Intrinsic image decomposition from multiple photographs

Pierre-Yves Laffont

▶ **To cite this version:**

Pierre-Yves Laffont. Intrinsic image decomposition from multiple photographs. Graphics [cs.GR]. Université Nice Sophia Antipolis; INRIA Sophia-Antipolis, 2012. English. NNT : . tel-00761119

HAL Id: tel-00761119
https://theses.hal.science/tel-00761119

Submitted on 5 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITY OF NICE - SOPHIA ANTIPOLIS

DOCTORAL SCHOOL STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

P H D T H E S I S

to obtain the title of

PhD of Science

of the University of Nice - Sophia Antipolis

Specialty : COMPUTER SCIENCE

Defended by

Pierre-Yves LAFFONT

Intrinsic image decomposition from multiple photographs

Thesis Advisor: George DRETTAKIS

Co-Advisor: Adrien BOUSSEAU

prepared at INRIA Sophia Antipolis, REVES Team

defended on October 12, 2012

Reviewers :	Brian CURLESS	-	University of Washington
	Hendrik P. A. LENSCH	-	Universität Tübingen
Advisor :	George DRETTAKIS	-	REVES / INRIA Sophia Antipolis
Co-advisor :	Adrien BOUSSEAU	-	REVES / INRIA Sophia Antipolis
President :	Frédéric PRECIOSO	-	Polytech'Nice-Sophia
Examiner :	Diego GUTIERREZ	-	Universidad de Zaragoza

Acknowledgments

It is a pleasure for me to thank those who have made this thesis possible, and who contributed to making the past three years enjoyable!

My first and sincere thanks go to George Drettakis for giving me the chance to join his group, and for guiding me through this research. I learned a lot from him thanks to his deep involvement. I would also like to show my gratitude to Adrien Bousseau for his close supervision and the time he spent on our various technical debates. This thesis greatly benefited from his expertise. I feel lucky to have worked with such a pair of complementary and supporting advisors.

It has been an honor for me to collaborate with Frédo Durand and Sylvain Paris, starting with a summer visit at MIT which changed my view on research. I am also grateful to Maneesh Agrawala, for allowing me to experience a summer in Berkeley at an early stage of my PhD, and to Luc Robert and Emmanuel Gallo for the interesting collaboration with Autodesk. Also thanks to Brian Curless, Hendrik P.A. Lensch, Diego Gutierrez and Frédéric Precioso for participating in my thesis committee and sending interesting feedback.

I am grateful to the awesome colleagues I have had in the REVES team at INRIA, who were a pleasure to work with and all helped me in some ways. I would like to particularly thank Ares, Marcio and Nicolas, who put me early on the right track; Peter for his epic advice and incentive burgers; Adrien for the liquid units and Engineer bro-ness; Emmanuelle for her kindness and desserts; Gaurav for sharing his results and tips; Carles and Jorge for the fun capture sessions together; and Laurent for letting me write parts of this thesis while on the California roads.

Thanks also to my friends for all the good times, especially Arnaud, Benji, and Pi, for our many visits. I owe my deepest gratitude to Eunsun for warmly encouraging me to join this program, and for her long-lasting care. Last but not least, I would like to thank my parents and two sisters for their unconditional support and love throughout my degree.

Contents

CHAPTER 1

Introduction

(a) Source: Randall Warniers

(b) Source: James Kerr

Figure 1.1: *Even common scenes can lead to exceptional pictures when the shot captures special lighting, such as the shadow aligned with the pedestrian's footstep (a), or a beautiful sunrise (b).*

1.1 Motivation

Lighting is a key factor in successful photography. It sets the mood in a picture and affects the experience of the viewer. Lighting can convey feelings about a scene: in Figure 1.1b, the sun rising above a cloud casts an orange glow on the park, and produces a *warm* light in a winter morning. Taking the same shot in midday light would reveal the leafless trees and frozen ground, and would convey a colder ambiance.

Noticing the light and carefully planning for it is important for photographers. Some times of day and weather conditions are particularly good for taking pictures: during the *golden hours* around sunrise and sunset, the sun is low in the sky and produces a nice diffuse light. A bright sunny day with strong shadows is great for photographing architecture. Soft hazy light, when the sun is slightly obscured by clouds, is well-adapted for taking pictures of people.

However, taking a picture at the decisive moment is not easy. Natural light can change quickly, especially near sunrise and sunset (Figure 1.2), and it is often difficult to find a good timing for both nice lighting and proper organization of the scene. Instead, studio photographers control the lighting at the time of capture: they set up lights and reflectors to

Source: Tim Smalley

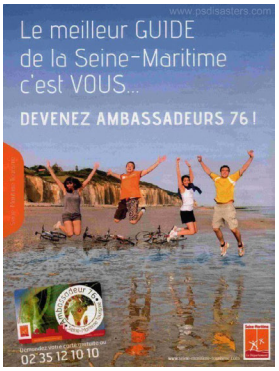
Figure 1.2: *Lighting can change quickly, especially during the "golden hours" around sunrise and sunset. This example shows photographs of a scene captured from similar viewpoints, within 10 minutes in the morning: the sky switches from characteristic dawn colors to a bland greyish dome in just a few minutes.*

enhance the appearance of the subjects. But such equipment is out of reach of casual photographers who only own a camera. For large outdoors scenes, it is essentially impossible to control lighting.

Consequently, the ability to manipulate the lighting *after* a photograph has been taken would simplify the capture process, and would allow for significantly more control on the final appearance of an image.

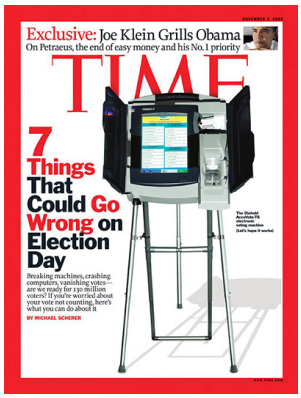
Image editing. Photographers often edit their digital pictures after the capture: they adjust the colors, enhance specific characteristics of the photographs, or manipulate their content to remove undesirable objects. Recently, the use of image editing software has become widespread and even casual photographers modify their photographs. Facebook reported that more than 200 million photos were uploaded per day in 2011 (source); a large proportion of these images has been edited.

Editing materials and lighting is a common image manipulation task that requires significant expertise to achieve plausible results. The photograph captured by a camera results from complex interaction between the incoming light and the scene, in particular its materials and geometry, which makes it difficult to edit manually. For example, changing the position of the sun affects the location and direction of shadows, and also the intensity of all pixels depending on their orientations (regions facing the sun should appear brighter). In addition, each pixel aggregates the effect of both material and lighting, but standard color manipulations are likely to affect both components simultaneously.



Source: PsDisasters

Source: The New York Times



Source: Time

Source: Time

Figure 1.3: *Recent examples of image manipulations which led to inconsistent lighting. Top row shows issues with reflections: on the left, the posture of jumping people does not match their reflections on the water surface; on the right, even though the wristwatch has been removed, its reflection on the table is still visible. Bottom shows shadow mismatches: the voting machine and the men seem to be floating over the ground, due to erroneous or missing shadows.*

Taking into account the lighting is important during image editing, because ignoring it, or modifying it incorrectly, yields images which look obviously manipulated. Figure 1.3 shows examples of such images, which exhibit non-coherent lighting. While plausible results can be achieved by skilled artists given enough time, these examples show that this step is often ignored due to its inherent difficulty.

In this thesis, one of our motivations is to create tools to simplify image editing, in particular "lighting-aware manipulations" which maintain coherent lighting in the edited image. Our approach builds on an image representation which decouples object materials from their illumination. We develop methods to separate the *intrinsic color* of the scene objects from the quantity of light they receive, and show how this can be used for advanced image manipulations.

Image-based rendering. Image-based rendering techniques exploit images of a scene and produce novel views, which enable interactive navigation in the virtual scene. Examples of such techniques include Google StreetView, which provides panoramic views from positions along many streets in the world, or Microsoft Photosynth and Google Photo Tours, which allow users to navigate in a scene in three dimensions. StreetView displays streets as they looked like at the time of capture; as a result, they can look drastically different from their current appearance in the real world. Photosynth combines images which are not necessarily captured at the same time of day; this can produce disturbing transitions when the lighting in successive images is very different.

The fact that image-based methods to date have been restricted to the lighting at the time of capture has seriously limited their utility in digital content creation. Providing the ability to modify the lighting in image-based captures will render such approaches much more attractive, and will open the way for the use of image-based assets in standard content creation pipelines. Allowing users to change the lighting as desired will also make applications such as virtual tourism much more immersive.

The work developed in this thesis allows the transfer of lighting across pictures of a photo collection, therefore enabling transitions with coherent illumination across views.

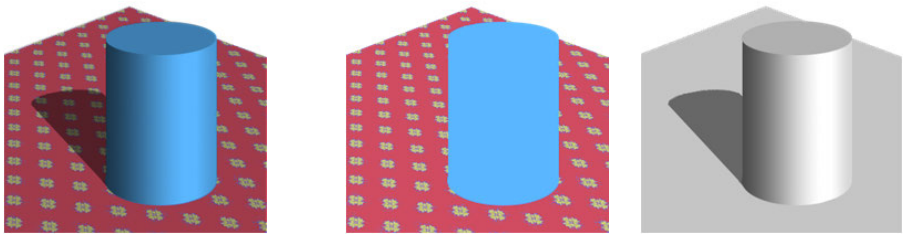
1.2 Context

We place ourselves in the context of tools related to extracting, removing or manipulating the lighting in a photograph. Our work builds on a central representation which separates the color of the materials from the received illumination, at each pixel of an input image. We first define this *intrinsic image* representation, then outline our approach in this thesis, and describe potential industrial applications.

1.2.1 Intrinsic images

Barrow and Tenenbaum [Barrow 1978] first proposed to describe a scene in terms of its intrinsic characteristics, such as surface orientation, reflectance, and illumination. Given one input image, they suggest to extract a set of *intrinsic images*, each representing one intrinsic characteristic at all the scene points visible in the original image. They motivate this separation by the fact that intrinsic characteristics give a more invariant and discriminating description of the scene than raw image colors. In addition, each intrinsic image can be accessed independently, which is particularly helpful for image understanding operations such as material recognition, image segmentation, or shape from shading.

Subsequent work has mostly focused on the problem of recovering two intrinsic images: the *reflectance* image (also called *albedo*), which corresponds to the material color at each point, and the *illumination* image (also called *shading*), which represents the ef-



(a) Input image $\mathbf{I}$ (b) Reflectance $\mathbf{R}$ (c) Illumination $\mathbf{S}$

Figure 1.4: *Illustration of intrinsic decomposition. Starting from a picture (a), here a synthetic image with direct illumination only, intrinsic decomposition yields two independent layers: reflectance (b) and illumination (c).*

fect of lighting at each point. More formally, a color photograph $\mathbf{I}$ is decomposed into a reflectance image $\mathbf{R}$ and an illumination image $\mathbf{S}$, so that:

$$\mathbf{I} = \mathbf{R} * \mathbf{S} \tag{1.1}$$

where $*$ denotes per-pixel and per-channel product.

Figure 1.4 illustrates the intrinsic decomposition of a simple synthetic image, which represents a cylinder with uniform color illuminated by a white directional light source. Each pixel in the input image (left) aggregates the effect of both material color and lighting; as a result, the color of the cylinder is not uniform even though it is made of a single material. Intrinsic decomposition yields two independent layers which, once recombined, yield the input image again. The reflectance layer looks "flat" because it does not contain shading effects and shadows, which provide important depth cues. In contrast, the illumination layer is independent of the objects's color and only depends on the light source position, color, and scene geometry.

Such a decomposition is powerful for image editing, because intrinsic layers can be manipulated separately and then recombined into a new image. However, the problem of intrinsic image decomposition is ill-posed, because at each pixel we try to recover the reflectance $\mathbf{R}$ and the illumination $\mathbf{S}$ using only the input image $\mathbf{I}$ (Equation 1.1). Existing approaches have incorporated different kinds of assumptions about the scene, in order to make the decomposition problem tractable. We review prior work in Section 2.3.2.

1.2.2 Our insight

In this thesis, we tackle the problem of *intrinsic image decomposition from multiple photographs*. We focus on the case where photographs captured from multiple views are available. A lot of information about the scene can be extracted from such a set of images; we exploit this additional information to add constraints which make the decomposition tractable and yield plausible results.

We build on recent advances in Computer Vision to automatically reconstruct partial scene geometry. We use off-the-shelf software to reconstruct a sparse 3D point cloud of the scene. We then leverage this geometric information to guide the intrinsic image decomposition process. The geometry reconstruction pipeline used in this thesis is briefly described in Section 2.4.

Automatic 3D reconstruction methods yield geometry which is often incomplete or inaccurate. A significant challenge consists in designing algorithms which identify and exploit the reliable parts of the reconstructed geometry, and which are robust to incomplete reconstructions. In our work, we select a subset of reliable 3D points and infer constraints on the corresponding pixels in the images to decompose. We then build on image-guided propagation algorithms to separate reflectance and illumination in regions where no geometric information is available.

The methods described in this manuscript apply to two types of input:

- Chapters 3 and 4 focus on outdoor scenes, using a few photographs with fixed lighting. Because the input photographs are taken at a *single time of day*, capture can be done quickly and with simple equipment.

- Chapter 5 targets sets of images with *varying illumination*, such as collections downloaded from photo-sharing websites, or sequences acquired indoors with a moving light source. This method can leverage the information contained in existing photographs in photo collections to facilitate the decomposition of new images with different lighting, requiring no additional capture.

1.2.3 Industrial applications

Intrinsic images are a representation which allows independent editing of material color and lighting in a photograph. We demonstrate in Chapters 3 and 5 advanced manipulations in image editing software, through the use of layers.

Eliminating the effect of illumination in images is the first indispensable step towards obtaining illumination-free models, which could be relit or inserted in different environments. Our work can simplify artists's tasks such as texturing and compositing: we show in Chapter 5 that lighting-free 3D models can be recovered from just a few photographs

from different viewpoints. This opens the way for applications in capture and rendering, enabling easy acquisition of real world objects rather than complex manual modification, and in architecture modeling, for example in cultural heritage. The work described in Chapter 3 led to a technology transfer agreement with Autodesk, which resulted in the development of the solution presented in Chapter 4.

Our intrinsic image decomposition from multiple views also has applications in image-based navigation and virtual tourism. Extracting the illumination from a photograph, and transferring it to different views, enables modification of lighting in images and is an important part of adapting the mood of a scene. In the European project VERVE which supported part of this research work, lighting manipulation will be used to create personalized and realistic virtual reality environments, in order to support the treatment of people who are at risk of social exclusion due to fear and apathy associated with ageing or a neurological disorder. In Chapter 5, we also demonstrate an application for virtual tourism, using lighting transfer to enable illumination-consistent view transitions.

1.3 Contributions

We present three approaches which exploit photographs from multiple views to extract information about the scene.

