-varying reflectances that can be expressed as a linear combination of basis BRDFs. However, these results apply only to convex scenes without attached or cast shadows. In addition, these results assume that there are a finite number of normals in the scene to derive a bound on the dimensionality of scene appearance under arbitrary directional

(environment map) lighting. In contrast, our analysis provides bounds on the appearance of a Lambertian scene with an arbitrary number of normals but illuminated by a finite number of light sources, and allows any form of shadowing.

In general, we do not know the visibility subspaces of a scene *a priori*, and we cannot permute the rows and columns of the observation matrix to directly obtain the factorization in Eq. 6.6. However, as we show next, we can identify the subspaces automatically using a subspace clustering technique.

6.4 Estimating Visibility Subspaces

RANSAC [66] is a statistical method for fitting models of known dimensions to data with noise and outliers. While RANSAC is traditionally used to discard outliers from a dataset, we follow [173] and use it to cluster subspaces. In this context, it can be seen as an alternative to other subspace-estimation techniques, such as GPCA [175] and LSA [186].

Each visibility subspace of the scene is contained in a three-dimensional space. If we randomly choose three surface points that happen to be in the same region S_k, the light estimates $\hat{\mathbf{L}}_k$ that we obtain by factoring the image intensities at these three points (using Eq. 6.2) will accurately explain the intensities for all pixels in S_k. Thus, we expect a large number of "inliers". (Of course, there will be outliers as well because the points in the remainder of the scene will not have the same set of visible lights, and projecting their intensities onto $\hat{\mathbf{L}}_k$ will produce large errors.) Conversely, if we happen to choose three scene points that are in different regions, the light directions obtained by SVD will be unlikely to accurately explain the intensities at many other scene points, and we expect the number of inliers to be small. These observations suggest the following algorithm:

1. Choose three pixels at random and factor their intensities as $\mathbf{I}_3 = \hat{\mathbf{L}}_3^T \hat{\mathbf{N}}_3$.

2. Use lights $\hat{\mathbf{L}}_3$ to estimate the normal at all the surface points as $\hat{N}_i = (\hat{\mathbf{L}}_3^T)^+ I_i$.

3. Compute the per-pixel error of the estimated lights and normals as $E_i = ||I_i - \hat{\mathbf{L}}_3^T \hat{N}_i||^2$.

4. Mark points with error $E_i < \epsilon$ as inliers and recompute the associated optimal lighting $\hat{\mathbf{L}}_k$ using intensities for all inliers.

5. Repeat steps 1 through 4 for t iterations, or until a sufficiently large set of inliers has been found. During these iterations, keep track of the largest set of inliers found.

6. Mark the largest set of points that are inliers as a valid visibility subspace S_k with associated lighting basis $\hat{\mathbf{L}}_k$. Remove these inliers from the point set, and repeat steps 1 to 5 until all visibility subspaces have been recovered.

This procedure samples the points in the scene to find three points that belong to the same visibility subspace. Each time the sampling is successful, as measured by the number of inliers in Step 4, it extracts the subspace and removes it from the set of unlabeled points. The algorithm does not depend on the scene geometry or the lighting directions; it depends only on the rank-three condition of any visibility subspace. The result of the procedure is the set of per-pixel surface normals $\hat{\mathbf{N}}$, the per-pixel subspace labels $\mathbf{S}$, and a redundant (per-subspace) set of estimates for the light directions $\{\hat{\mathbf{L}}_k\}$. Note that in an uncalibrated setting, the set of normals for each subspace and their corresponding lights $\hat{\mathbf{L}}_k$ are defined up to their own linear ambiguity per Eqs. 6.2 and 6.3.

In our experiments, we use $t = 1000$ iterations, set the error threshold ϵ according to the noise in the input images, and run the procedure until 99% of the pixels are assigned to a valid visibility subspace. The remaining 1% of pixels are assigned to the subspace that best explains their intensity variation.

6.4.1 Degenerate Subspaces

The RANSAC-based method described above assumes that all visibility subspaces have rank-three. This is valid for any region having at least three non-coplanar surface normals, and illuminated by

at least three non-coplanar light sources. However, in general, scenes may contain rank-deficient subspaces that corrupt the clustering. Under the assumption that every point in the scene sees at least three non-coplanar lights (without which surface normal recovery is ambiguous), a visibility subspace can only be rank-deficient if it has degenerate normals: a region with coplanar normals will have rank two and a planar region will have rank one. Our task, then, is to check our recovered rank-three subspaces to see if they are composed of smaller degenerate subspaces.

Given the form of the observation matrix factorization in Eq. 6.6, it follows that a rank-three subspace can only be one of the following three types:

1. A region with a single visibility vector and non-coplanar normals (i.e., a true rank-three subspace).

2. Two regions with distinct visibility vectors, where one region has coplanar normals, and the other is planar (i.e., a combination of rank-two and rank-one subspaces).

3. Three regions with distinct visibilities, each of which is planar (i.e., a combination of three rank-one subspaces).

To ensure that our subspaces estimated by RANSAC are not of type 2 or type 3, we test every estimated rank-three subspace by searching for embedded rank-two and rank-one subspaces. If the number of pixels corresponding to the smaller embedded subspaces subsume more than a fraction α of the original set ($\alpha = 0.5$ in our experiments) we relabel them as being members of a different rank-deficient subspace.

6.5 Subspaces to Surface Normals

This subspace clustering identifies surface regions with uniform visibility, but does not provide a clean visibility vector V_k (or accurate shadows) for each region. Put another way, the non-visible entries of each $\hat{\mathbf{L}}_k$ are not necessarily zero-valued. To recover the visibility vectors and refine the

light matrices, we separately examine the light estimates in each subspace $\hat{\mathbf{L}}_k = [\hat{L}_{k1}, \hat{L}_{k2}, \cdots, \hat{L}_{kn}]$, and provided that the subspace is not degenerate, we set

$$v_{ki} = ||\hat{L}_{ki}^T|| > \tau, \tag{6.7}$$

with $\tau = 0.25$ in our experiments. This simple approach succeeds because the normals $\hat{\mathbf{N}}_k$ in each non-degenerate subspace span three dimensions, so the product $I_{ki} \approx \hat{L}_{ki}^T \hat{\mathbf{N}}_k$ can be zero only if the light strength $||\hat{L}_{ki}||$ is zero. Effectively, we are able to recover the visibility for each subspace by reasoning about the magnitude of the subspace lighting—an approach that is independent of scene albedo and is, therefore, not confounded by texture.

To estimate the visibility for degenerate subspaces, we first project the subspace lighting onto the column-space of the subspace normals before thresholding their magnitudes. This removes the component of the lighting orthogonal to the subspace normals that could be arbitrarily large while not contributing to the observed intensities.

Once the visibility vector for each subspace is known, we can recover the surface normals and reconstruct the surface. In the calibrated case, this is quite straightforward. Since the light sources $\mathbf{L}$ are known, they are combined with the visibility vectors using Eq. 6.5, and then the normals in every subspace are given by:

$$\mathbf{N}_k = (\mathbf{L} \otimes V_k)^+ \mathbf{I}_k \ , \ k = 1 \ldots s. \tag{6.8}$$

If the light sources are *not* calibrated, the situation is more complex because the subspace clustering induces a distinct linear ambiguity in each subspace, (i.e., $\mathbf{L}_k^T = \hat{\mathbf{L}}_k^T \mathbf{A}_k, \mathbf{N}_k = \mathbf{A}_k^{-1} \hat{\mathbf{N}}_k, k = 1 \ldots s$). Recovering the entire surface up to a single global ambiguity $\mathbf{A}$, which is the best we can do without additional information, requires that we somehow determine the transformations—one per subspace—that map each set of normals to a common coordinate system. Fortunately, this

can be achieved by solving the set of linear equations:

$$\hat{\mathbf{L}} \otimes V_k = \hat{\mathbf{L}}_k \mathbf{A}_k^T \ , \ k = 1 \dots s, \tag{6.9}$$