1.3.1 Rich decomposition of outdoor scenes

In Chapter 3, we present an approach to decompose a photograph of an outdoor scene. This method not only separates reflectance from illumination, but also introduces a decomposition of the illumination into sun, sky and indirect layers.

We use additional images captured from multiple views at a single time of day to automatically reconstruct a 3D point cloud of the scene. Although this point cloud is sparse and incomplete, it is sufficient to compute plausible sky and indirect illumination at each oriented 3D point, given a captured environment map that represents incoming distant radiance. We introduce an optimization method to estimate sun visibility over the point cloud, which compensates for the lack of accurate geometry and allows the extraction of precise cast shadows. We finally use image-guided algorithms to propagate the illumination computed over the sparse point cloud to every pixel, and to separate the illumination into distinct sun, sky, and indirect components.

This *rich intrinsic image decomposition* enables advanced manipulations which we demonstrate in image editing software, such as reflectance editing with coherent lighting, insertion of synthetic objects, and relighting.

1.3.2 Outdoor lighting extraction from a few photographs

The approach described in Chapter 3 requires user interaction during capture and calibration. In Chapter 4, we simplify capture by automatically identifying the direction of the sun, by estimating an environment map representing the incoming distant radiance, and by designing a simpler calibration process.

As a result, we estimate lighting incident to an outdoor scene from just a few photographs and minimal user interaction. First, we automate the estimation of sun direction by combining cues from the reconstructed geometry and captured photographs. Then, we automatically reconstruct an approximate environment map by extrapolating the portions of sky visible in the input photographs. Finally, we design a method to estimate the sun radiance from simple user indications (two clicks) instead of a grey card.

By simplifying the capture and calibration steps, we remove the most constraining aspects of our decomposition method and make it accessible to casual photographers. The work presented in Chapter 4 has been developed as part of a technology transfer agreement with Autodesk.

1.3.3 Coherent decomposition of photo collections

In Chapter 5, we focus on image collections with multiple viewpoints and multiple lighting conditions. Such collections can be gathered from photo-sharing websites, or captured indoors with a light source which is moved around the scene. We exploit the variations of lighting to process complex scenes without user assistance, nor precise or complete geometry.

We automatically reconstruct a set of 3D points and normals, from which we derive relationships between reflectance values at different locations, across multiple views and consequently different lighting conditions. We use robust estimation to reliably identify reflectance ratios between pairs of points. From these, we infer constraints for our optimization and enforce coherent reflectance in all views of a scene.

This constrained optimization yields *coherent intrinsic image decompositions* for multiple views of complex scenes. The resulting decompositions enable image-based illumination transfer between photographs of the collection, and view transitions with consistent illumination for image-based rendering applications.

CHAPTER 2

Background

2.1 Image formation and sensing

The color values of an image depend on the complex interactions of light with the scene geometry, environment, and materials, and on the properties of the capture system. Light energy is emitted by sources such as the sun; it then propagates through the environment, bounces off surfaces of objects with diverse geometry and material properties, and ultimately reaches camera sensors or human retinas which can record and process the signal. We briefly review here notions that will be useful in the remainder of this thesis.

The distribution of light in a scene is completely characterized by a quantity $L(\mathbf{p}, \vec{\omega})$, named *radiance*, which intuitively represents the quantity of light at position $\mathbf{p}$ travelling along direction $\vec{\omega}$. A more formal definition of radiance can be found in [Horn 1986].

An important aspect of light transport concerns the reflection of light from a surface. Reflection is defined as the process by which light incident to a surface point $\mathbf{p}$ leaves that surface on the same side, and is described by the *reflectance equation* [Hanrahan 1993]:

$$L_r(\mathbf{p}, \vec{\omega_r}) = \int_{\Omega_i} f_r(\mathbf{p}, \vec{\omega_i}, \vec{\omega_r})\ L_i(\mathbf{p}, \vec{\omega_i})\ \cos\theta_i(\mathbf{p})\ d\omega_i \tag{2.1}$$

The reflected radiance L_r in a particular direction $\vec{\omega_r}$ depends on the radiance arriving at point $\mathbf{p}$ from all incoming directions $\vec{\omega_i}$. Each incident radiance L_i is weighted by the angle θ_i between the incident direction and the surface normal at point $\mathbf{p}$, and by a function f_r. This *bidirectional reflection distribution function* (or BRDF) models the behaviour of the scene materials and can vary spatially.

While equation 2.1 models the reflection of light on opaque surfaces, other effects such as transparency, subsurface scattering, absorption, or fluorescence also affect the final appearance of a scene. These effects will not be treated in detail here; in addition, dependency on wavelength has been ignored.

Image sensors measure the radiant power per unit area received on their surface. This sensor *irradiance* is proportional to the radiance originating from the surface points visible to the sensor. Finally, the sensor irradiance is mapped to the observed image intensity by the camera response function. In this thesis, we assume that all input images have been

linearized to compensate for the camera response function[1], for example using the method described in [Debevec 1997].

2.2 Inverse rendering

The propagation of light in an environment has been studied in the field of physically based rendering, in an attempt to produce *more realistic* synthetic images. Because models such as Equation 2.1 accurately describe the physical quantities that would be measured from a real scene, they can also be used in inverse problems. *Inverse rendering* corresponds to the problem of recovering characteristics of the scene from observed intensities in recorded photographs.

Inverse rendering methods aim to recover at least one unknown scene attribute, which can be geometry, materials, or surrounding lighting, assuming other attributes are known and photographs of the scene are available. Despite extensive prior work on inverse rendering, most of the existing approaches focus on small objects or indoor settings [Sato 1997, Marschner 1998, Yu 1999, Loscos 1999, Boivin 2001, Lensch 2003, Yu 2006]. We describe here the approaches which try to recover the reflectance of real-world, natural scenes.

Known geometry and lighting. Yu and Malik [Yu 1998] recover the reflectance properties of an outdoor architectural scene. They acquire about 100 photographs of the scene and its surroundings (sky and landscape) at four different times of day, and measure the sun radiance with neutral density filters. After measuring and modeling the scene illumination, they use hand-modeled geometry to estimate spatially-varying diffuse and piecewise-constant specular reflectance.

Similarly, Debevec et al. [Debevec 1998, Debevec 2004] describe a process for estimating spatially-varying surface reflectance of a complex scene observed under natural illumination conditions. They use a laser-scanned model of the scene's geometry, photographs of the scene surface under a variety of illumination conditions, and capture the corresponding incident illumination with a lighting measurement apparatus. They use an iterative inverse global illumination technique to compute surface reflectances for the scene which, when rendered under the recorded illumination conditions, best reproduce the scene's appearance in the photographs. They also model non-Lambertian surface reflectance by measuring BRDFs of representative surfaces in the scene.

Known geometry, unknown lighting. Troccoli and Allen [Troccoli 2008] use a laser scanner and multiple photographs, with different viewpoints and lighting conditions, to

[1] We also assume that the linear images are scaled so that the pixel intensities correspond to the radiance incident to the camera. In our context of intrinsic image decomposition, there is a global scale ambiguity between the estimated reflectance and illumination layers.

estimate Lambertian reflectance of outdoor scenes. This approach does not require any light measurement device, but relies on a user-assisted shadow detector. Based on the estimated shadow map and known normals in regions where photographs overlap, it uses the ratio of two images to factor out the diffuse reflectance from the illumination.

Haber et al. [Haber 2009] propose an approach to recover the reflectance of a static scene from a collection of images with varying and unknown illumination. They simultaneously estimate the per-image distant illumination and the per-point BRDF, using an inverse rendering framework which handles non-Lambertian reflectance but neglects interreflections. The scene geometry can be reconstructed from images downloaded from the internet, using multi-view stereo. However, manual intervention remains necessary to correct spurious or inaccurate geometry, and this method assumes that the complete relevant scene geometry is reconstructed. This includes occluders which cast shadows on the objects, even though they might be visible in few pictures.

Discussion. These inverse rendering methods yield a textured, illumination-free 3D model of the scene and can estimate non-Lambertian BRDFs. This representation is convenient for applications such as free viewpoint navigation and dynamic relighting, which can generate renderings of the scene under novel lighting conditions. However, all these approaches assume the scene geometry is known and complete, and require manual intervention either during the capture (laser scanning, lighting acquisition) or processing (geometry modeling or cleaning) steps.

In contrast, we are interested in designing methods which are robust to incomplete geometry and handle sparse point clouds automatically reconstructed from a few photographs of the scene. In addition, we focus on image-based applications and aim to produce pixel-accurate decompositions despite possible misalignments in the reconstructed geometry.

2.3 Intrinsic image decomposition

Intrinsic images are a convenient representation which is more informative than just the original image, but also less complex than the full scene reconstructed by inverse rendering algorithms. Decomposing a photograph into a reflectance image and an illumination image (Equation 1.1) yields a compact representation which is well-suited for image-based applications.

We first study the relation between intrinsic images and the image formation model (Section 2.3.1). We then review existing work on intrinsic image decomposition (Section 2.3.2), and discuss their evaluation in Section 2.3.3. We present applications enabled by intrinsic images in Section 2.3.4.

2.3.1 Relation with the reflectance equation

We show that the decomposition into *reflectance* and *illumination* images is related to the image formation model described in Section 2.1. In particular, we can identify terms of Equation 1.1 in the reflectance equation, under a few assumptions that we will specify.

Assuming the scene reflectance is Lambertian, the light is equally likely to be scattered in any direction, regardless of the incident direction. In such a case, the BRDF $f_r(\mathbf{p}, \vec{\omega_i}, \vec{\omega_r})$ does not depend on the incoming and outgoing light directions, and we relate it to the reflectance $R(\mathbf{p})$ [Hanrahan 1993]:

$$f_r(\mathbf{p}, \vec{\omega_i}, \vec{\omega_r}) = f_r(\mathbf{p}) = \frac{R(\mathbf{p})}{\pi} \tag{2.2}$$

Equation 2.1 then becomes:

$$L_r(\mathbf{p}, \vec{\omega_r}) = \int_{\Omega_i} \frac{R(\mathbf{p})}{\pi} \, L_i(\mathbf{p}, \vec{\omega_i}) \, \cos\theta_i(\mathbf{p}) \, d\omega_i \tag{2.3}$$

$$= \frac{R(\mathbf{p})}{\pi} \int_{\Omega_i} L_i(\mathbf{p}, \vec{\omega_i}) \, \cos\theta_i(\mathbf{p}) \, d\omega_i \tag{2.4}$$

$$= \frac{R(\mathbf{p}) \, E(\mathbf{p})}{\pi} \tag{2.5}$$

where $E(\mathbf{p})$ represents the irradiance at point $\mathbf{p}$.

Assuming that the radiance towards the camera L_r is constant over the field of view of each sensor element[2], and that the image has been linearized, the image intensities are proportional to the radiance L_r. Relating Equations 1.1 and 2.5 in the three RGB channels then shows that for Lambertian scenes, the commonly named *illumination image* $\mathbf{S}$ is proportional to the irradiance at each visible scene point. In the rest of this thesis, we drop the π factor since the input images can be arbitrarily scaled and there is a global scale ambiguity between the reflectance and illumination images.

For non-diffuse reflectances, however, lighting and reflectance are coupled because the BRDF f_r depends on the incoming and outgoing directions of light. The intrinsic decomposition of Equation 1.1 does not represent such cases well, and more complex models are required.

2.3.2 Prior work

Estimating an intrinsic image decomposition is a severely ill-posed problem. The measured image colors encode the effects of both reflectance and illumination: at each pixel of an RGB image, $\mathbf{R}$ and $\mathbf{S}$ give 6 unknowns while $\mathbf{I}$ provides only 3 measured values. As a result, the decomposition is not unique and Equation 1.1 has an infinite number of

[2] The case where the radiance is not constant over a pixel is discussed in Section 5.4.1.1.

solutions. However, most of the mathematically valid solutions yield images which do not represent the reflectance and illumination components we look for. For example, this is the case with the trivial solution $\mathbf{S} = \mathbf{I}$ and $\mathbf{R} = 1$.

Successful methods make assumptions about the scene or use additional information, such as multiple images or user intervention, in order to constrain the decomposition to plausible solutions. We classify existing approaches based on the input they require.

2.3.2.1 Single-image methods

Analysis of local variations. Earlier methods have focused on classifying edges in the input image as illumination or reflectance edges, according to various assumptions.

Horn extends the Retinex theory [Land 1971] in order to estimate the reflectance of a particular class of scenes, the *Mondrians*, which consist of flat patches of uniform matte color under uneven illumination. In images of such scenes, the reflectance is constant within each patch and has sharp discontinuities at the boundaries between patches, whereas the illumination varies smoothly over the image. Horn thresholds small derivatives in the original image to estimate derivatives of the reflectance image; the reflectance image is then obtained by re-integrating the modified derivatives [Horn 1974]. This assumes that small image variations correspond to illumination changes, which is valid in the world of Mondrians. However, non-uniform reflectance and sharp illumination changes, due to corners or shadows, can make this method fail on real-world scenes. Funt et al. [Funt 1992] extend this approach to color images. They instead identify reflectance changes as large variations in chromaticity, and assume monochromatic illumination. Different formulations of the Retinex problem for intrinsic images are reviewed and unified in [Kimmel 2003].

Sinha and Adelson [Sinha 1993] discriminate edges based on the type of their junctions, then verify the global consistency of these local inferences. They consider the domain of painted polyhedral/origami objects in the absence of occlusions and cast shadows. Hsieh et al. [Hsieh 2009] transform the input image into a color domain where most significant illumination changes appear in a single channel. They then create a weighted map where the reflectance derivatives are in general larger than the illumination derivatives, and discriminate edges by applying a threshold on this map.

[Bell 2001] and [Tappen 2005] are learning-based approaches which predict the derivatives of reflectance and illumination images. Their authors generate synthetic images showing examples of reflectance and illumination changes, and train classifiers to interpret local variations. In [Tappen 2005], local estimation is then propagated using belief propagation in order to disambiguate locally ambiguous regions. More recently, [Tappen 2006] estimates the illumination image with low-dimensional local estimators based on small image patches. These estimators are learned from training data, which consists of captured real-world images with associated ground truth. Instead of classifying image derivatives as *either* reflectance *or* illumination changes, Tappen et al. reconstruct the final image by

weighting the different local estimates based on their reliability.

Jiang et al. [Jiang 2010] analyze the correlations between local mean luminance and local luminance amplitude to interpret luminance variations in the input image. They separate the image into frequency and orientation components using steerable filters, and reconstruct illumination and reflectance images from weighted combinations of these components.

Although these approaches discriminate reflectance and illumination changes based on diverse classifiers and heuristics, many configurations of reflectance and illumination commonly encountered in natural images remain hard to classify.

Global constraints. More recent approaches incorporate non-local constraints or global cues to improve the decompositions.

Shen et al. [Shen 2008] improve the Retinex approach by combining it with non-local texture constraints. They identify distant pixels with the same texture configuration by matching chromaticity in neighborhoods, and force such pixels to share the same reflectance value. Incorporating such non-local constraints weakens dependencies on the original Retinex assumptions, such as illumination smoothness. Zhao et al. [Zhao 2012] propose an optimization formulation which encompasses the Retinex constraints and the non-local texture constraints, and which has a closed-form solution.

Garces et al. [Garces 2012] detect clusters of similar chromaticities in the input image and assume they share the same reflectance. They relax the Retinex assumption of smooth illumination, and instead assume that the illumination at cluster boundaries is continuous. They formulate the decomposition as a linear system which describes the connections between *clusters*, rather than pixels, resulting in an a fast decomposition.

Shen and Yeo [Shen 2011b] exploit a global prior on the reflectance. They assume that the set of reflectances is sparse, i.e., that the scene contains a limited number of different material colors. In addition, they relax the Retinex assumptions and instead assume that neighboring pixels with similar chromaticities share the same reflectance. Gehler et al. [Gehler 2011] enforce a similar global sparsity term on the reflectance, but formulate the decomposition as a probabilistic problem where reflectance values are drawn from a sparse set of basis colors.

Barron and Malik [Barron 2012] focus on the related problem of "shape, albedo, and illumination from shading". From a grayscale image of a single object, they aim to recover its shape, reflectance, and distant incident lighting as a spherical harmonic model; intrinsic images for reflectance and illumination can be deduced once these three components have been estimated. In order to solve this ill-posed problem, they also use a combination of local priors, such as reflectance and orientation smoothness, and global priors, such as reflectance sparsity.

Discussion. Estimating intrinsic images from a single RGB image is an under-constrained problem, which requires all methods to make assumptions about the scene to obtain plausible decompositions. Such assumptions limit the applicability to particular scenes where they are valid. In particular, most of the methods described here assume monochromatic illumination, which reduces the number of unknowns and simplifies the problem. However, this assumption does not hold in the case of real outdoor scenes.

In the thesis, we adopt a more physically-based approach and start from the reflectance equation described in Section 2.3.1. In Chapter 3 we constrain values of the illumination image using the scene irradiance estimated with coarse scene geometry, while in Chapter 5 we derive non-local constraints between pairs of distant points which are consistently illuminated.

Shadow removal. Intrinsic image decomposition is also related to shadow detection and removal methods [Mohan 2007, Wu 2007b, Shor 2008, Arbel 2011, Guo 2011, Sanin 2012], which aim to remove cast shadows in an image, either automatically or with user assistance. Finlayson et al. [Finlayson 2002, Finlayson 2004] also recover a shadow-free image, but it does not represent the reflectance image as defined in Equation 1.1. While intrinsic image decomposition aims to extract a reflectance image as well as illumination variations, it also allows the subsequent removal of shadows by editing the illumination layer. Note that in Chapter 3, we explicitly estimate the sun visibility (i.e., cast shadows) at sparse reconstructed points of the scene.

2.3.2.2 User-assisted methods

Instead of making strong assumptions about the scene, which are necessary to constrain the problem of decomposing a single image, some approaches rely on user assistance to disambiguate reflectance and illumination.

Bousseau et al. [Bousseau 2009] propose a method which enables users to guide the decomposition with a sparse set of simple annotations. These *user scribbles* indicate regions of constant reflectance, constant illumination, or known absolute illumination (Figure 2.1). Inspired by [Levin 2008], Bousseau et al. propagate the user-specified constraints to all pixels using an image-guided energy formulation, which assumes that local reflectance variations lie in a plane in color space. Combining user scribbles with their propagation energy enables user-assisted decomposition of complex images, including on scenes which receive colored illumination.

Shen et al. [Shen 2011a] use similar user scribbles but a different propagation energy: they express the reflectance at each pixel as a weighted combination of the reflectance of its neighbors. They define *affinity weights* between pairs of pixels according to the assumption that neighboring pixels which share similar intensity and chromaticity values should have similar reflectances.

(a) Input image (b) User scribbles (c) Reflectance image (d) Illumination image

Figure 2.1: *User-assisted intrinsic decomposition of [Bousseau 2009]. Starting from a single input image (a), users mark scribbles indicate regions of similar reflectance, similar illumination, or known absolute illumination (b). Constraints inferred from the scribbles are then propagated to all pixels and guide the decomposition into reflectance (c) and illumination (d) images.*

Dong et al. [Dong 2011] propose an interactive system for modeling materials from a single texture image. In particular, they separate illumination from reflectance and identify different materials in the image. They assume that nearby pixels with similar chroma values correspond to the same material and have the same reflectance, and that the large scale geometry is almost flat. They then interactively correct the estimation with user scribbles in regions that violate their assumptions.

Okabe et al. [Okabe 2006] propose a pen-based interface to specify approximate normals, and propagate them over the image. Their algorithm then simultaneously refines these normals and estimates the reflectance at each pixel. It assumes that the scene is mostly illuminated with directional lighting, and shadows are not dominant in the image. The extraction of reflectance and normals then allows the photograph to be relit under different illumination conditions.

In work developed concurrently, Karsch et al. [Karsch 2011] describe a system to realistically render synthetic objects in an existing photograph. They exploit user annotation to recover a simple geometric model of the scene, and the position, shape, and intensity of light sources. They then iteratively refine the lighting model and estimate the scene reflectance, with a Retinex-inspired intrinsic image decomposition method which exploits the geometry estimation. Although they start from a single image, the user annotation allows them to extract a model of the scene that is suitable for inserting virtual objects.

Discussion. The methods we propose in Chapters 3 and 5 share similarities with the scribble-based approaches: we also define constraints at a few pixels in the input images, and use image-guided propagation to disambiguate other regions. However, we avoid the need for user annotations, and instead infer constraints at a sparse set of points using reconstructed geometry. In contrast with the user-assisted model estimation described

in [Karsch 2011], we automatically reconstruct approximate scene geometry using multiple views.

In Chapter 3, we ask users to capture a few additional pictures and perform simple calibration steps, once for each scene, instead of providing scribbles on each input image. After these steps, we automatically compute illumination constraints and extract multiple illumination components (sun, sky, indirect), which would be hard to disambiguate by users. We further simplify the capture and calibration steps in Chapter 4. In contrast, the method we describe in Chapter 5 does not require user assistance, and handles a large number of images from a photo collection automatically.

2.3.2.3 Multiple-images methods

Several methods use images captured from a single viewpoint under multiple lighting conditions (i.e., *timelapse* sequences) to constrain the decomposition. A timelapse sequence of N frames can be factored into N illumination images and a *single* reflectance image, assuming the scene is static.

Weiss [Weiss 2001] exploits the statistics of natural images to decompose a timelapse sequence. He formulates the problem as a maximum-likelihood estimation, based on the assumption that derivative filters applied to natural images tend to yield sparse outputs. He shows that the reflectance derivatives can be robustly estimated by applying a median operator on the N observations of the image gradients (in the log domain). The reflectance image can then be re-integrated from its derivatives. However, Matsushita et al. [Matsushita 2004a] observe that shading residuals can appear in the reflectance image when neighboring pixels have different normals and the input images do not cover the illumination directions uniformly. They instead use the median estimator to detect flat surfaces, on which they explicitly enforce smooth illumination. Matsushita et al. [Matsushita 2004b] extend Weiss's method to handle non-Lambertian scenes. They derive time-varying reflectance images instead of extracting a single reflectance image.

Sunkavalli et al. [Sunkavalli 2007] decompose timelapse sequences of cloudless outdoor scenes into a shadow mask and images illuminated only by the sky or by the sun. This separation allows them to factorize and compress timelapse sequences. In subsequent work, Sunkavalli et al. [Sunkavalli 2008] model distinct time-varying colors of ambient daylight and direct sunlight, which allows them to extract a reflectance image, and illumination images corresponding to sunlight and skylight; they also recover scene information such as sun direction, camera position, and partial geometry.

Matusik et al. [Matusik 2004] additionally measure the radiance incident to an outdoor scene for each frame of a timelapse sequence. They use two cameras and a chrome sphere to capture high-dynamic range images of the scene and the sky every two minutes over a period of three days. They then estimate the reflectance field, i.e., a description of the transport of light through the scene. Although the reflectance field can be used for relight-

ing, this method does not explicitly estimate a reflectance and an illumination image, and instead treats the scene as a black-box linear system that transforms an input signal (the incident radiance) into an output signal (the reflected radiance towards camera).

These methods assume a fixed viewpoint and varying illumination. For outdoor scenes, this leads to a lengthy and inconvenient capture process since lighting due to the main illuminant (the sun) evolves slowly. In contrast, Liu et al. [Liu 2008] retrieve images from different (yet similar) viewpoints and varying lighting, and use them to colorize an input grayscale image. They extend Weiss's approach [Weiss 2001] to recover reflectance and illumination of the scene as viewed from the viewpoint of the grayscale image. While the output of this decomposition is sufficient to transfer color to the input grayscale picture, it produces blurry reflectance images and focuses on photographs from similar viewpoints.

Discussion. Our approaches build on this family of work, but we seek to take advantage of the partial 3D information provided by multiple views and avoid the sometimes cumbersome timelapse capture process. We use multiple views to reconstruct sparse scene geometry, which allows us to constrain the decomposition based on the image formation model of Section 2.1.

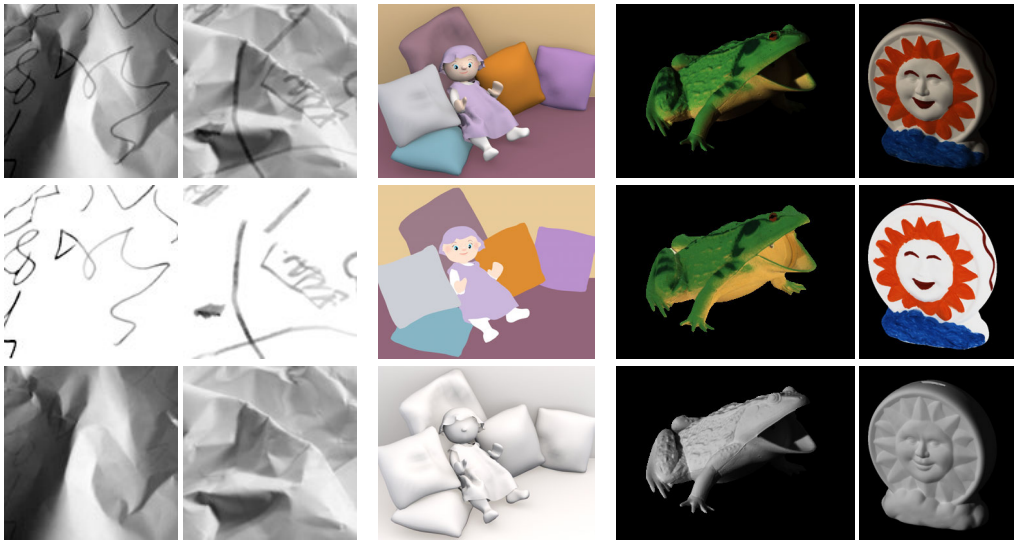
In Chapter 3, we describe a method which uses a few images captured *at a single time of day*. The main advantage of this approach is to reduce the acquisition time, while it also allows us to separate the illumination due to sun, sky, and indirect lighting. However, it requires a chrome sphere to capture incident radiance, and manual calibration steps; we simplify this process in Chapter 4.

In contrast, the method we describe in Chapter 5 exploits lighting variations to derive constraints between pairs of pixels. It therefore applies to unstructured photo collections, where the lighting is varying and unconstrained, or to indoor scenes, in which a light source can be moved around to vary the lighting conditions.

2.3.3 Evaluation

Evaluating the results of intrinsic image decomposition methods is a non-trivial task. Residual shadows or shading artifacts in the reflectance layers are clearly visible, but it is more difficult to evaluate the decomposition for different surfaces with often varying orientations and material colors. As a result, a visual comparison is not sufficient to compare the output of different approaches.

An additional difficulty stems from the fact that there is no *ground truth* available to validate the results on arbitrary images, because creating such ground truth would require measuring the geometry and material properties everywhere in the scene. A few efforts have been made towards providing datasets associated with ground truth, for training and/or evaluation. However, existing datasets consist of single objects, or scenes rendered with non-photorealistic lighting conditions (Figure 2.2).



(a) [Tappen 2005] (b) [Bousseau 2009] (c) [Grosse 2009]

Figure 2.2: *Existing datasets for intrinsic decomposition with ground truth. Top row shows the input image, second row the ground truth reflectance layer, bottom row the ground truth illumination layer. (a) Photographs of crumpled papers scribbled with a green marker. (b) Synthetic rendering with simple illumination. (c) Photographs of isolated objects. In Chapter 5, we create a synthetic dataset made of realistic renderings of a complex textured scene, with a physically-based sky and global illumination.*

Tappen et al. [Tappen 2006] collect a set of images of crumpled paper which they color with a green marker. As the green channel of the captured images does not contain any of these markings, a ground-truth grayscale albedo image can be extracted by dividing the red and green channels of the photograph.

Grosse et al. [Grosse 2009] present the *MIT Intrinsic Images dataset*, which includes ground truth intrinsic image decompositions for 16 real objects, in three categories: artificially painted surfaces, printed objects, and toy animals. They also capture images with a fixed viewpoint and ten different positions for a handhelp lamp, to use with Weiss's approach [Weiss 2001]. The decompositions are obtained using polarization techniques to remove specular highlights, and various paints to recover diffuse images of the same object with and without reflectance variation; however, they do not account for interreflections. Grosse et al. use this dataset to quantitatively compare several existing algorithms, with an error criterion that they define as the Local Mean Squared Error (LMSE).

The MIT dataset is very useful for evaluating intrinsic decomposition methods. However, it consists of isolated objects illuminated with a single direct light source; therefore, it targets a simpler version of the intrinsic image problem, because real scenes made of several objects exhibit complex (and possibly colored) illumination, interreflections, occlusion boundaries at the objects's outlines, or colored cast shadows. Bousseau et al. [Bousseau 2009] also present a synthetic image with ground truth; however, it consists of a simple scene composed of objects with uniform reflectance, and does not contain noticeable interreflections or cast shadows.

Several approaches presented in Section 2.3.2 obtain good scores on the MIT benchmark for isolated objects, but it is unclear how well they work on real scenes and outdoors environments. In particular, most existing methods assume monochrome lighting while outdoor scenes are often lit by a mixture of colored sun and sky light.

In Chapter 5, we propose a synthetic dataset which contains physically-based renderings of an outdoor scene with complex geometry and reflectance, under varying viewpoints and lighting conditions. This allows a more meaningful comparison to ground truth since our dataset captures indirect lighting, shadows, and occlusions between parts of the captured scene. We quantitatively compare the results of several existing methods using this dataset. Our evaluation in Chapters 3 and 5 also demonstrates that our methods are more robust to common outdoor lighting scenarios such as mid-day shadows, sunset, or urban night lights, because we do not make the assumption of monochromatic lighting shared by many existing methods.