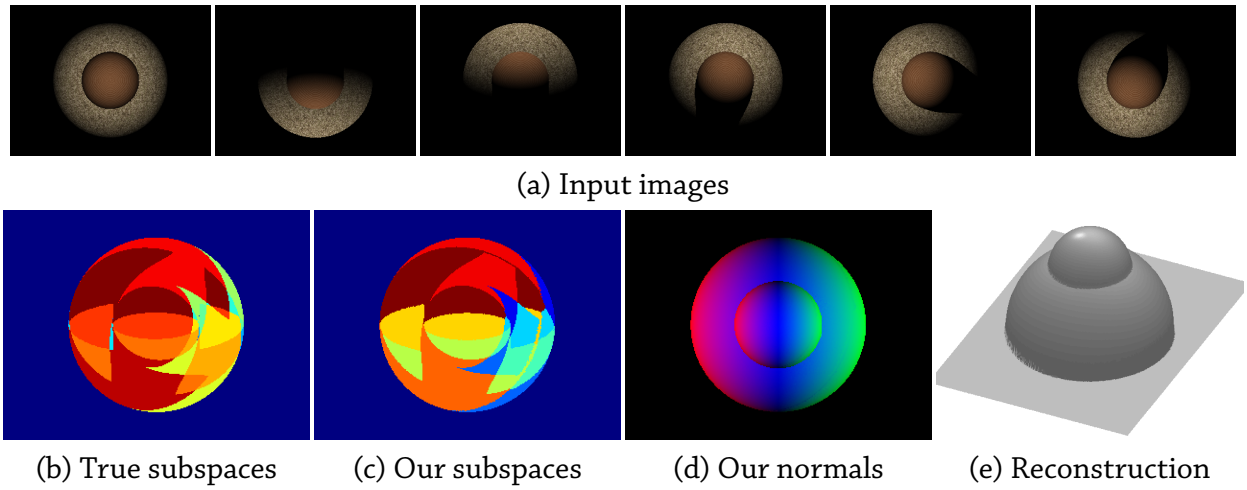
where both the global lights $\hat{\mathbf{L}}$ (i.e., those defined up to a single global ambiguity) and the per-subspace ambiguity matrices $\mathbf{A}_k$ are unknown. This is an over-constrained homogeneous system of linear equations since, for n lights and s subspaces, it contains $3ns$ constraints and $3n + 9s$ unknown variables. To avoid the trivial solution $\hat{\mathbf{L}} = \mathbf{A}_k = 0$ we set the ambiguity matrix for one reference subspace (chosen to be the non-degenerate subspace with the largest number of visible lights) to be the identity matrix. Accordingly, we recover the global lights $\hat{\mathbf{L}}$ and normals $\hat{\mathbf{N}}$ up to a single 3×3 ambiguity, which is that of the reference subspace.

To handle degenerate subspaces in the uncalibrated case, we first solve Eq. 6.9 using all non-degenerate subspaces, and as long as all of the global lights are visible to at least one of these regions, we can recover all of them. We then use these "auto-calibrated" lights to solve for the normals in the degenerate rank-one and rank-two subspaces using Eq. 6.8.

As a final step in the uncalibrated scenario, we may reduce or eliminate the global ambiguity using additional constraints, such as integrability of the normal field [19, 190], specular or glossy highlights [51, 71, 164], interreflections [37], or a prior model of object albedo [8, 153]. Then, in either calibrated or uncalibrated conditions, the estimated normals can be integrated to recover scene depth. In this integration process, one may optionally enforce the depth constraints that are induced by the visibility vectors and lights, and an elegant procedure for doing so can be found in [36].

6.6 Results

We evaluate the uncalibrated instantiation of our approach on two synthetic datasets and three captured datasets. In each case, we automatically cluster subspaces, determine visibility vectors,



(a) Input images

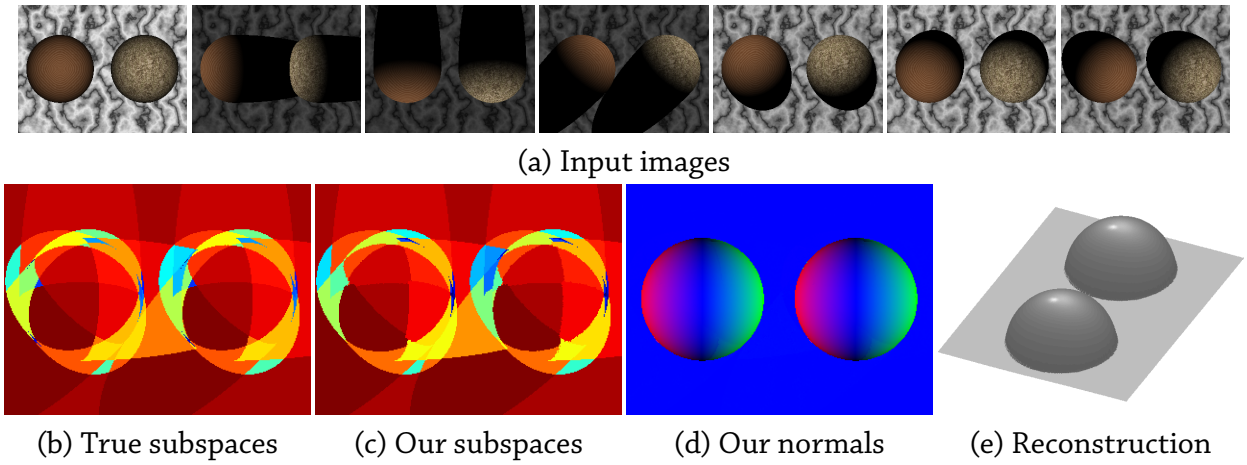
(b) True subspaces (c) Our subspaces (d) Our normals (e) Reconstruction

Figure 6.6.1: *Surface reconstruction for the spheres synthetic dataset. Attached and cast shadows divide this scene into intricate visibility subspaces (b). We are able to recover them almost perfectly (c), and estimate the surface normals (d) and depth (e) accurately.*

and compute lights and surface normals up to a global 3×3 linear ambiguity. As mentioned above, there are ways to resolve this ambiguity, and since this is not the focus of this work, we simply do so by manual intervention.

For synthetic examples, we evaluate the recovered normals, lights, and visibility subspaces by comparing them to the ground-truth values that are used to synthesize the input images. For the captured examples, the "true" values for comparison are obtained as follows. First, we acquire a dense set of calibrated photometric stereo images using approximately 50 different light directions. From such a dense set of calibrated images, we can robustly estimate surface albedos, and the image intensities can be reliably thresholded to detect per-pixel shadows and "true" visibilities. Then, we discard the shadowed measurements and recover the "true" normals via calibrated Lambertian photometric stereo. To make a direct comparison between this ground truth and our results, we execute our algorithm using a small subset of the dense input images, with the calibration information held out.

Figure 6.6.1 is a synthetic example in which the attached and cast shadows induce intricate visibility subspaces. From the six input images, our approach recovers the visibilities and normals



(a) Input images

(b) True subspaces (c) Our subspaces (d) Our normals (e) Reconstruction

Figure 6.6.2: *Surface reconstruction for the spheres and plane synthetic dataset. The shadows cast by the spheres on the plane create degenerate subspaces (b). We are able to disambiguate them and recover the visibility subspaces (c) and surface normals (d), and reconstruct the scene (e).*

almost perfectly. Figure 6.6.2 is a similar example, but in this case, the shadows cast on the back plane create degenerate visibility subspaces. These degenerate rank-one and rank-two subspaces are successfully detected by our approach, and the final visibilities and normals computed from the seven input images are again very close to ground truth. The median angular errors in surface normals for these two examples are 0.49° and 0.51°, respectively. Note that both of these synthetic scenes have high-frequency texture and large variations in albedos. These conditions often lead to poor results when using intensity-based shadow detection from such a small number of images, but this is not the case for the proposed method.

In the two captured datasets we consider—the frog (Fig. 6.7.1) and scholar (Fig. 6.7.2) sequences—our algorithm was given 8 and 12 input images, respectively. For each of these datasets, we capture images under densely-sampled calibrated lighting. We robustly estimate albedos from these dense image sets and threshold the images by a scaled albedo image to detect the ground truth visibilities. Using these visibilities and the calibrated light sources, we recover the "true" ground truth normals from these dense image sets, and use them as the reference for our results. We also compare the normals to those obtained using calibrated Lambertian photometric stereo applied to the same smaller set of (8 and 12) images that are available to our algorithm. We give

this algorithm access to both the calibrated light directions as well as the ground truth visibilities. We refer to these normals as the "best calibrated" normals because they can be interpreted as calibrated Lambertian photometric stereo supplied with "perfect" shadow detection, or equivalently, as the best-possible result from a calibrated shadow-detection method, such as [36, 183] applied on this small set of input images.

The input images have significant cast and attached shadows, and they exhibit non-idealities such as mutual illumination and slight specularity. Despite this, our method does reasonably well at locating the visibility subspaces (and shadows) from a small number of images. The median angular errors in the estimated normals (relative to the ground truth) are 7.44° and 4.45° for the frog and scholar datasets, respectively. The largest errors are made in regions with few non-shadowed measurements and where mutual illumination is most significant. This is not unique to our approach, however, and the errors from calibrated Lambertian photometric stereo with perfect shadow detection have a very similar structure. This suggests that our approach, which automatically handles shadows and is uncalibrated, introduces limited additional errors compared to an ideal calibrated algorithm.

Finally, we also compare our method to the calibrated photometric stereo technique of Chandraker et al. [36] (Fig. 6.7.3).

6.7 Summary

In this chapter, we have looked at the problem of recovering geometry by analyzing the variations in the appearance of a scene caused due to changes in illumination. This problem is especially hard in the presence of attached and cast shadows. Most previous techniques either require calibrated light sources, or detect shadows by using simple heuristics about image intensities at every pixel. In contrast to this, we show that regions of uniform visibility lead to subspaces that can estimated directly from image data without any prior knowledge. This insight has two major implications.

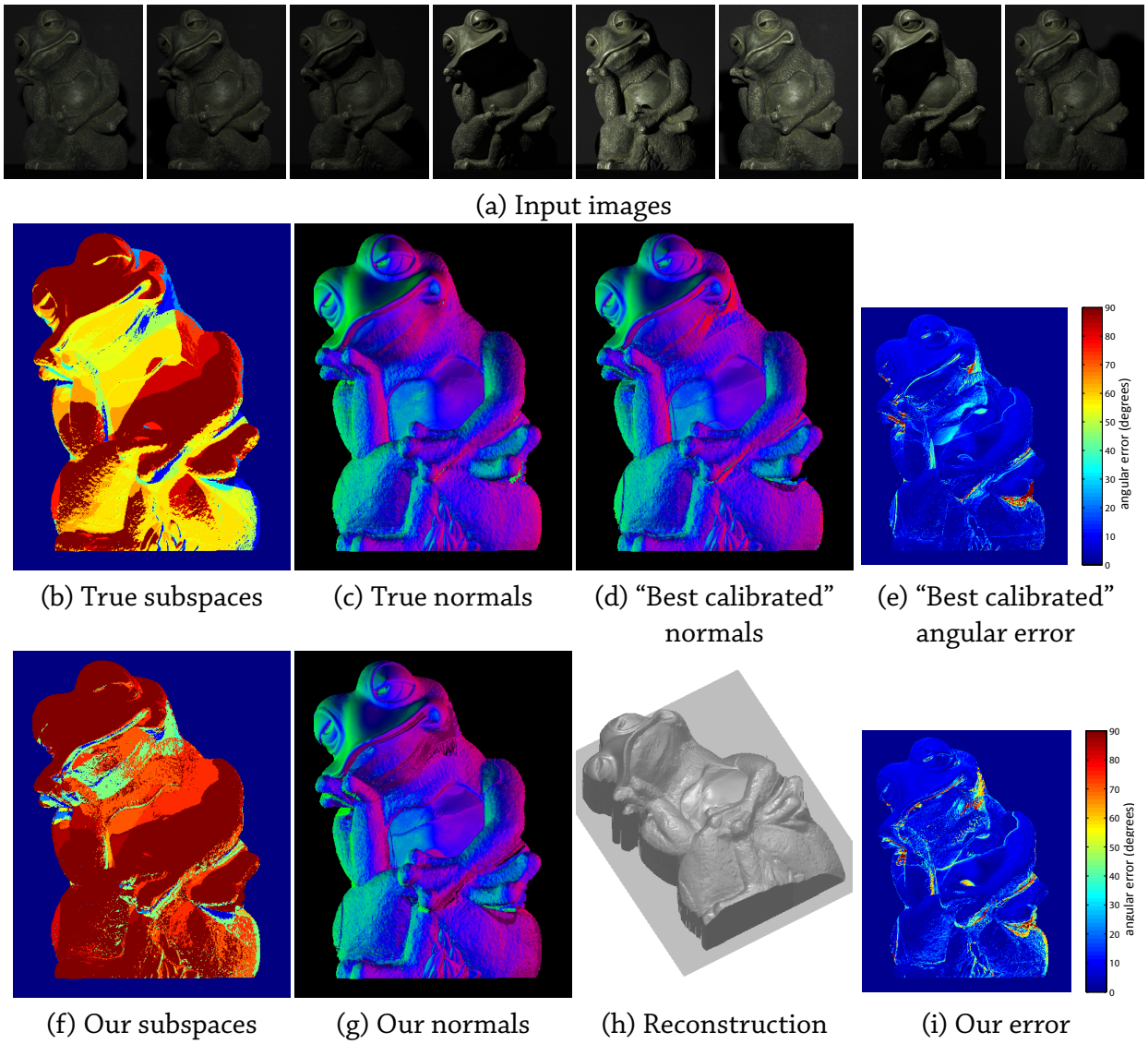
First, it leads to a novel bound on the dimension of the image-span of a Lambertian scene under a discrete set of lights, and this bound has the rare property of incorporating arbitrary shadowing. Second, it allows us to formulate shadow-detection in Lambertian photometric stereo as a subspace clustering task. This avoids heuristic reasoning about the intensities at individual pixels, and it allows handling cast and attached shadows in uncalibrated conditions when only a small number of input images are available. We have shown that we can obtain high-quality scene reconstructions even in real-world scenes with complex shadowing.

Unlike many previous approaches to shadow detection [36, 80], ours does not impose a preference for spatial coherence while detecting shadow regions. Indeed, we find that subspace clustering naturally leads to relatively coherent regions without this imposition. It is quite likely, however, that incorporating a spatial coherence constraint during subspace clustering could improve the results, especially in the presence of non-idealities like mutual illumination, and this may be a fruitful direction for future research.

Also, we have restricted ourselves to Lambertian scenes illuminated by directional lights, and it is worth considering how this analysis can be extended to handle more general conditions. One such extension would be to the ability to handle a combination of ambient illumination and directional lighting. It is known that the images of a Lambertian scene imaged under such a combination of light sources lie in a rank-four subspace [16, 190]. This would indicate that it might be possible to easily extend the techniques proposed in this chapter to ambient illumination by searching for rank-four subspaces in the image matrix. The natural extension of this would be photometric stereo algorithms under general environment map lighting [16], where a proper consideration of visibility would overcome the current (and severe) restriction to convex surfaces.

Finally, the RANSAC-based clustering algorithm presented in this chapter is robust to certain amount of non-idealities (for e.g., specularities, inter-reflections, noise, etc.). However, as the number of images increases and the scene gets more complex, the number of visibility subspaces and their overlap increases rapidly. Estimating surface normals accurately in such cases would

require robust subspace algorithms. One such possibility could be to estimate subspaces in a hierarchical fashion—i.e., to first estimate the subspaces from a small set of images, and then progressively refine them on the basis of new observations. Alternatively, instead of clustering individual surface points into clusters, we could over-segment the scene into representative regions and cluster these regions.



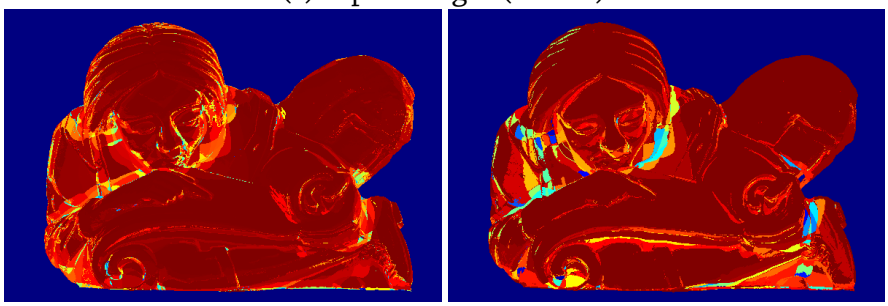
(a) Input images

(b) True subspaces (c) True normals (d) "Best calibrated" normals (e) "Best calibrated" angular error

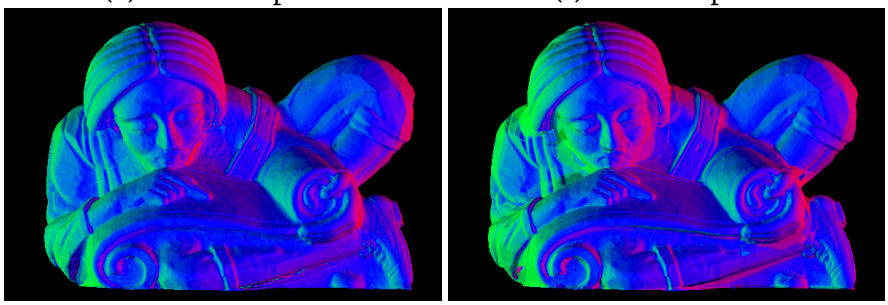
(f) Our subspaces (g) Our normals (h) Reconstruction (i) Our error