2.3.4 Applications

Intrinsic images enable a variety of applications in image editing. In particular, Bousseau et al. [Bousseau 2009] modify the reflectance layer to alter textures while preserving coherent illumination, and Beigpour et al. [Beigpour 2011] similarly re-color objects. Yan et al. [Yan 2010] focus on re-texturing objects in videos. The separation of reflectance and illumination facilitates the re-texturing, whereas previous work such as TextureShop [Fang 2004] used the luminance channel to approximate the illumination layer; this approximation is not valid for images with varying reflectance or colored lighting. Given an intrinsic image decomposition, Carroll et al. [Carroll 2011] propose a user-assisted decomposition to isolate the indirect contribution of each colored material in the scene; this enables the manipulation of object colors with consistent interreflections. Liu et al. [Liu 2008] use intrinsic image decompositions and color transfer to colorize grayscale images.

The illumination layer can be manually edited to enable image-based relighting, as the day-to-night example shown in [Bousseau 2009]. Luo et al. [Luo 2012] infer normals from the illumination image and estimate the subtle 3D relief of oil paintings, which can then be re-rendered under different lighting conditions. The system of Melendez et al. [Melendez 2011] reconstructs 3D models of historic buildings, and transfers the reflectance of material exemplars to the model texture. The textured 3D model can then be rendered under novel lighting conditions. Karsch et al. [Karsch 2011] leverage intrinsic images to estimate a lighting model and reflectance from a single image and user annotations, thus enabling the insertion of virtual objects in existing photographs.

In Chapter 3, we propose a rich intrinsic decomposition which separates reflectance from illumination and further decomposes the illumination into sun, sky, and indirect components. We show that modifying each layer independently in image editing software

allows advanced manipulations such as lighting-aware editing, insertion of virtual objects, or relighting. In Chapter 5, we demonstrate how our multi-view decomposition facilitates texturing illumination-free 3D models. Lastly, we develop a method for image-based illumination transfer, which enables the transfer the lighting from an image to a different viewpoint in a photo collection; this allows for illumination-consistent view transitions between photographs of the collection.

2.4 Geometry reconstruction

In this thesis, we leverage the information in multiple photographs to extract geometric information about the scene and guide the intrinsic image decomposition.

Several methods have been developed in order to automatically extract 3D geometry using photographs captured from multiple viewpoints. In this section, we briefly describe the reconstruction pipeline used in our work, which is based on off-the-shelf software for automatic 3D reconstruction. We refer the reader to [Snavely 2010] for more detail.

Starting from unorganized images of a scene, captured from different viewpoints, possibly at different times and with different cameras, the reconstruction aims to recover two elements:

- a position and orientation for each input photograph (the *camera pose*), describing where it was taken and the parameters of the corresponding camera;
- a set of 3D points corresponding to physical points in the scene (the reconstructed *point cloud*), and a list which indicates, for each point, the images in which it is visible.

Reconstructing cameras. The first step consists in recovering the camera pose and intrinsic parameters for each input photograph, in order to relate all photographs in a single 3D coordinate system. To do so, distinctive local features are extracted from the input pictures, then matched across images in order to identify similar-looking features in different views [Lowe 2004, Wu 2007a]. Pixel correspondences may correspond to the same physical points observed from different angles. Given enough matches, they can be used to recover the 3D camera poses and the 3D position of each point in the set of matches; this process is known as Structure from Motion. We use publicly available software for recovering camera poses, namely Bundler [Snavely 2006] and VisualSFM [Wu 2011].

Reconstructing 3D geometry. Once the camera parameters have been estimated, Multi-View Stereo algorithms [Seitz 2006] can recover the 3D structure of the scene. We use a patch-based approach (namely PMVS [Furukawa 2009b]) which extracts a set of points on the scene surface, where both the 3D position and 3D normal are estimated, and a list of images in which each point is visible. However, the normals estimated by PMVS are often

noisy; we recalculate the normal to each point by fitting a local plane on the 3D position of neighboring points [Hoppe 1992].

This automatic reconstruction pipeline results in a point cloud which represents the scene geometry. This point cloud is often very irregularly sampled: textured regions are densely reconstructed thanks to the presence of numerous image features, while uniform textureless regions and specular objects contain few reconstructed points. As a result, constructing a mesh from this point cloud, for example with Poisson reconstruction [Kazhdan 2006], results in incomplete and inaccurate geometry. In this thesis, we design methods which can handle such geometry resulting from automatic 3D reconstruction. We identify reliable parts of the reconstructed point cloud, and guide the intrinsic image decomposition problem with this knowledge of partial geometry.

CHAPTER 3

Rich Intrinsic Image Decomposition of Outdoor Scenes

In this chapter[1], we focus on outdoor scenes and use multiple photographs, captured at a *single time of day* from different viewpoints, to guide the decomposition. We introduce a *rich intrinsic image decomposition* which extracts reflectance and illumination layers from an input image, and also separates the illumination into components due to sun, sky, and indirect lighting.

Our algorithm takes as input a small number of photographs of the scene, an environment map which represents the illumination coming from the sky and distant environment in all directions, and two pictures of a photographer's grey card for calibration. From this lightweight capture we use recent computer vision algorithms to reconstruct a sparse 3D point cloud of the scene. Although the point cloud only provides an imprecise and incomplete representation of the scene, we show that this is sufficient to compute plausible sky and indirect illumination at each reconstructed 3D point. The coarse geometry is however unreliable for sun illumination, which typically contains high-frequency features such as cast shadows. We introduce a new parameterization of reflectance with respect to sun visibility that we integrate in an optimization algorithm to robustly identify the 3D points that are in shadow. We developed an optimization inspired by mean shift [Comaniciu 2002] where we use asymmetric regions of influence and constrain the evolution of the estimates.

Image-guided propagation algorithms are typically used to propagate user scribbles [Levin 2004, Bousseau 2009]; we show how to use these algorithms to propagate the illumination information computed at 3D points to all the image pixels. Our approach generates intrinsic images of similar quality as scribble-based approaches, with only a small amount of user intervention for capture and calibration. In addition, our ability to separate sun, sky and indirect illumination (Figure 3.1e-h) opens the door for advanced image manipulations, as demonstrated in Figure 3.1b-d.

[1] The work described in this chapter will be published in IEEE Transactions on Visualization and Computer Graphics [Laffont 2012a]. An early version also appeared in [Laffont 2011].

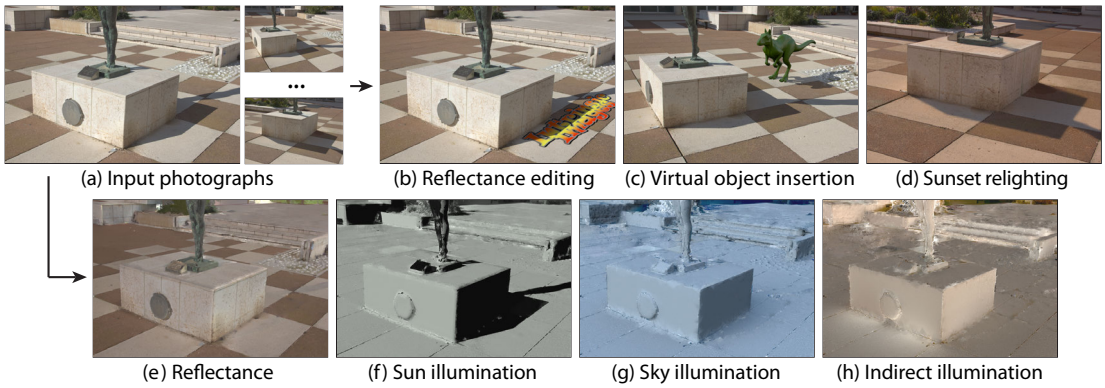


Figure 3.1: *Starting from multiple views of the scene (a), our method decomposes photographs into four intrinsic components — reflectance (e), illumination due to sun (f), illumination due to sky (g) and indirect illumination (h). Each intrinsic component can then be manipulated independently for advanced image editing applications (b-d).*

In summary, this chapter makes the following contributions:

- We show how to compute sky, indirect, and sun (ignoring cast shadows) illumination at automatically reconstructed 3D points, using incomplete and imprecise geometry and a small set of input images.
- We introduce an algorithm to reliably identify points in shadow based on a new parameterization of the reflectance with respect to sun visibility. Our algorithm compensates for the lack of accurately reconstructed and complete 3D information.
- We show how to propagate reflectance, sun, sky and indirect illumination to all pixels in an image, without user intervention or involved inverse global illumination computation. We achieve this by using the illumination values computed at 3D points as constraints for image propagation algorithms.

After the definition of our image formation model and a description of the capture process, the structure of this chapter follows these three contributions.

3.1 Overview

Image formation model. We assume Lambertian surfaces and model the image values at each pixel as the product between the incident illumination and the object reflectance $\mathbf{R}$. Formally, the radiance $\mathbf{I}$ towards the camera at each non-emissive, visible point corresponding to a pixel is given by the equation:

$$\mathbf{I} = \mathbf{R} * \int_{\Omega} \cos\theta_{\omega} \mathbf{L}(\omega) d\omega \tag{3.1}$$

where we integrate over the hemisphere Ω centered on the normal at the visible point, $\mathbf{L}(\omega)$ is the incoming radiance in direction ω, θ_ω is the angle between the normal at the visible point and direction ω. Capital bold letters represent RGB color values and $*$ denotes per-channel multiplication.

For our purposes, we will separate out the incoming radiance into three components: the radiance due to the sun, that due to the sky and that due to indirect lighting. To simplify notation, we define two subsets of the hemisphere: Ω_{sky}, i.e., the subset of directions in which the visible point sees the sky, and Ω_{ind} the subset of directions in which another object is visible, and thus contributes to indirect lighting. We however explicitly represent the sun visibility v_{sun}, first because precise computation of v_{sun} is necessary to capture sharp shadows, and second because estimating v_{sun} robustly is one of our main contributions.

We can now re-write Equation 3.1:

$$\mathbf{I} = \mathbf{R} * \Big(v_{\text{sun}} \max(0, \cos\theta_{\text{sun}}) \mathbf{L}_{\text{sun}} + \int_{\Omega_{\text{sky}}} \cos\theta_\omega \mathbf{L}_{\text{sky}}(\omega) d\omega + \int_{\Omega_{\text{ind}}} \cos\theta_\omega \mathbf{L}_{\text{ind}}(\omega)\, d\omega \Big)$$

where $\mathbf{L}_{\text{sun}}$, $\mathbf{L}_{\text{sky}}$ and $\mathbf{L}_{\text{ind}}$ are radiance from the sun, the sky and indirect lighting respectively, θ_{sun} is the angle between the normal at the visible point and the sun modeled as a directional light source, and θ_ω is the angle between the normal and the direction of integration ω over the hemisphere. The scalar $v_{\text{sun}} \in [0, 1]$ models the visibility of the sun (0 for completely hidden, 1 for fully visible).

We next define simplified quantities at each pixel:

$$\mathbf{S}_{\text{sun}} = v_{\text{sun}} \max(0, \cos\theta_{\text{sun}}) \mathbf{L}_{\text{sun}} = v_{\text{sun}} \hat{\mathbf{S}}_{\text{sun}} \tag{3.2}$$

$$\mathbf{S}_{\text{sky}} = \int_{\Omega_{sky}} \cos\theta_\omega \mathbf{L}_{\text{sky}}(\omega)\, d\omega \tag{3.3}$$

$$\mathbf{S}_{\text{ind}} = \int_{\Omega_{ind}} \cos\theta_\omega \mathbf{L}_{\text{ind}}(\omega)\, d\omega. \tag{3.4}$$

where $\hat{\mathbf{S}}_{\text{sun}}$ corresponds to the sun illumination when cast shadows are ignored. We define a simplified image formation model from these quantities:

$$\mathbf{I} = \mathbf{R} * (\mathbf{S}_{\text{sun}} + \mathbf{S}_{\text{sky}} + \mathbf{S}_{\text{ind}}) \tag{3.5}$$

$$= \mathbf{R} * \mathbf{S}_{\text{total}} \tag{3.6}$$

where $\mathbf{R}$ is the object RGB reflectance. $\mathbf{S}_{\text{sun}}$, $\mathbf{S}_{\text{sky}}$ and $\mathbf{S}_{\text{ind}}$ are the RGB illumination (or irradiance) from the sun, sky and indirect lighting respectively.

Outline. Our first goal is to extract the reflectance $\mathbf{R}$ and illumination $\mathbf{S}_{\text{total}}$ from this image formation model. We demonstrate how to make this problem tractable by leveraging the sparse geometric information generated by multiview stereo algorithms to compute $\hat{\mathbf{S}}_{\text{sun}}$, $\mathbf{S}_{\text{sky}}$ and $\mathbf{S}_{\text{ind}}$ at each reconstructed 3D point (Section 3.3). We then introduce a new algorithm to estimate the sun visibility v_{sun} precisely despite the approximate available geometry (Section 3.4), which will allow us to obtain all illumination components using image-guided propagation (Section 3.5). Figure 3.2 illustrates the main steps of our approach in the form of a block diagram.

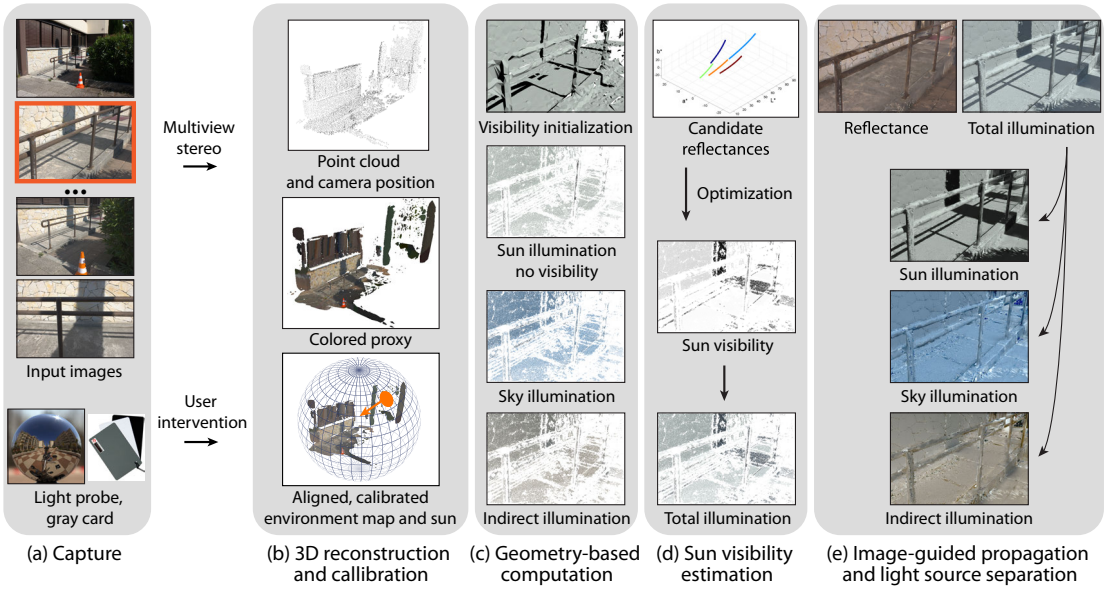


Figure 3.2: *Overview of our approach. Users capture a small set of pictures of the scene, along with an environment map and two pictures of a gray card in sun light and in shadow (a). We illustrate our intrinsic image decomposition with the picture highlighted in orange. We use multiview stereo to reconstruct a point cloud of the scene and a coarse proxy geometry (b). Users align the environment map and the sun with this point cloud and use the gray card to calibrate their intensity. Once this calibration is performed, all the remaining steps are automatic. We use the reconstructed 3D geometry to compute sun, sky and indirect lighting over the point cloud (c). We also compute an initial guess of the sun visibility using the coarse proxy. These lighting values give us the necessary information to compute a set of candidate reflectances for each 3D point. The candidate reflectances form curves in color space parametrized by the sun visibility (d). We introduce an iterative optimization that identifies the reflectance of each 3D point from these candidates, along with a precise estimation of the sun visibility. The final step of our method consists in propagating the illumination values computed at 3D points to every pixel in the image (e). We decompose the propagated illumination into the sun, sky and indirect lighting components.*

3.2 Capture and Reconstruction

Our method relies on a lightweight capture setup composed of a digital camera (preferably on a tripod), a photographer's gray card and a simple reflective sphere (Figure 3.3a). No other special capture or measurement hardware is required.