Figure 6.7.1: *Surface reconstruction for the frog dataset. Reconstruction results from sparse input images (a). Despite slight specularity and convexities with mutual illumination, our estimated subspaces (f) match the ground truth (b) reasonably well. The angular differences between our normals (f) and ground truth normals (c) are most significant in regions having few non-shadowed measurements (i). For comparison, the normals estimated using calibrated photometric stereo equipped with perfect shadow detection (d) exhibit similar deviations from the ground truth (e).*

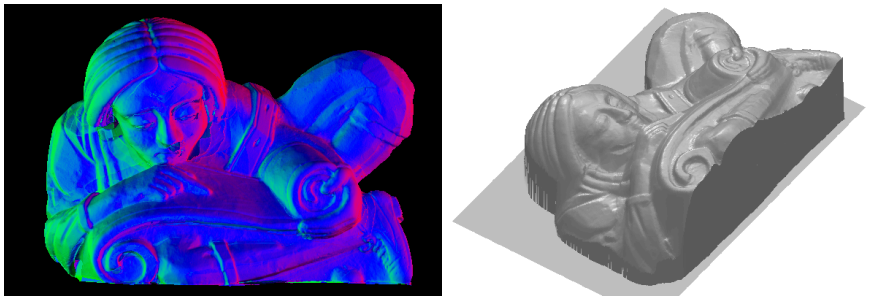
(a) Input images (8 of 12)



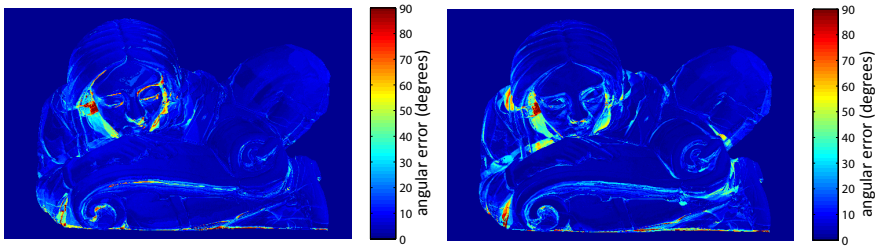
(b) True subspaces (c) Our subspaces



(d) True normals (e) Our normals

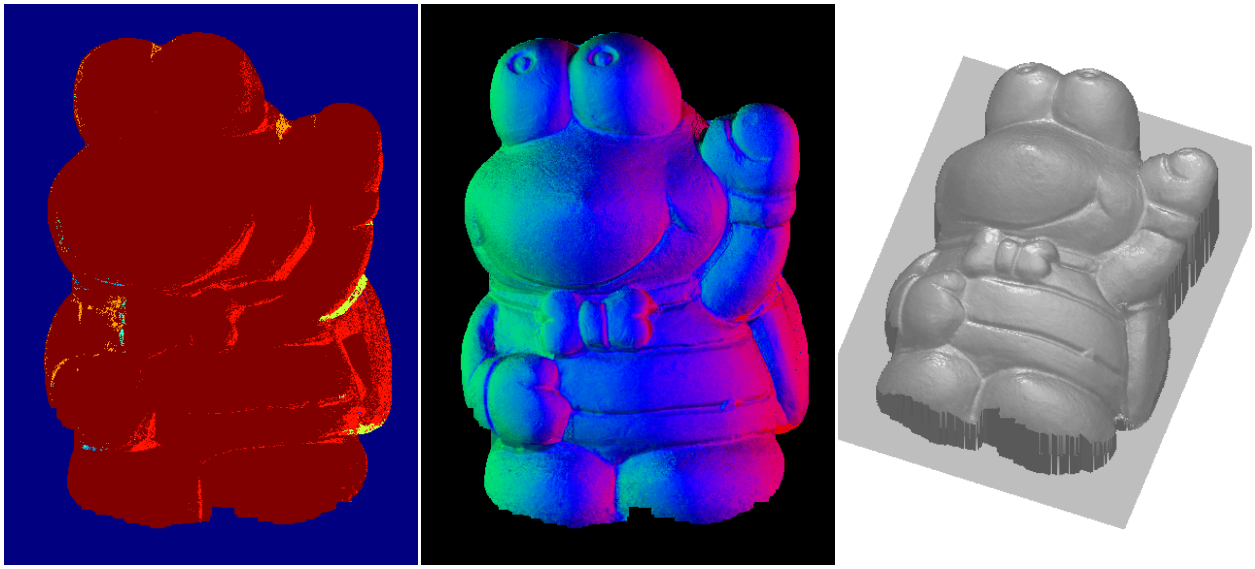


(f) "Best calibrated" normals (g) Reconstruction



(h) "Best calibrated" angular error (i) Our angular error

Figure 6.7.2: *Surface reconstruction for the scholar dataset. The left column shows ground truth (b,d) and normals obtained by calibrated photometric stereo applied to sparse input images (f). Our results with the same sparse set of images (a) are shown in the right column (c,e,g). The angular differences between the true normals (d) and our estimates (e) show that most errors are small and that large errors are restricted to small regions with strong inter-reflections (i). For comparison, the calibrated result (f) also exhibits similar deviations (h).*



(a) Estimated subspaces (b) Recovered normals (c) Recovered depth

Figure 6.7.3: *Visibility subspace estimation and normal recovery on data from Chandraker et al. [36].*

7

Summary and Future Directions

Modeling visual appearance is a fundamental goal in vision and graphics; while this dissertation makes a contributions towards this goal, our work is only a small step towards full-fledged image understanding. In this chapter, we summarize the contributions of our work and discuss directions for future research.

7.1 Summary

Images are formed as the result of a complex set of interactions between scene geometry and reflectance properties, illumination, and cameras; however, each of these factors varies in struc-

tured ways leading to a tremendous amount of coherence in image and video data. In this dissertation, we have proposed models of visual appearance that explicitly leverage this coherence to make analysis and editing tasks tractable. In particular, we have focused on two important goals:

1. Recovering scene properties from image and video data, and
2. Manipulating these properties to edit images and videos in intuitive ways.

With these goals in mind, we have looked at a number of vision and graphics tasks. These include:

Image and video compositing: The first application we discussed was blending and compositing images and videos that differ significantly in their appearance. In Chapter 2, we presented a multi-scale representation that leverages pixel correlations in natural images; we utilized it to transfer appearance between images by manipulating the statistical distributions of their pyramid decompositions. We also looked at the problem of face compositing in videos (Chapter 3) and used a multi-linear model for face geometry to track, align, and replace facial performances.

Enhancing low-quality images: In Chapter 4, we looked at the problem of creating a single high-quality snapshot from a video clip. We took advantage of the coherence in the images captured by a moving camera to invert the camera's imaging process. We incorporated importance weights corresponding to image features such as sharpness and saliency to produce snapshots that capture the activity in the video while improving the resolution, noise, and blur.

Analyzing illumination changes in image sequences: The third set of models we have presented analyze variations in appearance caused due to changes in illumination. In Chapter 5, we analyzed changes in outdoor scenes imaged over the course of a day. We proposed a novel model that leverages coherence in the temporal and colorimetric structure of natural illumination, and applied it to recover scene properties such as scene albedo and geometry. We demonstrated that this representation is particularly useful for visual tasks such as color constancy, background estimation, and camera geolocation. In Chapter 6, we analyzed the effect of shadows on Lambertian scene

appearance, and based on this analysis, presented a novel, robust photometric stereo algorithm.

7.2 Future Directions

There are number of interesting avenues for future work that we discuss below.

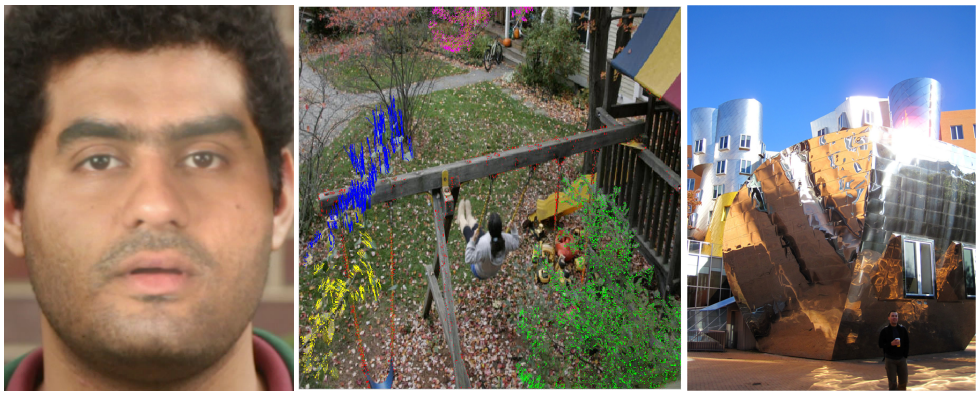
7.2.1 Modeling and editing other cues

Editing lighting: In Chapters 2 and 3, we were able to align the appearance of disparate images and videos to create photo-realistic composites. While our approach can handle differences in geometry, texture, noise, blur, etc., it does not compensate for differences in lighting. In fact, compositing images captured under widely different illumination conditions leads to results that look unnatural (see Fig. 7.2.1(a)). While progress has been made towards the problem of recovering and editing lighting in images (including some of the work in this dissertation), performing these tasks on images and videos captured in the "wild" still remains a challenging problem.

Editing motion: Motion plays a fundamental part in the way we perceive the world. Camera motion has been extensively used for tasks such as image enhancement (Chapter 4) and scene reconstruction [157], and motion estimation (or optical flow) techniques have a long history in computer vision [158]. However, these models can not be used to model, and subsequently edit motion in videos. While some recent work has started to look at this problem [41, 117, 147], we believe that this is an exciting area to work on.

7.2.2 Richer models for appearance

Full-fledged appearance modeling from a few images is a highly ill-posed problem, and one of the insights of this work is to use lower-dimensional models that make this more tractable. While the models we used are sufficient for a number of visual tasks, real-world scenes are often more



(a) (b) (c)

Figure 7.2.1: *Directions for future work. Editing lighting (a) and motion (b) realistically in images and videos remains a challenging problem. (c) Real-world scenes exhibit complex behavior that can be better explained by richer, more expressive models. Images credit: Liu et al. [117] (b) and Flickr user* `happy via` *(c).*

complex. For example, Fig. 7.2.1(c) shows a photograph depicting non-stochastic textures, buildings with intricate geometry and complex reflectances, and second-order illumination effects like inter-reflections and caustics. Explaining all these phenomena requires richer, more expressive models for appearance. This would in turn entail new inference algorithms that can robustly fit these richer models to image data. While researchers have already started exploring such algorithms for tasks such as scene reconstruction [16, 74] and reflectometry [145], many of these efforts are still confined to indoor laboratory conditions (the work of Lalonde [106] being an important exception). Extending such techniques to images captured under general conditions is a challenging problem with broad applications.

7.2.3 Leveraging more data

Estimating more general models of appearance as discussed above will certainly be very challenging. One way to make this problem better constrained is to make use of more image data. For example, richer models for outdoor scene appearance can be estimated by leveraging the large number of photographs already available online. Time-lapse videos of an outdoor scene captured over a large period of time (e.g., a year) or multiple images and videos captured from different

view-points could be used to recover high-quality appearance models. Alternatively, imaging devices can be modified to record data [2] that makes analysis and editing tasks easier. In addition, many devices today record auxiliary data (such as GPS coordinates, time-stamps) that can be potentially be exploited to inform image understanding algorithms.

7.2.4 Leveraging user interaction

Another way of making image modeling and editing more tractable is to involve the user in the loop. This approach has been used for a number of tasks, including, intrinsic image decompositions [26] and scene modeling [99]. Such approaches rely on the user to provide input that informs a computational image understanding algorithm. Of course, any approach that involves the user needs to answer questions about the amount of user involvement it requires (preferably, very little), and the interfaces for this interaction (ideally, highly intuitive).

References

[1] Ackermann, J., Ritz, M., Storck, A., Goesele, M.: Removing the example from photometric stereo by example. In: Proceedings of the ECCV Workshop on Reconstruction and Modeling of Large-Scale 3-D Virtual Environments (2010)

[2] Adams, A., Talvala, E.V., Park, S.H., Jacobs, D.E., Ajdin, B., Gelfand, N., Dolson, J., Vaquero, D., Baek, J., Tico, M., Lensch, H.P.A., Matusik, W., Pulli, K., Horowitz, M., Levoy, M.: The Frankencamera: an experimental platform for computational photography. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 29(4), 29:1–29:12 (Jul 2010)

[3] Agarwala, A.: Efficient gradient-domain compositing using quadtrees. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 26(3) (Jul 2007)

[4] Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., Cohen, M.: Interactive digital photomontage. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 23(3), 294–302 (Aug 2004)

[5] Aharon, M., Elad, M., Bruckstein, A.: The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing 54, 4311–4322 (Nov 2006)

[6] Alexander, O., Rogers, M., Lambeth, W., Chiang, M., Debevec, P.: The Digital Emily project: Photoreal facial modeling and animation. In: ACM SIGGRAPH 2009 Courses. pp. 12:1–15 (2009)

[7] Alldrin, N., Kriegman, D.: Toward reconstructing surfaces with arbitrary isotropic reflectance : A stratified photometric stereo approach. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1–8 (Oct 2007)

[8] Alldrin, N., Mallick, S., Kriegman, D.: Resolving the generalized bas-relief ambiguity by entropy minimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–7 (Jun 2007)

[9] Bae, S., Paris, S., Durand, F.: Two-scale tone management for photographic look. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 25(3), 637–645 (Jul 2006)

[10] Baker, S., Kanade, T.: Limits on super-resolution and how to break them. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(1), 1167–1183 (Sep 2002)

[11] Barnard, K., Finlayson, G.D., Funt, B.V.: Color constancy for scenes with varying illumination. Computer Vision and Image Understanding 65(2), 311–321 (Mar 1997)

[12] Barrow, H., Tenenbaum, J.: Recovering intrinsic scene characteristics from images (computer vision systems). In: Hanson, A., Riseman, E. (eds.) Computer Vision Systems, pp. 3–26. Academic Press (1978)

[13] Barsky, S., Petrou, M.: The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(10), 1239–1252 (Oct 2003)

[14] Bascle, B., Blake, A., Zisserman, A.: Motion deblurring and super-resolution from an image sequence. In: Proceedings of the European Conference on Computer Vision. pp. 573–582 (1996)

[15] Basri, R., Jacobs, D.: Lambertian reflectance and linear subspaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(2), 218–233 (Feb 2003)

[16] Basri, R., Jacobs, D., Kemelmacher, I.: Photometric stereo with general, unknown lighting. International Journal of Computer Vision 72(3), 239–257 (May 2007)

[17] Beeler, T., Hahn, F., Bradley, D., Bickel, B., Beardsley, P., Gotsman, C., Sumner, R.W., Gross, M.: High-quality passive facial performance capture using anchor frames. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 30(4), 75:1–75:10 (Aug 2011)

[18] Belhumeur, P.N., Kriegman, D.J.: What is the set of images of an object under all possible illumination conditions? International Journal of Computer Vision 28(3), 245–260 (Jul 1998)

[19] Belhumeur, P.N., Kriegman, D.J., Yuille, A.L.: The bas-relief ambiguity. International Journal of Computer Vision 35(1), 33–44 (Nov 1999)

[20] Bennett, E.P., McMillan, L.: Video enhancement using per-pixel virtual exposures. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 24(3), 845–852 (Jul 2005)

[21] Bickel, B., Botsch, M., Angst, R., Matusik, W., Otaduy, M., Pfister, H., Gross, M.: Multi-scale capture of facial geometry and motion. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 26(3) (Jul 2007)

[22] Bitouk, D., Kumar, N., Dhillon, S., Belhumeur, P., Nayar, S.K.: Face swapping: automatically replacing faces in photographs. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 27(3), 39:1–39:8 (Aug 2008)

[23] Blanz, V., Basso, C., Poggio, T., Vetter, T.: Reanimating faces in images and video. Computer Graphics Forum 22(3), 641–650 (2003)

[24] Blanz, V., Scherbaum, K., Vetter, T., Seidel, H.P.: Exchanging faces in images. Computer Graphics Forum 23(3), 669–676 (2004)

[25] Borshukov, G., Piponi, D., Larsen, O., Lewis, J., Tempelaar-Lietz, C.: Universal capture – Image-based facial animation for "The Matrix Reloaded". In: ACM SIGGRAPH 2003 Sketches & Applications (2003)

[26] Bousseau, A., Paris, S., Durand, F.: User-assisted intrinsic images. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia) 28(5), 130:1–130:10 (Dec 2009)

[27] Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9), 1124–1137 (Sep 2004)

[28] Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(11), 1222 –1239 (Nov 2001)

[29] Bradley, D., Heidrich, W., Popa, T., Sheffer, A.: High resolution passive facial performance capture. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 29(4), 41:1–41:10 (Jul 2010)

[30] Bregler, C., Covell, M., Slaney, M.: Video Rewrite: Driving visual speech with audio. In: Proceedings of ACM SIGGRAPH. pp. 353–360 (1997)

[31] Buades, A., Coll, B., Morel, J.M.: Nonlocal image and movie denoising. International Journal of Computer Vision 76, 123–139 (Feb 2008)

[32] Burt, P., Adelson, E.: The Laplacian pyramid as a compact image code. IEEE Transactions on Communications 31(4), 532–540 (Apr 1983)