Even though we use a reflective sphere to capture an environment map, the apparatus we use is by far simpler than the ones required for similar inverse-rendering based approaches. In particular, Debevec et al. [Debevec 2004] measured the reflectance of a mirrored sphere, whereas Yu et al. [Yu 1998] used a couple of neutral density filters to measure the radiance of the sun. In contrast, we use an inexpensive *pétanque* ball (Figure 3.3b) to capture the environment map, and propose a simple calibration step described in Section 3.2.3. While existing methods rely on complex equipment to yield high precision results, we target casual photographers and propose a simpler capture which yields plausible decompositions.

(a) Capture equipment (b) Game of *pétanque* in Cannes

Figure 3.3: *(a) Equipment used in the capture process: a DSLR camera, a photographer's gray card, and a reflective sphere. (b) Pétanque is a popular ball game in southern France and is played with metal balls. We use such a ball as an inexpensive device to capture an environment map.*

3.2.1 Photography

We first capture a few ordinary low-dynamic range photographs (LDR) which we use to perform approximate geometric reconstruction of the scene. This set of photographs should have a good coverage of the scene from different viewpoints and sufficient overlap between neighboring viewpoints to facilitate multiview stereo reconstruction. The number of photographs required to obtain an acceptable reconstruction depends on the complexity of the scene and the presence of image features. We captured between 10 and 31 LDR photographs for the scenes in this chapter.

We then capture two linear, high-dynamic range (HDR), images of the front and side of the reflective sphere, placed in the scene, to obtain an angular environment map of the scene (Figure 3.4). We do this using the standard HDR assembly technique of Debevec et al. [Debevec 1997].

We finally capture linear HDR images of the viewpoints that we want to decompose. Recall that we capture all images in one session, *at a single time of day*.

3.2.2 Scene reconstruction

We apply structure-from-motion using Bundler [Snavely 2006] and the patch-based multi-view stereo (PMVS) algorithm [Furukawa 2009b] on the set of LDR+HDR photographs, using the publicly available implementations. The result of this process is an oriented point cloud (3D positions and normals), and calibrated cameras (extrinsic and intrinsic parameters). The process also returns whether each point is visible from each camera. We rectify the images for radial distortion, using the recovered camera parameters.

We found the PMVS normals to be too noisy for illumination computation. We instead estimate normals at each 3D point by fitting a plane on the local point cloud using PCA [Hoppe 1992]. We discard 3D points for which the neighborhood is too sparse or degenerate. We consider a neighborhood as degenerate when the first singular value of the PCA is twice greater than the second one.

We recover a geometric proxy from the oriented point cloud using the scale space meshing method of Digne et al. [Digne 2011]. This automatic approach produces detailed accurate meshes in regions where the point cloud is dense, while leaving holes in areas where the point cloud is irregularly sampled. We use the automatic hole filling tool available in MeshLab [Cignoni 2008] to further improve the reconstruction in those areas. We also experimented with the Poisson reconstruction of Kazhdan et al. [Kazhdan 2006]. Although this algorithm generates a closed surface with no holes, we found that it tends to produce bumpy surfaces due to the irregular sampling of the point cloud. We nevertheless used the Poisson reconstruction on the "Rocks" scene for which the point cloud is dense and uniform. Note that recent approaches could improve the quality of reconstructions in specific scenarios, for instance by detecting planar regions (e.g., [Sinha 2009, Furukawa 2009a, Gallup 2010]) in urban scenes.

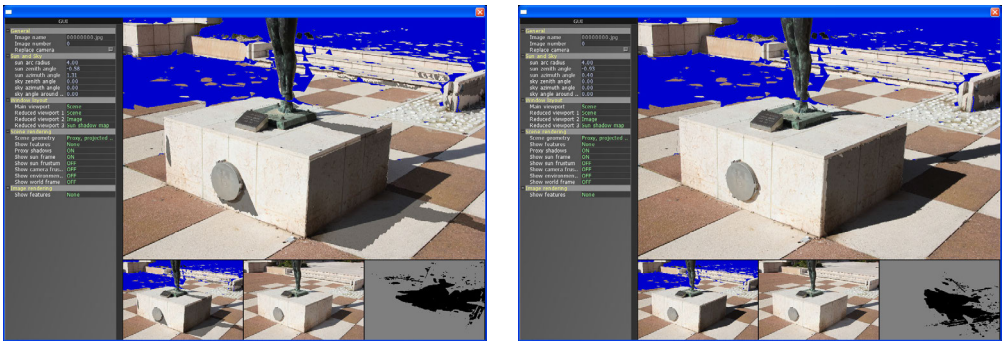
3.2.3 Illuminant Calibration

Some manual interaction is required to calibrate the sun and sky illumination in the system described in this chapter. Users specify the sun position, using cues from the shadows in the image and reconstructed geometry. They label sky pixels in the environment map, and rotate the sky dome until it aligns with the specified sun position and the scene horizon. Finally, they take two photographs of a gray card placed in the scene, in sunlight and in shadow, at the time of capture in order to estimate the color transfer function of the reflective sphere and the radiance $\mathbf{L}_{\text{sun}}$ of the sun. We now describe details of this process.

First, because our model separates sun light from sky light, we need to remove sun pixels from the environment map. We define the sun position as the barycenter of the saturated sun pixels, and use inpainting to fill-in these saturated pixels from their neighbors. Since our model also separates sky light from indirect light, we use a standard color selection tool to label sky pixels that will contribute to the sky illumination, while other pixels (buildings, trees) will contribute to indirect lighting. This is illustrated in Figure 3.4b.

(a) Input photographs of the sphere (b) Assembled environment map

Figure 3.4: *We use the standard HDR assembly technique of Debevec et al. [Debevec 1997] to merge two pictures of a reflective sphere (a) into an environment map (b), shown in a latitude-longitude parameterization. The red curve separates pixels contributing to sky illumination and those contributing to distant indirect illumination, and is specified by the user (see Section 3.2.3).*



(a) Bad orientation of the sun (b) Correct orientation of the sun

Figure 3.5: *The user specifies the orientation of the sun by adjusting the sun zenith and azimuth angles. In our calibration program shown here, a directional light source generates shadows using the proxy geometry. The user compares this virtual shadow with the shadow in the original photograph (a), and changes the sun position until both shadows overlap (b).*

(a) Location of the gray target (b) Shadowed gray target (c) Lit gray target

Figure 3.6: *A simple calibration step with a photographer's gray card is used to calibrate the illuminants. The gray card is placed in a planar region whose geometry is well reconstructed (a, red circle). We take two photographs of the gray card, once in shadow (b) and once in sunlight (c).*

Second, we align the environment map and sun with the reconstructed scene. To do so we manually mark a vertical edge of the reconstructed geometry and rotate the environment map and sun until the cast shadow of the virtual edge is aligned with that in the photograph. This process is illustrated on Figure 3.5.

Finally, the environment map only captures a scaled version of the incident lighting since the sphere is not perfectly specular. We need to compensate for this scaling factor in each color channel.

In our system the environment map is used to compute both the sky illumination $\mathbf{S}_{\text{sky}}$ and part of the indirect illumination $\mathbf{S}_{\text{ind}}$, the other part being computed from the geometric proxy (see Section 3.3 for more details). We estimate the color transfer function of the reflective sphere $\mathbf{K}$ (represented as a RGB vector) by taking a photograph of a neutral gray card with known reflectance $\mathbf{R}$ placed in sun shadow (Figure 3.6b), corresponding to the case $v_{\text{sun}} = 0$. We intentionally place the card at a position where we expect its geometry to be well reconstructed (Figure 3.6a), which allows us to estimate its illumination using the proxy geometry. From the image formation model we have:

$$\begin{aligned} \mathbf{I} &= \mathbf{R} * (\mathbf{S}_{\text{sky}} + \mathbf{S}_{\text{ind}}) \\ &= \mathbf{R} * (\mathbf{K} * \mathbf{S}^{\text{env}}_{\text{sky}} + \mathbf{K} * \mathbf{S}^{\text{env}}_{\text{ind}} + \mathbf{S}^{\text{proxy}}_{\text{ind}}) \end{aligned} \tag{3.7}$$

where $\mathbf{S}^{\text{env}}_{\cdot}$ denotes the illumination terms computed from the environment map and $\mathbf{S}^{\text{proxy}}_{\cdot}$ those computed from the geometric proxy and environment map. $\mathbf{I}$ is the radiance of the gray card, which can be looked up in the captured photograph. We can estimate $\mathbf{K}$ from this equation, since it is the only unknown.

We similarly recover the sun radiance $\mathbf{L}_{\text{sun}}$ by taking a second picture of the gray card placed in sunlight (Figure 3.6c). From this picture we have:

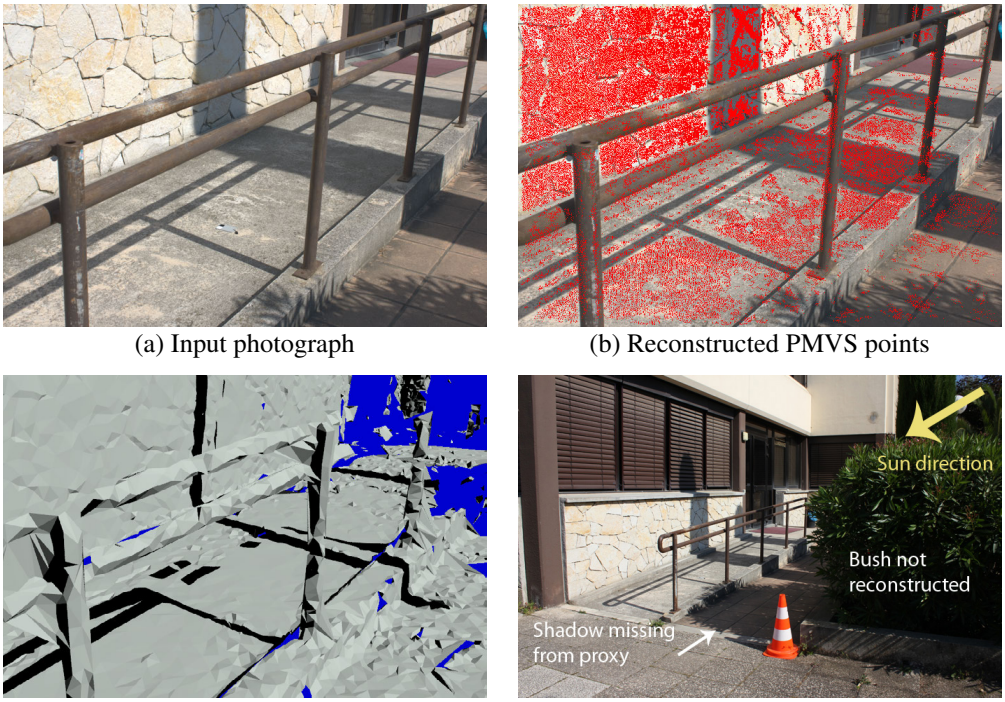
$$\begin{aligned} \mathbf{I} &= \mathbf{R} * (\mathbf{S}_{\text{sun}} + \mathbf{S}_{\text{sky}} + \mathbf{S}_{\text{ind}}) \\ &= \mathbf{R} * (v_{\text{sun}} \cos\theta_{\text{sun}} \mathbf{L}_{\text{sun}} + \mathbf{S}_{\text{sky}} + \mathbf{S}_{\text{ind}}) \end{aligned} \tag{3.8}$$

where $v_{\text{sun}} = 1$ if the gray target is placed in direct sunlight, θ_{sun} is the angle between the scene normal and the sun direction previously specified by the user. Only $\mathbf{L}_{\text{sun}}$ is unknown.

Output. The output of the capture and calibration steps is:

- a moderately dense point cloud reconstruction of the scene which we will refer to as the *PMVS points* (Figure 3.7b)
- a very approximate geometric *proxy*, which often contains significant geometric errors (Figure 3.7c)
- the direction and radiance of the sun and a correctly aligned and scaled HDR environment map containing the sky and distant indirect radiance (Figure 3.4b).

Note that Chapter 4 focuses on simplifying these capture and calibration steps.



(a) Input photograph (b) Reconstructed PMVS points

(c) Proxy shadows (d) Shadow configuration

Figure 3.7: *Multiview geometry provides us with a sparse oriented point cloud (b, red pixels), from which we generate a mesh that represents the scene geometry. Compared to the original image (a), the initial guess of the sun shadows computed using the proxy reconstructed from the point cloud is very inaccurate due to geometric errors, and incomplete (c). In particular, a bush casting a large shadow on the floor (d) is not reconstructed.*

3.3 Geometry-Based Computation

We describe here how to compute sun, sky and indirect illumination values for each PMVS point. These points have been generated using multiview stereo and also have normals (Section 3.2.2).

We first compute sun illumination $\hat{\mathbf{S}}_{\text{sun}}$ ignoring cast shadows, i.e., unoccluded sunlight. We already know the required quantities for this computation, i.e., the normal at the point and the sun radiance and direction. Treating sun visibility requires much higher precision than the one provided by the proxy, and is treated separately in Section 3.4.

We then compute sky and indirect illumination at each PMVS point. Figure 3.8 illustrates this computation that we detail below. In a nutshell, we use the HDR environment map to compute both sky illumination and *distant* indirect illumination, while we compute the *near-field* indirect illumination from the proxy geometry. Note however that we do not need to know the reflectance of the proxy for this step, we instead use the captured photographs to recover the necessary outgoing radiance over the geometry.