[33] Burt, P.J., Adelson, E.H.: A multiresolution spline with application to image mosaics. ACM Transactions on Graphics 2(4), 217–236 (Oct 1983)

[34] Carroll, R., Ramamoorthi, R., Agrawala, M.: Illumination decomposition for material recoloring with consistent interreflections. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 30(4), 43:1–43:10 (Aug 2011)

[35] Chakrabarti, A., Scharstein, D., , Zickler, T.: An empirical camera model for internet color vision. In: Proceedings of the British Machine Vision Conference (BMVC) (2009)

[36] Chandraker, M., Agarwal, S., Kriegman, D.: Shadowcuts: Photometric stereo with shadows. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (Jun 2007)

[37] Chandraker, M., Kahl, F., Kriegman, D.: Reflections on the Generalized Bas-Relief ambiguity. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 1, pp. 788–795 (Jun 2005)

[38] Chatterjee, P., Milanfar, P.: Is denoising dead? IEEE Transactions on Image Processing 19(4), 895–911 (Apr 2010)

[39] Chen, J., Tang, C.K., Wang, J.: Noise brush: interactive high quality image-noise separation. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia) 28(5), 146:1–146:10 (Dec 2009)

[40] Chong, H., Gortler, S., Zickler, T.: The von Kries hypothesis and a basis for color constancy. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1–8 (Oct 2007)

[41] Chuang, Y.Y., Goldman, D.B., Zheng, K.C., Curless, B., Salesin, D.H., Szeliski, R.: Animating pictures with stochastic motion textures. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 24(3), 853–860 (Jul 2005)

[42] CIE-110-1994: Spatial distribution of daylight-luminance distributions of various reference skies. Color Research and Application 20(1), 80–81 (1994)

[43] Cohen-Or, D., Sorkine, O., Gal, R., Leyvand, T., Xu, Y.Q.: Color harmonization. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 25(3), 624–630 (Jul 2006)

[44] Coleman, E., Jain, R.: Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry. Computer Graphics and Image Processing 18(4), 309–328 (Apr 1982)

[45] Cozman, F., Krotkov, E.: Robot localization using a computer vision sextant. In: Proceedings of the IEEE International Conference on Robotics and Automation. vol. 1, pp. 106–111 (May 1995)

[46] Cutting, J.E.: Representing motion in a static image: constraints and parallels in art, science, and popular culture. Perception 31(10), 1165–1193 (2002)

[47] Davis, T.A.: Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2). Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2006)

[48] Debevec, P., Tchou, C., Gardner, A., Hawkins, T., Poullis, C., Stumpfel, J., Jones, A., Yun, N., Einarsson, P., Lundgren, T., Fajardo, M., Martinez, P.: Estimating surface reflectance properties of a complex scene under captured natural illumination (2004), USC ICT Technical Report ICT-TR-06.2004

[49] Debevec, P.: Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: Proceedings of ACM SIGGRAPH. pp. 189–198 (1998)

[50] DeCarlo, D., Metaxas, D.: The integration of optical flow and deformable models with applications to human face shape and motion estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 231–238 (Jun 1996)

[51] Drbohlav, O., Chaniler, M.: Can two specular pixels calibrate photometric stereo? In: Proceedings of the IEEE International Conference on Computer Vision. vol. 2, pp. 1850–1857 (Oct 2005)

[52] Dror, R.O., Willsky, A.S., Adelson, E.H.: Statistical characterization of real-world illumination. Journal of Vision 4(9) (2004)

[53] Ebner, M.: Color constancy using local color shifts. In: Proceedings of the European Conference on Computer Vision. pp. 276–287 (May 2004)

[54] Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image Processing 15, 3736–3745 (Dec 2006)

[55] Elad, M., Feuer, A.: Super-resolution reconstruction of image sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(9), 817–834 (Sep 1999)

[56] Essa, I., Basu, S., Darrell, T., Pentland, A.: Modeling, tracking and interactive animation of faces and heads using input from video. In: Proceedings of Computer Animation. pp. 68– (Jun 1996)

[57] Everingham, M., Sivic, J., Zisserman, A.: "Hello! My name is... Buffy" – automatic naming of characters in TV video. In: Proceedings of the British Machine Vision Conference. pp. 899–908 (2006)

[58] Ezzat, T., Geiger, G., Poggio, T.: Trainable videorealistic speech animation. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 21(3), 388–398 (Jul 2002)

[59] Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 27(3), 67:1–67:10 (Aug 2008)

[60] Farbman, Z., Hoffer, G., Lipman, Y., Cohen-Or, D., Lischinski, D.: Coordinates for instant image cloning. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 28(3), 67:1–67:9 (Jul 2009)

[61] Fattal, R., Agrawala, M., Rusinkiewicz, S.: Multiscale shape and detail enhancement from multi-light image collections. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 26(3) (Jul 2007)

[62] Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 25(3), 787–794 (Jul 2006)

[63] Finlayson, G.D., Hordley, S.D., Drew, M.S.: Removing shadows from images. In: Proceedings of the European Conference on Computer Vision. pp. 823–836 (May 2002)

[64] Finlayson, G.D., Drew, M.S., Funt, B.V.: Color constancy: generalized diagonal transforms suffice. Journal of the Optical Society of America A 11(11), 3011–3019 (Nov 1994)

[65] Finlayson, G.D., Drew, M.S., Lu, C.: Intrinsic images by entropy minimization. In: Proceedings of the European Conference on Computer Vision. pp. 582–595 (May 2004)

[66] Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (Jun 1981)

[67] Fiss, J., Agarwala, A., Curless, B.: Candid portrait selection from video. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 30, 128:1–128:8 (Dec 2011)

[68] Flagg, M., Nakazawa, A., Zhang, Q., Kang, S.B., Ryu, Y.K., Essa, I., Rehg, J.M.: Human video textures. In: Proceedings of the ACM SIGGRAPH Symposium on Interactive 3-D Graphics and Games. pp. 199–206 (2009)

[69] Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. IEEE Computer Graphics and Applications 22, 56–65 (Mar 2002)

[70] Garg, R., Du, H., Seitz, S., Snavely, N.: The dimensionality of scene appearance. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1917–1924 (Oct 2009)

[71] Georghiades, A.: Incorporating the Torrance and Sparrow model of reflectance in uncalibrated photometric stereo. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 816 –823 vol.2 (Oct 2003)

[72] Georgiev, T.: Photoshop healing brush: a tool for seamless cloning. In: Workshop on Applications of Computer Vision (ECCV 2004). pp. 1–8 (2004)

[73] Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 349–356 (Oct 2009)

[74] Goldman, D.B., Curless, B., Hertzmann, A., Seitz, S.: Shape and spatially varying BRDFs from photometric stereo. In: Proceedings of the IEEE International Conference on Computer Vision. vol. 1, pp. 341–348 (Oct 2005)

[75] Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2001)

[76] Guenter, B., Grimm, C., Wood, D., Malvar, H., Pighin, F.: Making faces. In: Proceedings of ACM SIGGRAPH. pp. 55–66 (Jul 1998)

[77] Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)

[78] Hayakawa, H.: Photometric stereo under a light source with arbitrary motion. Journal of the Optical Society of America A 11(11), 3079–3089 (1994)

[79] Heeger, D.J., Bergen, J.R.: Pyramid-based texture analysis/synthesis. In: Proceedings of ACM SIGGRAPH. pp. 229–238 (1995)

[80] Hernández, C., Vogiatzis, G., Cipolla, R.: Shadows in three-source photometric stereo. In: Proceedings of the European Conference on Computer Vision (2008)

[81] Hernández-Andrés, J., Romero, J., Nieves, J.L.: Color and spectral analysis of daylight in southern europe. Journal of the Optical Society of America 18(6), 1031–1036 (Jun 2001)

[82] Hernández Esteban, C., Vogiatzis, G., Cipolla, R.: Multiview photometric stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(3), 548–554 (Mar 2008)

[83] Hertzmann, A., Seitz, S.: Example-based photometric stereo: shape reconstruction with general, varying brdfs. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), 1254–1264 (Aug 2005)

[84] Holroyd, M., Lawrence, J., Humphreys, G., Zickler, T.: A photometric approach for estimating normals and tangents. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia) 27(5), 133:1–133:9 (Dec 2008)

[85] Ikeuchi, K.: Determining surface orientations of specular surfaces by using the photometric stereo method. IEEE Transactions on Pattern Analysis and Machine Intelligence 3(6), 661–669 (Nov 1981)

[86] Irani, M., Peleg, S.: Improving resolution by image registration. CVGIP: Graphical Models and Image Processing 53, 231–239 (May 1991)

[87] Itti, L., Baldi, P.: A principled approach to detecting surprising events in video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 1, pp. 631–637 (Jun 2005)