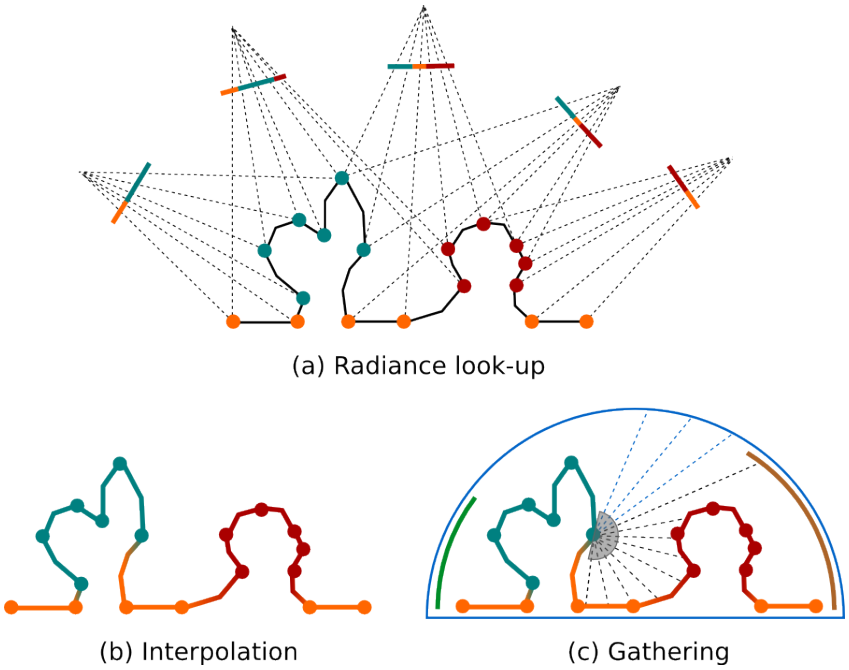


Figure 3.8: *Evaluation of sky and indirect illumination at each PMVS point. We look up the radiance value of the PMVS points in each image and average the values over the images where they are visible (a). We then project and interpolate this radiance over the proxy geometry (b). Finally we gather the sky and indirect illumination at each PMVS point by shooting rays which sample the hemisphere (c, dotted lines). Rays that reach sky pixels in the environment map (c, blue lines) contribute to sky illumination, while other rays contribute to indirect illumination (c, dark lines).*

(a) Input photograph (b) Colored proxy

Figure 3.9: *We estimate sky and indirect illumination at PMVS points using a colored proxy (b), obtained by projecting and interpolating the radiance of PMVS points on the reconstructed geometry. When viewed from the estimated position of the camera, the proxy resembles a coarse version of the input image (a).*

Assigning radiance to the proxy geometry. We first assign radiance to each PMVS point by looking up its pixel values in each HDR image where it appears (Figure 3.8a). We assign the average value as the outgoing radiance of the point, which is assumed constant in all directions. We then assign the radiance of the closest PMVS point to each vertex of the proxy, and interpolate these values over the faces of the mesh (Figure 3.8b). This step yields a colored proxy that stores the outgoing radiance, as captured in the photographs (Figure 3.9). This colored proxy allows easy computation of indirect illumination, avoiding complex multiple-bounce estimation typical of inverse global illumination approaches, since it stores the *outgoing radiance* $\mathbf{L}_{\text{ind}}$ at each point instead of its *reflectance*.

Sky and indirect illumination. We separate the environment map into two regions (see Figure 3.4b), one for the sky and the other one for distant objects. At every PMVS point we cast a set of rays towards the hemisphere (Figure 3.8c). If a ray reaches a sky pixel of the environment map, the resulting radiance contributes to sky illumination $\mathbf{S}_{\text{sky}}$. Otherwise, the ray intersects the proxy or hits a non-sky pixel of the environment map, and therefore contributes to indirect illumination $\mathbf{S}_{\text{ind}}$. If the ray hits the proxy, it uses the radiance assigned to the geometry as described in the previous paragraph.

We rewrite Equations 3.3 and 3.4 to express this computation, changing the integration domain to the entire hemisphere Ω for both integrals, and introducing a visibility term $v_{\text{sky}}(\omega)$ which is 1 when the ray in direction ω reaches the sky, and 0 otherwise (it contributes to indirect illumination):

$$\mathbf{S}_{\text{sky}} = \int_{\Omega} v_{\text{sky}}(\omega) \cos\theta_{\omega} \mathbf{L}_{\text{sky}}(\omega)\, d\omega \tag{3.9}$$

$$\mathbf{S}_{\text{ind}} = \int_{\Omega} (1 - v_{\text{sky}}(\omega)) \cos\theta_{\omega} \mathbf{L}_{\text{ind}}(\omega)\, d\omega. \tag{3.10}$$

The computation of Equations 3.9 and 3.10 is robust to the coarse geometry of the proxy since the integration over Ω averages the values over the entire hemisphere. We implement this computation with a custom renderer in the PBRT stochastic raytracer [Pharr 2010].

Our approach shares similarities with the techniques of Yu and Malik [Yu 1998], but while their method was designed for accurate geometry constructed manually, we handle sparse incomplete geometry reconstructed automatically.

Finally we compute an initial guess of the sun visibility $v_{\text{sun}}^{\text{init}}$ at each PMVS point by intersecting a ray with the the proxy geometry in the direction of the sun. Note however that this visibility test is very sensitive to errors in the reconstructed proxy, as illustrated in Figure 3.7c. This fact underlines the importance of accurately estimating v_{sun} and we next show how to refine this initial estimate.

3.4 Estimating Sun Visibility at 3D Points

One key contribution of this approach is a novel algorithm for identifying visibility v_{sun} with respect to the sun for each PMVS point. In the following, we wish to decompose one photograph and thus consider only PMVS points visible from the corresponding viewpoint.

From our image formation model in Equations 3.2 and 3.5 we express the reflectance at each PMVS point as a function of the visibility term:

$$\mathbf{R}(v_{\text{sun}}) = \frac{\mathbf{I}}{(v_{\text{sun}}\hat{\mathbf{S}}_{\text{sun}} + \mathbf{S}_{\text{sky}} + \mathbf{S}_{\text{ind}})} \tag{3.11}$$

where $\mathbf{I}$ is the RGB pixel value in the image we wish to process, $\mathbf{S}_{\text{sky}}$, $\mathbf{S}_{\text{ind}}$ and $\hat{\mathbf{S}}_{\text{sun}}$ are the illumination values of the corresponding PMVS point computed in Section 3.3, and the division is per-channel. With this parameterization, varying v_{sun} in $[0, 1]$ generates a *curve of candidate reflectances* in RGB space. Our goal is to find the reflectance $\mathbf{R}$ (or correspondingly visibility v_{sun}) of each PMVS point; by construction, this reflectance lies on the *candidate curve* corresponding to that PMVS point.

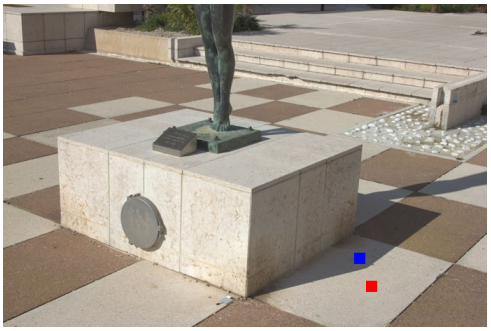
The intuition of our approach is that multiple PMVS points sharing the same reflectance will generate intersecting curves in color space; their (shared) reflectance will be the color where the candidate curves intersect (Figure 3.10a-b). By finding these intersections we can deduce the value of v_{sun} for the PMVS point corresponding to each curve.

However, imprecision in the capture process and in the geometry-based computation prevents the curves from perfectly intersecting in color space. In addition, multiple intersections can occur along a curve, giving multiple candidates for the visibility. We address these issues with a robust iterative procedure inspired by the *mean shift* algorithm.

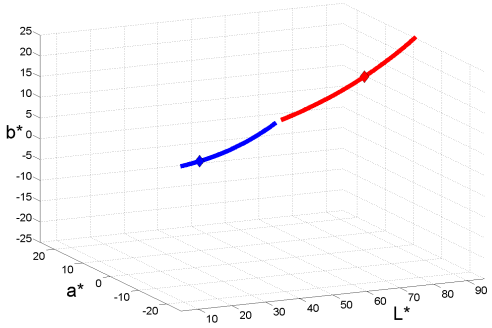
Overview. Mean shift [Fukunaga 1975, Comaniciu 2002] is a non-parametric mode-seeking algorithm that aims to locate the maxima of a density function, given a set of *data points*. First, a *kernel* (or window) is placed at each data point; it represents the region of influence of this point. In an iterative process, each kernel is then moved in a direction that increases the local density, computed as the weighted average of nearby data points. The process stops when all kernels have reached a stationary point (or mode).

In our approach, we define an asymmetric region of influence for each candidate curve. We use *mean shift iterations* to maximize an energy that measures the overlap among pairs of curves, and iteratively update the estimated reflectances while constraining them to lie on their candidate curves. After convergence, for each curve we obtain the reflectance (and corresponding visibility) that tends to maximize the number of PMVS points sharing a similar reflectance, and correspondingly tends to minimize the number of reflectances in the scene.

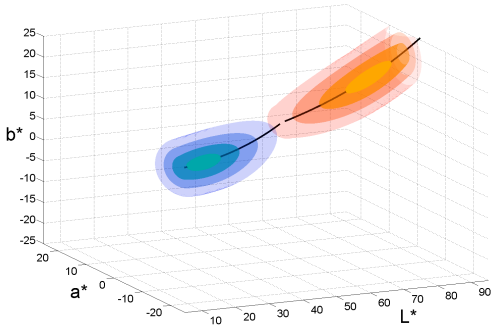
This algorithm assumes that the scene is composed of a sparse set of reflectances shared by multiple points, which is a common assumption recently used in image segmentation [Omer 2004] and white balance algorithms [Hsu 2008].



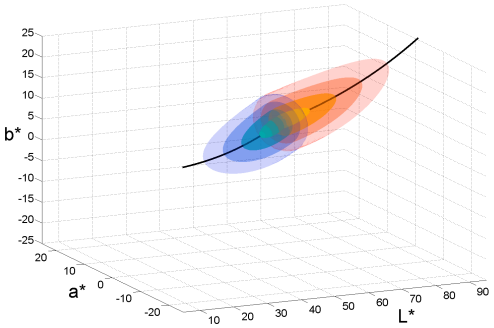
(a) Selected PMVS points



(b) Candidate reflectance curves



(c) Regions of influence at initialization



(d) Regions of influence after one iteration

Figure 3.10: *Multiple PMVS points sharing the same reflectance will generate intersecting curves in color space. (a) We selected two PMVS points with the same reflectance but different illuminations (red and blue squares). (b) The corresponding candidate reflectance curves (nearly) intersect at one end, which corresponds to the reflectance of both points. Diamond markers on the curves correspond to the initial guess for visibility (randomly set in this example). (c) Each curve affects a region of the surrounding color space, according to Equation 3.12 . Regions of influence at initialization are illustrated as isosurfaces with varying opacity. Regions closer to the curve and the current visibility estimate are more affected. (d) After one iteration, the visibility estimates have moved towards the intersecting end of the curves, increasing the overlap between the regions of influence (i.e., the energy E_{total} in Equation 3.14).*

Region of influence. Equation 3.11 defines the candidate reflectances of a PMVS point as a rational curve that is parameterized non-uniformly by v_{sun}. In order to obtain a uniform parametrization we first approximate each curve c as a piecewise linear curve in CIE L*a*b* space, and parameterize it by arc length from the shadowed end of the curve: t in $[0, 1]$ so that $\mathbf{R}(v_{\text{sun}} = 0) = \mathbf{R}(t = 0)$. We use this uniform parametrization to compute distances along the curve. We chose to work in CIE L*a*b* space because it defines a perceptually uniform distance metric.

We then define the influence of curve c on a point $\mathbf{x}$ of color space as:

$$A_{\mathbf{x}c} = h_{\perp}\left(\frac{d_{\perp}^2(\mathbf{x},c)}{\sigma_{\perp}^2}\right) h_{\parallel}\left(\frac{d_{\parallel}^2(\mathbf{x},c)}{\sigma_{\parallel}^2}\right) \tag{3.12}$$

where:

- $d_{\perp}^2(\mathbf{x},c) = \|\mathbf{x} - proj_{3D}(\mathbf{x},c)\|^2$ is the squared *distance perpendicular to the curve*, defined by squared distance in color space between $\mathbf{x}$ and its projection on curve c.
- $d_{\parallel}^2(\mathbf{x},c) = (t_c - proj_t(\mathbf{x},c))^2$ is the squared *distance along the curve*, defined by squared difference between the arc length t_c (i.e., the position of the current reflectance estimate along the curve) and the arc length $proj_t(\mathbf{x},c)$ (i.e., the projection of $\mathbf{x}$ on the curve).
- $h_{\perp}$ and $h_{\parallel}$ are Gaussian kernel profiles with the form $h(x) = e^{-x}$ controlled by the standard deviations $\sigma_{\perp}^2$ and $\sigma_{\parallel}^2$ (larger σ values correspond to a wider region of influence).

The first term in Equation 3.12 compensates for curves that do not exactly intersect, by defining a region of influence around the curve with a Gaussian falloff orthogonal to the curve. The second factor makes our algorithm robust to "false intersections", i.e., intersections of curves that in fact do not share the same reflectance. Due to the Gaussian kernel $h_{\parallel}$, each curve only influences regions close to its current reflectance estimate. As the reflectance estimate converges toward the most likely reflectance, the value of $h_{\parallel}$ at intersections lying at other positions along the curve will decrease.

The regions of influence of two curves are illustrated in Figure 3.10c.

Energy Formulation. We then define an energy that measures the overlap between the regions of influence of pairs of curves:

$$E = \int_V \left(\sum_{c\in C}\sum_{c'\neq c} A_{\mathbf{x}c}A_{\mathbf{x}c'}\right) d\mathbf{x} \tag{3.13}$$

where we integrate over the entire color space V.

We evenly discretize the 3D color space into a set S of samples, and rewrite this energy as a discrete sum:

$$E_{\text{total}} = \sum_{\mathbf{s}\in S} E_{\text{sample}}(\mathbf{s}) \tag{3.14}$$

where the energy of a sample $E_{\text{sample}}(\mathbf{s})$ accumulates the contribution of each pair of curves (c, c') intersecting nearby:

$$E_{\text{sample}}(\mathbf{s}) = \sum_{c\in C}\sum_{c'\neq c} A_{\mathbf{s}c}A_{\mathbf{s}c'}.$$

With this formulation, two curves will contribute to the energy of a sample only if they (almost) intersect near this sample.

Derivation of the energy gradient. We seek the positions of the reflectance estimates along the curves t_c for $c \in C$ that maximize E_{total}:

$$\arg\max_{t_c} E_{\text{total}} \tag{3.15}$$

i.e., that are located at the zeros of the gradient function.

The derivative of E_{total} with respect to t_c is given by:

$$\frac{\partial E_{\text{total}}}{\partial t_c} = \frac{4}{\sigma_{\parallel}^2} \sum_{\mathbf{s} \in S} \sum_{c' \neq c} (proj_t(\mathbf{s}, c) - t_c) \, A_{\mathbf{s}c} A_{\mathbf{s}c'} \tag{3.16}$$

Setting Equation 3.16 to 0 for all $c \in C$ gives:

$$\begin{aligned} &\frac{\partial E_{\text{total}}}{\partial t_c} = 0 \quad \Leftrightarrow \\ &\left(\sum_{\mathbf{s} \in S} \sum_{c \neq c'} A_{\mathbf{s}c} A_{\mathbf{s}c'} \right) \\ &\left(t_c - \frac{\sum_{\mathbf{s} \in S} proj_t(\mathbf{s}, c) \left(\sum_{c' \neq c} A_{\mathbf{s}c} A_{\mathbf{s}c'} \right)}{\sum_{\mathbf{s} \in S} \sum_{c' \neq c} A_{\mathbf{s}c} A_{\mathbf{s}c'}} \right) = 0 \end{aligned}$$

which is analogous to the form obtained in the mean-shift algorithm [Comaniciu 2002].

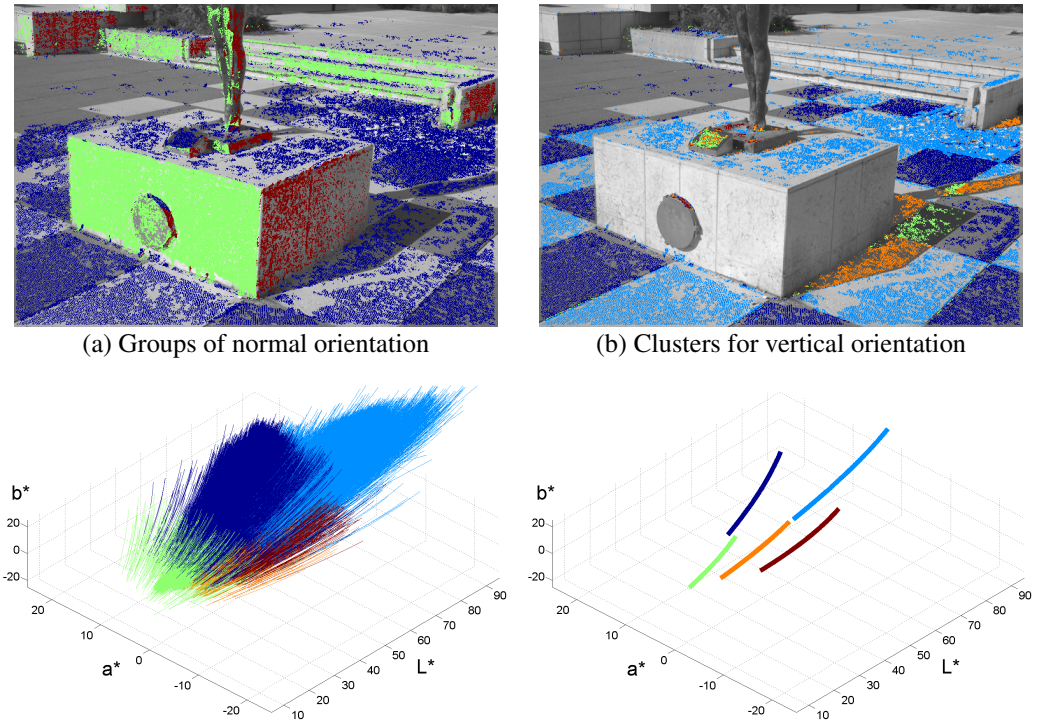
Iterative process. We define our iterative procedure recursively by computing at iteration i the weights $A_{\mathbf{s}c}^i$ (i.e., the regions of influence) using the estimates t_c^i for all curves $c \in C$, then updating the estimates t_c^{i+1} for each curve c using a weighted average of the projection of nearby samples on c:

$$t_c^{i+1} = \frac{\sum_{\mathbf{s} \in S} proj_t(\mathbf{s}, c) \left(\sum_{c' \neq c} A_{\mathbf{s}c}^i A_{\mathbf{s}c'}^i \right)}{\sum_{\mathbf{s} \in S} \sum_{c' \neq c} A_{\mathbf{s}c}^i A_{\mathbf{s}c'}^i} \tag{3.17}$$

We initialize t_c^0 using the initial guess of the sun visibility $v_{\text{sun}}^{\text{init}}$, obtained in Section 3.3 by casting the shadow of the geometric proxy. We ensure that this iterative process converges to a maximizer by checking that the energy does not decrease after each iteration [Comaniciu 2000]; if $E_{\text{total}}^{i+1} < E_{\text{total}}^i$ we set $t_c^{i+1} = (t_c^{i+1} + t_c^i)/2$ for all c and iterate. The algorithm stops when $|t_c^{i+1} - t_c^i|$ is small enough for all c.

As a result of this process, the reflectance estimates converge to regions of space where many curves intersect, while being constrained to lie on their candidate curves due to our parameterization. This is illustrated in Figure 3.10d.

Orientation separation. We observed that false intersections sometimes occur when two curves of different reflectances have different orientations, sky or indirect illumination. In order to further reduce the presence of such intersections, we separate PMVS points



(a) Groups of normal orientation (b) Clusters for vertical orientation

(c) Candidate curves before clustering (d) Candidate curves after clustering

Figure 3.11: *Orientation separation and curve clustering. (a) PMVS points are separated into three groups according to the orientation of their normals. Each group is then processed independently. (b-c) Within one group (in this illustration, points with vertical normals, i.e. blue pixels in (a)), PMVS points are clustered according to the endpoints of their candidate curves. (d) After clustering, pairs of representative curves intersect as expected (orange and light blue, green and dark blue), since each pair represents PMVS points sharing the same reflectance.*

into separate groups based on the orientation of their normals; using three groups worked well in our experiments. We first apply mean-shift clustering [Comaniciu 2002] where the 3D feature vectors contain the normals of the points. The first two clusters correspond to the two dominant groups of normal orientations in the scene (Figure 3.11a, blue and green), while the remaining points form a third group (Figure 3.11a, red). We then run our optimization independently on each group.

Curve clustering. Even within one group of orientations, the candidate curves corresponding to the numerous PMVS points (up to dozens of thousands, in our scenes) tend to intersect each other at several positions along the curves (Figure 3.11c), since many similar (but not equal) reflectances might be present in the scene. We enforce sparsity in our algorithm by grouping all the curves corresponding to a similar reflectance and illumination (Figure 3.11b). Note that this grouping is only used to identify the sun visibility, we do not enforce sparsity in the final reflectance image.

We use mean-shift clustering to perform this grouping, representing each curve as a 6D feature vector that contains the L*a*b* reflectances of its endpoints. Clusters that

contain a small number of curves (less than 1% of the largest cluster) are discarded, and the corresponding PMVS points are ignored in the remaining of the algorithm. This leads to the clustering shown in Figure 3.11b-c.

Finally, we replace all the curves belonging to each cluster by one *representative curve*: the curve closest to the median of this cluster's feature vectors. The initial guess of the sun visibility $v_{\text{sun}}^{\text{init}}$ is assigned to the median visibility of each cluster. This process greatly simplifies and cleans up the set of curves, and lowers the computational cost of our algorithm. As shown in Figure 3.11d, the curves corresponding to groups of PMVS points with similar reflectances (nearly) intersect after clustering.

After clustering, we run the optimization to maximize E_{total}. Upon convergence, for each cluster we obtain the position of the estimated reflectance along its representative curve t^{final}. This position value is assigned to all curves belonging to this cluster, from which we can deduce the final estimated reflectance of each PMVS point corresponding to these curves.

We found that clustering candidate curves largely improves the stability of the algorithm. The clustering groups candidate curves corresponding to PMVS points with same reflectance *and* same illumination, while our subsequent optimization identifies the reflectances shared between clusters with different illumination. We show more examples of candidate curves after clustering in the appendix.

At the end of this process, we obtain a list of PMVS points for which the position along the curve t, the sun visibility v_{sun} and the reflectance $\mathbf{R}$ have been estimated.

Implementation. In practice we use a truncated kernel profile $h_\perp$ (Equation 3.12) so that $h_\perp(x) = 0$ when $x > \lambda$ (we use $\lambda = 3$, which discards negligible values). This means that each curve will only influence a limited number of samples; for each curve c we can precompute the indices of these samples, as well as their orthogonal distance $\|\mathbf{s} - proj_{3D}(\mathbf{s}, c)\|$ and position of their projection along the curve $proj_t(\mathbf{s}, c)$.

We evenly discretized the CIE L*a*b* color space into $60 \times 36 \times 36$ samples, with L* in $[5, 95]$, a* in $[-25, 25]$, b* in $[-25, 25]$. We used a fixed bandwidth for the curve clustering using 6D mean-shift clustering, as well as for the 3D mean-shift clustering for orientation separation.

The clustering is the most costly part of the algorithm and takes from 25 seconds to a few minutes with our Matlab implementation, depending on the number of PMVS points. Once the clustering has been performed, the iterative optimization takes around 10 seconds, which allowed us to test many parameters for $\sigma_\perp^2$ and $\sigma_\parallel^2$.

We found that the algorithm produces good results for a wide range of parameters, and that the best values were scene-dependent. In our experiments we found that large values of $\sigma_\perp^2$ can compensate for calibration errors that prevent reflectance curves from perfectly intersecting, while large values of $\sigma_\parallel^2$ can compensate for the erroneous initialization of

$v_{\text{sun}}^{\text{init}}$ provided by the approximate proxy. However, σ values should remain small enough to prevent curves of different materials from influencing one another. Table 3.1 summarizes the parameter values used for the examples in this chapter.

	Statue	Rocks	Ramp	Stairs
$(\sigma_{\parallel}^2, \sigma_{\perp}^2)$	$(0.1, 1)$	$(0.01, 1)$	$(0.3, 1)$	$(0.01, 20)$
	$(0.1, 5)$	$(0.01, 5)$	$(0.3, 5)$	

Table 3.1: *Sets of region of influence parameters used on the four scenes.*

3.5 Estimating Illumination at Each Pixel

In previous steps, we have used multiview stereo methods to generate a sparse set of 3D points on which we compute the illumination values $\hat{\mathbf{S}}_{\text{sun}}$, $\mathbf{S}_{\text{sky}}$ and $\mathbf{S}_{\text{ind}}$ (Section 3.3) along with the visibility of the sun v_{sun} (Section 3.4). We next show how to leverage image-guided propagation methods to assign reflectance and illumination values to all pixels in the input photographs. We first show how to propagate the total illumination $\mathbf{S}_{\text{total}}$ and then describe a method to subsequently separate the contribution of each lighting component (i.e., sun, sky and indirect).

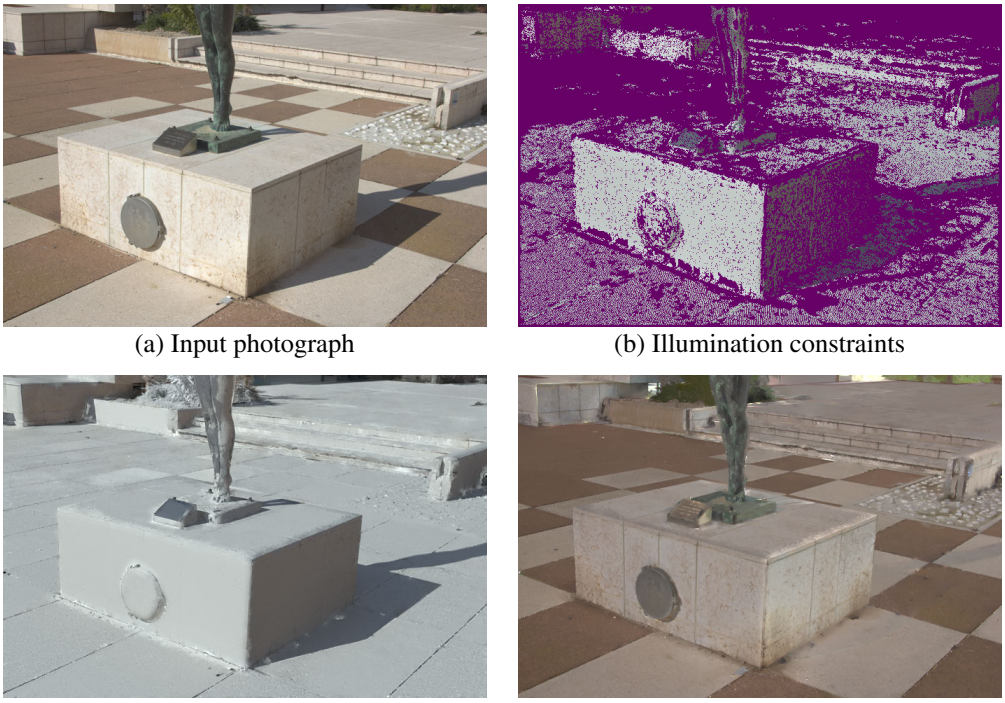
3.5.1 Image Guided Propagation

We use the intrinsic images algorithm of Bousseau et al. [Bousseau 2009] that was designed to propagate user indications for separating reflectance and illumination in a single image. This algorithm makes the intrinsic image decomposition tractable by assuming that the reflectance values in a pixel neighborhood lie in a plane in color space. This planar reflectance assumption translates to a set of linear equations so that the illumination image is expressed as the minimizer of a least-square energy

$$\arg\min_{\bar{\mathbf{S}}_{\text{total}}} \bar{\mathbf{S}}_{\text{total}}^T M \bar{\mathbf{S}}_{\text{total}} \tag{3.18}$$

where the vector $\bar{\mathbf{S}}_{\text{total}}$ stacks the pixels of the unknown illumination image and the matrix M encodes the planar reflectance assumption (see the paper [Bousseau 2009] for the complete derivation). For colored illumination the optimization is solved for each color channel separately.

In their original paper, Bousseau et al. constrain the least-square system with user indications. Users can specify the value of $\bar{\mathbf{S}}_{\text{total}}$ over a few pixels or indicate that several pixels share the same illumination or reflectance. In our approach we use instead the illumination and visibility estimated at PMVS points to constrain the optimization. We express these



(a) Input photograph (b) Illumination constraints

(c) Estimated total illumination (d) Estimated reflectance

Figure 3.12: *Separation of reflectance and total illumination. The optimization procedure described in Equation 3.19 enforces illumination constraints at PMVS points (b) and propagates them to all pixels, in order to separate a photograph (a) into total illumination (c) and reflectance (d).*

constraints as an additional quadratic penalty

$$\arg\min_{\bar{\mathbf{S}}_{\text{total}}} \bar{\mathbf{S}}_{\text{total}}^T M \bar{\mathbf{S}}_{\text{total}} + w \sum_{p \in \mathcal{P}} (\bar{\mathbf{S}}_{\text{total}}^p - \mathbf{S}_{\text{total}}^p)^2 \tag{3.19}$$

where $\mathcal{P}$ is the set of pixels covered by PMVS points and $\mathbf{S}_{\text{total}}^p = v_{\text{sun}}^p \hat{\mathbf{S}}_{\text{sun}}^p + \mathbf{S}_{\text{sky}}^p + \mathbf{S}_{\text{ind}}^p$ their illumination values computed in Sections 3.3 and 3.4. The weight w controls the importance of the constraints, we use $w = 1$ to give equal importance to the constraints and the propagation model.