[88] Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(11), 1254–1259 (Nov 1998)

[89] Jacobs, N., Bies, B., Pless, R.: Using cloud shadows to infer scene structure and camera calibration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1102–1109 (Jun 2010)

[90] Jacobs, N., Roman, N., Pless, R.: Consistent temporal variations in many outdoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–6 (Jun 2007)

[91] Jacobs, N., Satkin, S., Roman, N., Speyer, R., Pless, R.: Geolocating static cameras. In: Proceedings of the IEEE International Conference on Computer Vision (Oct 2007)

[92] Jain, A., Thormählen, T., Seidel, H.P., Theobalt, C.: Moviereshape: Tracking and reshaping of humans in videos. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia) 29(5), 148:1–10 (Dec 2010)

[93] Jia, J., Sun, J., Tang, C.K., Shum, H.Y.: Drag-and-drop pasting. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 25(3), 631–637 (Jul 2006)

[94] Jones, A., Gardner, A., Bolas, M., McDowall, I., Debevec, P.: Simulating spatially varying lighting on a live performance. In: Proceedings of the European Conference on Visual Media Production. pp. 127–133 (Nov 2006)

[95] Joshi, N., Cohen, M.: Seeing Mt. Rainier: Lucky imaging for multi-image denoising, sharpening, and haze removal. In: IEEE International Conference on Computational Photography. pp. 1–8 (Mar 2010)

[96] Joshi, N., Matusik, W., Adelson, E.H., Kriegman, D.J.: Personal photo enhancement using example images. ACM Transactions on Graphics 29(2), 12:1–15 (Mar 2010)

[97] Judd, D.B., Macadam, D.L., Wyszecki, G., Budde, H., Condit, H., S.T.Henderson, Simonds, J.: Spectral distribution of typical daylight as a function of correlated color temperature. Journal of the Optical Society of America 54(8), 1031–1036 (1964)

[98] Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.: High dynamic range video. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 22(3), 319–325 (Jul 2003)

[99] Karsch, K., Hedau, V., Forsyth, D., Hoiem, D.: Rendering synthetic objects into legacy photographs. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia) 30(6), 157:1–157:12 (Dec 2011)

[100] Kemelmacher-Shlizerman, I., Sankar, A., Shechtman, E., Seitz, S.M.: Being John Malkovich. In: Proceedings of the European Conference on Computer Vision. pp. 341–353 (2010)

[101] Kim, S.J., Frahm, J.M., Pollefeys, M.: Radiometric calibration with illumination change for outdoor scene analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (Jun 2008)

[102] Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: image and video synthesis using graph cuts. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 22(3), 277–286 (Jul 2003)

[103] Lalonde, J.F., Efros, A.: Using color compatibility for assessing image realism. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1–8 (Oct 2007)

[104] Lalonde, J.F., Efros, A.A., Narasimhan, S.G.: Webcam clip art: appearance and illuminant transfer from time-lapse sequences. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia) 28(5), 131:1–131:10 (Dec 2009)

[105] Lalonde, J.F., Narasimhan, S.G., Efros, A.A.: What do the sun and the sky tell us about the camera? International Journal of Computer Vision 88(1), 24–51 (May 2010)

[106] Lalonde, J.F.: Understanding and Recreating Visual Appearance Under Natural Illumination. Ph.D. thesis, Carnegie Mellon University (2011)

[107] Langer, M., Zucker, S.W.: Shape-from-shading on a cloudy day. Journal of the Optical Society of America A 11, 467–478 (1994)

[108] Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 26(3) (Jul 2007)

[109] Levin, A., Zomet, A., Peleg, S., Weiss, Y.: Seamless image stitching in the gradient domain. In: Proceedings of the European Conference on Computer Vision (2004)

[110] Leyvand, T., Cohen-Or, D., Dror, G., Lischinski, D.: Data-driven enhancement of facial attractiveness. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 27(3), 38:1–38:9 (Aug 2008)

[111] Li, H., Adams, B., Guibas, L.J., Pauly, M.: Robust single-view geometry and motion reconstruction. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 28(5), 175:1–175:10 (Dec 2009)

[112] Li, Y., Sharan, L., Adelson, E.H.: Compressing and companding high dynamic range images with subband architectures. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 24(3), 836–844 (Jul 2005)

[113] Lin, Z., Shum, H.Y.: Fundamental limits of reconstruction-based superresolution algorithms under local translation. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(1), 83–97 (Jan 2004)

[114] Liu, C., Freeman, W.T.: A high-quality video denoising algorithm based on reliable motion estimation. In: Proceedings of the European Conference on Computer Vision. pp. 706–719 (2010)

[115] Liu, C., Sun, D.: A bayesian approach to adaptive video super resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 209–216 (Jun 2011)

[116] Liu, C., Szeliski, R., Kang, S.B., Zitnick, C.L., Freeman, W.T.: Automatic estimation and removal of noise from a single image. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(2), 299–314 (Feb 2008)

[117] Liu, C., Torralba, A., Freeman, W.T., Durand, F., Adelson, E.H.: Motion magnification. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 24(3), 519–526 (Jul 2005)

[118] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)

[119] Ma, W.C., Jones, A., Chiang, J.Y., Hawkins, T., Frederiksen, S., Peers, P., Vukovic, M., Ouhyoung, M., Debevec, P.: Facial performance synthesis using deformation-driven polynomial displacement maps. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia) 27(5), 121:1–10 (2008)

[120] Matsushita, Y., Nishino, K., Ikeuchi, K., Sakauchi, M.: Illumination normalization with time-dependent intrinsic images for video surveillance. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(10), 1336–1347 (Oct 2004)

[121] Matusik, W., Loper, M., Pfister, H.: Progressively-refined reflectance functions from natural illumination. In: Proceedings of the Eurographics Symposium on Rendering. pp. 299–308 (2004)

[122] Matusik, W., Pfister, H., Brand, M., McMillan, L.: A data-driven reflectance model. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 22(3), 759–769 (Jul 2003)

[123] McCann, J., Pollard, N.S.: Real-time gradient-domain painting. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 27(3), 93:1–93:7 (Aug 2008)

[124] Meer, P.: Robust techniques for computer vision. In: Medioni, G., Kang, S.B. (eds.) Emerging Topics in Computer Vision, chap. 4. Prentice Hall (Jul 2004)

[125] Nayar, S., Ikeuchi, K., Kanade, T.: Determining shape and reflectance of hybrid surfaces by photometric sampling. IEEE Transactions on Robotics and Automation 6(4), 418–431 (Aug 1990)

[126] Ngan, A., Durand, F., Matusik, W.: Experimental analysis of brdf models. In: Proceedings of the Eurographics Symposium on Rendering. pp. 117–226 (2005)

[127] Nicodemus, F.E.: Directional reflectance and emissivity of an opaque surface. Applied Optics 4(7), 767–773 (Jul 1965)

[128] Nimeroff, J.S., Simoncelli, E., Dorsey, J.: Efficient Re-rendering of Naturally Illuminated Environments. In: Proceedings of the Eurographics Workshop on Rendering. pp. 359–373 (1994)

[129] Oren, M., Nayar, S.K.: Generalization of the Lambertian model and implications for machine vision. International Journal of Computer Vision 14(3), 227–251 (Apr 1995)

[130] Park, S.C., Park, M.K., Kang, M.G.: Super-resolution image reconstruction: a technical overview. IEEE Signal Processing Magazine 20(3), 21–36 (May 2003)

[131] Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 22(3), 313–318 (Jul 2003)

[132] Perez, R., Seals, R., Michalsky, J.: All-weather model for sky luminance distribution – preliminary configuration and validation. Solar Energy 50(3), 235–245 (Mar 1993)

[133] Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7), 629–639 (Jul 1990)

[134] Pickup, L., Capel, D., Roberts, S., Zisserman, A.: Bayesian methods for image super-resolution. The Computer Journal (2007)

[135] Pighin, F., Szeliski, R., Salesin, D.: Resynthesizing facial animation through 3-d model-based tracking. In: Proceedings of the IEEE International Conference on Computer Vision. vol. 1, pp. 143–150 (1999)

[136] Pitie, F., Kokaram, A.C., Dahyot, R.: N-dimensional probablility density function transfer and its application to colour transfer. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1434–1439 (Dec 2005)

[137] Porter, T., Duff, T.: Compositing digital images. In: Proceedings of ACM SIGGRAPH. pp. 253–259 (1984)

[138] Portilla, J., Strela, V., Wainwright, M., Simoncelli, E.: Image denoising using scale mixtures of Gaussians in the wavelet domain. IEEE Transactions on Image Processing 12(11), 1338–1351 (Nov 2003)

[139] Portilla, J., Simoncelli, E.P.: A parametric texture model based on joint statistics of complex wavelet coefficients. International Journal of Computer Vision 40(1), 49–70 (Oct 2000)

[140] Preetham, A.J., Shirley, P., Smits, B.: A practical analytic model for daylight. In: Proceedings of ACM SIGGRAPH. pp. 91–100 (1999)

[141] Rabiner, L., Juang, B.H.: Fundamentals of speech recognition. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1993)

[142] Ramamoorthi, R., Hanrahan, P.: On the relationship between radiance and irradiance: determining the illumination from images of a convex Lambertian object. Journal of the Optical Society of America A 18(10), 2448–2458 (Oct 2001)

[143] Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Computer Graphics and Applications 21(5), 34–41 (Sep 2001)

[144] Robertson, B.: What's old is new again. Computer Graphics World 32(1) (2009)

[145] Romeiro, F., Vasilyev, Y., Zickler, T.: Passive reflectometry. In: Proceedings of the European Conference on Computer Vision. pp. 859–872 (2008)

[146] Roth, S., Black, M.: Fields of experts: a framework for learning image priors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 2, pp. 860–867 (Jun 2005)

[147] Rubinstein, M., Liu, C., Sand, P., Durand, F., Freeman, W.: Motion denoising with application to time-lapse photography. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 313–320 (Jun 2011)

[148] Ruderman, D.L., Bialek, W.: Statistics of natural image: Scaling in the woods. Physics Review Letters 73(6), 814–817 (1994)

[149] Sato, Y., Ikeuchi, K.: Reflectance analysis under solar illumination. In: IEEE Workshop on Physics-Based Modeling and Computer Vision. pp. 180–187 (Jun 1995)

[150] Shahar, O., Faktor, A., Irani, M.: Super-resolution from a single video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3353–3360 (Jun 2011)

[151] Shashua, A.: On photometric issues in 3-d visual recognition from a single 2-d image. International Journal of Computer Vision 21(1-2), 99–122 (Jan 1997)

[152] Shen, L., Tan, P.: Photometric stereo and weather estimation using internet images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1850–1857 (Jun 2009)

[153] Shi, B., Matsushita, Y., Wei, Y., Xu, C., Tan, P.: Self-calibrating photometric stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1118–1125 (Jun 2010)

[154] Simoncelli, E.: Statistical models for images: compression, restoration and synthesis. In: Proceedings of the Asilomar Conference on Signals, Systems and Computers. vol. 1, pp. 673–678 (Nov 1997)

[155] Simoncelli, E., Adelson, E.: Noise removal via Bayesian wavelet coring. In: Proceedings of the IEEE International Conference on Image Processing. vol. 1, pp. 379–382 (Sep 1996)

[156] Singular Inversions Inc.: FaceGen Modeller manual. www.facegen.com (2011)

[157] Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3-d. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 25(3), 835–846 (Jul 2006)

[158] Sun, D., Roth, S., Black, M.: Secrets of optical flow estimation and their principles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2432–2439 (Jun 2010)

[159] Sun, J., Jia, J., Tang, C.K., Shum, H.Y.: Poisson matting. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 23(3), 315–321 (Aug 2004)

[160] Sunkavalli, K., Johnson, M.K., Matusik, W., Pfister, H.: Multi-scale image harmonization. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 29(4), 125:1–125:10 (Jul 2010)

[161] Sunkavalli, K., Matusik, W., Pfister, H., Rusinkiewicz, S.: Factored time-lapse video. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 26(3), 101:1–101:10 (Jul 2007)

[162] Szeliski, R.: Locally adapted hierarchical basis preconditioning. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 25(3), 1135–1143 (Jul 2006)

[163] Takeda, H., Milanfar, P., Protter, M., Elad, M.: Super-resolution without explicit subpixel motion estimation. IEEE Transactions on Image Processing 18(9), 1958–1975 (Sep 2009)

[164] Tan, P., Zickler, T.: A projective framework for radiometric image analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2977–2984 (Jun 2009)

[165] Tappen, M.F., Freeman, W.T., Adelson, E.H.: Recovering intrinsic images from a single image. In: Advances in Neural Information Processing Systems. pp. 1343–1350 (Dec 2002)

[166] Tappen, M.F., Russell, B.C., Freeman, W.T.: Exploiting the sparse derivative prior for super-resolution and image demosaicing. In: IEEE Workshop on Statistical and Computational Theories of Vision (2003)

[167] Teodosio, L., Bender, W.: Salient video stills: content and context preserved. In: Proceedings of the ACM International Conference on Multimedia. pp. 39–46 (1993)

[168] Tipping, M.E., Bishop, C.M.: Bayesian image super-resolution. In: Advances in Neural Information Processing Systems. pp. 1279–1286 (2002)

[169] Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 839–846 (Jan 1998)

[170] Torr, P., Faugeras, O., Kanade, T., Hollinghurst, N., Lasenby, J., Sabin, M., Fitzgibbon, A.: Geometric motion segmentation and model selection. Philosophical Transactions: Mathematical, Physical and Engineering Sciences 356(1740) (May 1998)

[171] Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: Proceedings of the IEEE International Conference on Computer Vision. vol. 1, pp. 255–261 (1999)

[172] Trebi-Ollennu, A., Huntsberger, T., Cheng, Y., Baumgartner, E., Kennedy, B., Schenker, P.: Design and analysis of a sun sensor for planetary rover absolute heading detection. IEEE Transactions on Robotics and Automation 17(6), 939–947 (Dec 2001)

[173] Tron, R., Vidal, R.: A benchmark for the comparison of 3-d motion segmentation algorithms. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (Jun 2007)

[174] Tsai, R.Y., Huang, T.S.: Multiframe image restoration and registration. In: Advances in Computer Vision and Image Processing. vol. 1, pp. 317–339 (1984)

[175] Vidal, R., Hartley, R., Vidal, R., Hartley, R.: Motion segmentation with missing data by PowerFactorization and Generalized PCA. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 2, pp. 310–316 (Jun 2004)

[176] Viola, P., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision 57(2), 137–154 (May 2004)

[177] Vlasic, D., Brand, M., Pfister, H., Popović, J.: Face transfer with multilinear models. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 24(3), 426–433 (Jul 2005)

[178] Wang, J., Agrawala, M., Cohen, M.F.: Soft scissors: an interactive tool for realtime high quality matting. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 26(3) (Jul 2007)

[179] Weise, T., Li, H., Gool, L.V., Pauly, M.: Face/Off: Live facial puppetry. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. pp. 7–16 (2009)

[180] Weiss, Y.: Deriving intrinsic images from image sequences. In: Proceedings of the IEEE International Conference on Computer Vision. vol. 2, pp. 68–75 (2001)

[181] Williams, L.: Performance-driven facial animation. Computer Graphics (Proceedings of ACM SIGGRAPH) 24(4), 235–242 (Sep 1990)

[182] Woodham, R.: Photometric stereo: A reflectance map technique for determining surface orientation from image intesity. In: Proceedings of the SPIE Conference on Image Understanding Systems and Industrial Applications. vol. 155, pp. 136–143 (1978)

[183] Wu, T.P., Tang, C.K.: Photometric Stereo via Expectation Maximization. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(3), 546–560 (Mar 2010)

[184] Wu, T.P., Tang, K.L., Tang, C.K., Wong, T.T.: Dense photometric stereo: A markov random field approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(11), 1830–1846 (Nov 2006)

[185] Wyszecki, G., Stiles, W.: Color Science: Concepts and Methods, Quantitative Data and Formulae. Wiley New York, second edn. (2000)

[186] Yan, J., Pollefeys, M.: A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: Proceedings of the European Conference on Computer Vision. pp. 94–106 (2006)

[187] Yang, F., Wang, J., Shechtman, E., Bourdev, L., Metaxas, D.: Expression flow for 3-d-aware face component transfer. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 30(4), 60:1–60:10 (Aug 2011)

[188] Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution via sparse representation. IEEE Transactions on Image Processing 19, 2861–2873 (Nov 2010)

[189] Yu, Y., Malik, J.: Recovering photometric properties of architectural scenes from photographs. In: Proceedings of ACM SIGGRAPH. pp. 207–217 (1998)

[190] Yuille, A., Snow, D.: Shape and albedo from multiple images using integrability. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 158–164 (Jun 1997)

[191] Zhang, L., Snavely, N., Curless, B., Seitz, S.M.: Spacetime faces: high resolution capture for modeling and animation. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 23(3), 548–558 (Aug 2004)

[192] Zickler, T., Mallick, S.P., Kriegman, D.J., Belhumeur, P.N.: Color subspaces as photometric invariants. International Journal of Computer Vision 79(1), 13–30 (Aug 2008)