Figure 3.12 shows the results of the image guided propagation. This result shows the power of our approach. We exploit the information provided by the sparse and imprecise geometry to automatically generate constraints for the propagation of [Bousseau 2009], thus eliminating the need for user scribbles. A visualization of the constraints for all scenes can be found in the appendix. While our method supports user scribbles to guide the decomposition in regions that could not be reconstructed (e.g., trees, specular objects), all our results have been generated using only the automatically computed constraints.

3.5.2 Light Source Separation

Given the estimated illumination image $\bar{\mathbf{S}}_{\text{total}}$, we wish to separate the contribution of each illumination component $\bar{\mathbf{S}}_{\text{sun}}$, $\bar{\mathbf{S}}_{\text{sky}}$ and $\bar{\mathbf{S}}_{\text{ind}}$. Inspired by previous work on white balance under mixed lighting [Hsu 2008], we express our light source separation as two successive matting problems. We first decompose the illumination into a sun component and a *diffuse* component $\bar{\mathbf{S}}_{\text{diff}}$ that includes the contribution of both sky and indirect lighting. In a second step we decompose the diffuse component into its two terms.

We first express each illumination term $\bar{\mathbf{S}}$ as the product between a scalar intensity $\mathbf{s} = \|\bar{\mathbf{S}}\|$ and a chromaticity $\mathbf{C} = \bar{\mathbf{S}}/\|\bar{\mathbf{S}}\|$, so that:

$$\begin{aligned} \bar{\mathbf{S}}_{\text{total}} &= \mathbf{s}_{\text{sun}}\mathbf{C}_{\text{sun}} + \mathbf{s}_{\text{sky}}\mathbf{C}_{\text{sky}} + \mathbf{s}_{\text{ind}}\mathbf{C}_{\text{ind}} \\ &= \mathbf{s}_{\text{sun}}\mathbf{C}_{\text{sun}} + \mathbf{s}_{\text{diff}}\mathbf{C}_{\text{diff}}. \end{aligned} \tag{3.20}$$

where the low-frequency component $\mathbf{S}_{\text{diff}} = \mathbf{S}_{\text{sky}} + \mathbf{S}_{\text{ind}}$ sums the contribution of sky and indirect illumination. Denoting $\alpha = \mathbf{s}_{\text{sun}}/(\mathbf{s}_{\text{sun}} + \mathbf{s}_{\text{diff}})$ we express the illumination image values at each pixel as a mixture between two values weighted by α:

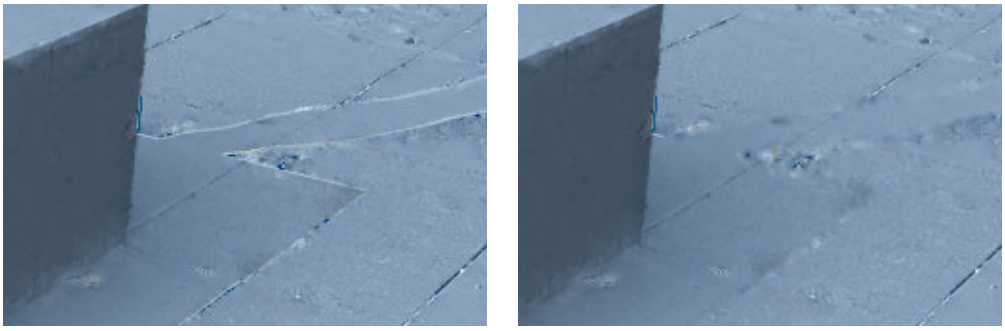
$$\begin{aligned} \bar{\mathbf{S}}_{\text{total}} &= \alpha(\mathbf{s}_{\text{sun}} + \mathbf{s}_{\text{diff}})\mathbf{C}_{\text{sun}} \\ &\quad +(1-\alpha)(\mathbf{s}_{\text{sun}} + \mathbf{s}_{\text{diff}})\mathbf{C}_{\text{diff}} \qquad (3.21) \\ &= \alpha\mathbf{F} + (1-\alpha)\mathbf{B} \qquad (3.22) \end{aligned}$$

We can now recover $\bar{\mathbf{S}}_{\text{sun}} = \alpha\mathbf{F}$ and $\bar{\mathbf{S}}_{\text{diff}} = (1-\alpha)\mathbf{B}$ by solving a standard matting problem. We compute α at each PMVS point from the illumination values estimated in Sections 3.3 and 3.4. We then propagate α over the image $\bar{\mathbf{S}}_{\text{total}}$ using the *matting Laplacian* algorithm of Levin et al. [Levin 2008]. Finally, given α at every pixel and the known $\mathbf{S}_{\text{sun}}$ and $\mathbf{S}_{\text{diff}}$ at each PMVS point, we solve for the sun and diffuse illumination images with the following least-square optimization:

$$\begin{aligned} \arg\min_{\mathbf{F},\mathbf{B}} \sum_{i\in\mathcal{I}} \Big(& \Big(\bar{\mathbf{S}}^i_{\text{total}} - (\alpha^i\mathbf{F}^i + (1-\alpha^i)\mathbf{B}^i)\Big)^2 \\ & + \lambda\Big((\mathbf{F}^i_x)^2 + (\mathbf{F}^i_y)^2 + (\mathbf{B}^i_x)^2 + (\mathbf{B}^i_y)^2\Big)\Big) \\ & + w \sum_{p\in\mathcal{P}} \left(\alpha^p\mathbf{F}^p - \mathbf{S}^p_{\text{sun}}\right)^2 + \left((1-\alpha^p)\mathbf{B}^p - \mathbf{S}^p_{\text{diff}}\right)^2 \end{aligned}$$

where $\mathcal{I}$ is the image domain, $\mathcal{P}$ is the set of pixels covered by PMVS points and $\mathbf{F}_x$, $\mathbf{F}_y$, $\mathbf{B}_x$ and $\mathbf{B}_y$ are the x and y derivatives of $\mathbf{F}$ and $\mathbf{B}$ computed with finite differences.

The first term of this functional ensures that the decomposition explains the input illumination $\bar{\mathbf{S}}_{\text{total}}$ and follows the estimated ratio α. The second term adds a smoothness regularization on each component while the third term constrains the solution to agree with the illumination values computed at PMVS points. We used $\lambda = 0.1$ and $w = 0.01$ for all



(a) Sky illumination before inpainting (b) Sky illumination after inpainting

Figure 3.13: *The reflectance, sky illumination (here, close up) and indirect illumination estimated by our algorithm can contain residual variations along hard shadow boundaries (a). We use inpainting to remove these artifacts (b).*

our results.

As a second step we apply the same matting approach to further separate the diffuse illumination $\bar{\mathbf{S}}_{\text{diff}}$ as the sum of the sky illumination $\bar{\mathbf{S}}_{\text{sky}}$ and the indirect illumination $\bar{\mathbf{S}}_{\text{ind}}$. We show in Figure 3.14 and in the appendix the results of this decomposition. The overall decomposition takes around 90 seconds to compute for a 3.2 megapixel image, where 30 seconds are necessary to compute $\bar{\mathbf{S}}_{\text{total}}$ and 60 seconds to perform the two subsequent separations.

Inpainting. This overall process gives a satisfactory decomposition in the scenes we have tested. However, a small residual border can remain around the hard shadow boundary (see Figure 3.13 (left)), a common artifact of shadow removal [Finlayson 2004, Wu 2007b]. We identify these shadow boundaries as pixels located on sun illumination discontinuities but not on normal discontinuities. We first propagate normals from the PMVS points over the image using the method of Okabe et al. [Okabe 2006]. We then run an edge detector over the sun illumination image and label edge pixels that do not correspond to edges in the normal image. We remove the labeled pixels and their immediate neighbors from the reflectance, sky and indirect illumination images and use inpainting to fill in the holes (see Figure 3.13 (right)). This post-process takes 60 seconds per image on average.

3.6 Results and Discussion

3.6.1 Rich intrinsic decomposition results

In Figure 3.14 we show results on four different scenes. For all results, we show reflectance, sun, sky and indirect illumination layers. The number of photographs used for each scene is shown in Table 3.2. The estimated illumination at PMVS points and additional views for each scene are shown in the appendix.

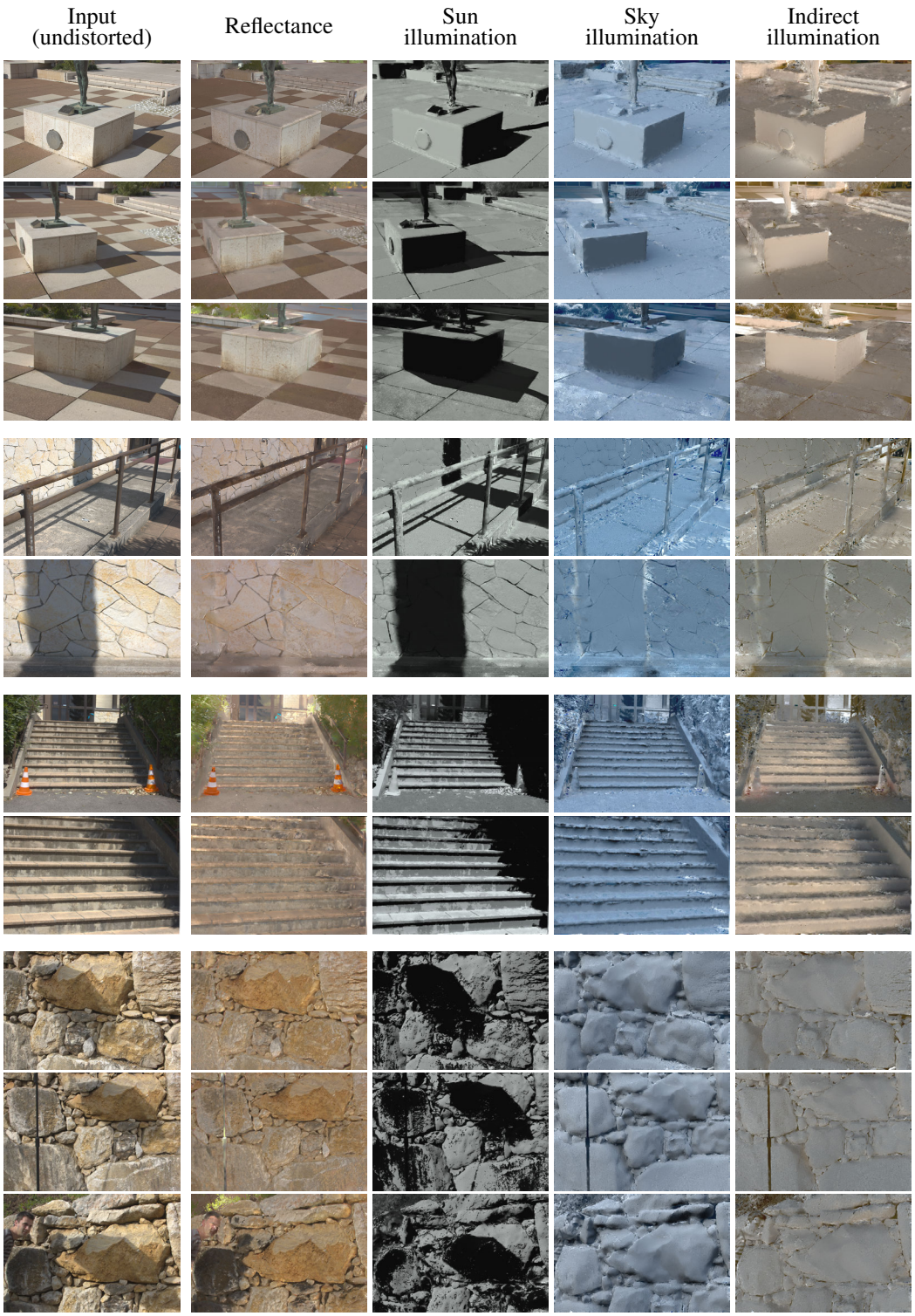


Figure 3.14: *Results of our rich intrinsic decomposition on four scenes: Statue, Ramp, Stairs, Rocks. We adjusted the brightness of images for illustration purposes (the scaling factors used can be found in the appendix). For each scene, the sun illumination is usually much more intense than sky illumination on average, and the sky illumination is more intense than indirect illumination on average. The bottom row illustrates a failure case, where a spurious character appeared in one view only and therefore was not reconstructed at all.*

The first scene shows the base of a statue on a square; we took HDR images for three different views and we show results for all three. The reflectance layer is plausible in all views. The sun and sky layers have been successfully separated in all cases, however the third view shows a slight color shift of the reflectance in the shadow area; this is due to the insufficient number of PMVS points on the ground. Please refer to the appendix for an illustration of the distribution of 3D points. The indirect layer clearly shows the indirect light bouncing off the front of the base (first and second views) which is in direct sunlight. The sides of the base do not receive the same amount of sky and indirect illumination due to the configuration of the scene (see Figure 3.4a).

The second scene is challenging, since the reconstruction process is unable to capture details of the railings and does not reconstruct the vegetation casting the main shadows (see also Figure 3.7d). In this view, despite the intricate geometric configuration, our method successfully removes shadows from the reflectance layer, and the three other layers show good results. In the third scene, a staircase is shown. There are some residual artifacts at the shadow boundaries because the vegetation moved in the breeze during HDR acquisition. In the fourth scene, an umbrella casts a shadow onto a rock wall. Notice how the indirect layer well represents lighting in the cracks between rocks where neither sun nor skylight is present.

Statue	Rocks	Ramp	Stairs
30	11	31	10

Table 3.2: *Number of photographs captured for the geometric reconstruction, for each scene in this chapter.*

3.6.2 Comparisons

We compare our approach to three state-of-the-art methods in Figure 3.15. All these methods take a single image as input. The user-assisted approach of Bousseau et al. [Bousseau 2009] produces results of a similar quality to ours, but requires a significant number of user indications (between 25 and 105 scribbles). The automatic method of Shen et al. [Shen 2008] is able to extract most of the illumination variations but colored shadows remain in the reflectance image. These colored residuals are due to the variation in color between sun and sky illumination, which violates the gray illumination assumption of this method. Residual shadows are also present in the reflectance estimated with the automatic method of Shen et al. [Shen 2011a], as well as reflectance residuals in the illumination image (tiled floor in the statue scene). Although this method can support user scribbles, the authors reported that scribbles did not improve the result significantly in these examples. All the results of these comparisons have been kindly provided by the respective authors. While these approaches can be applied to single images from various sources, to our knowledge ours is the first to provide a richer decomposition by separating the illumination into different components.

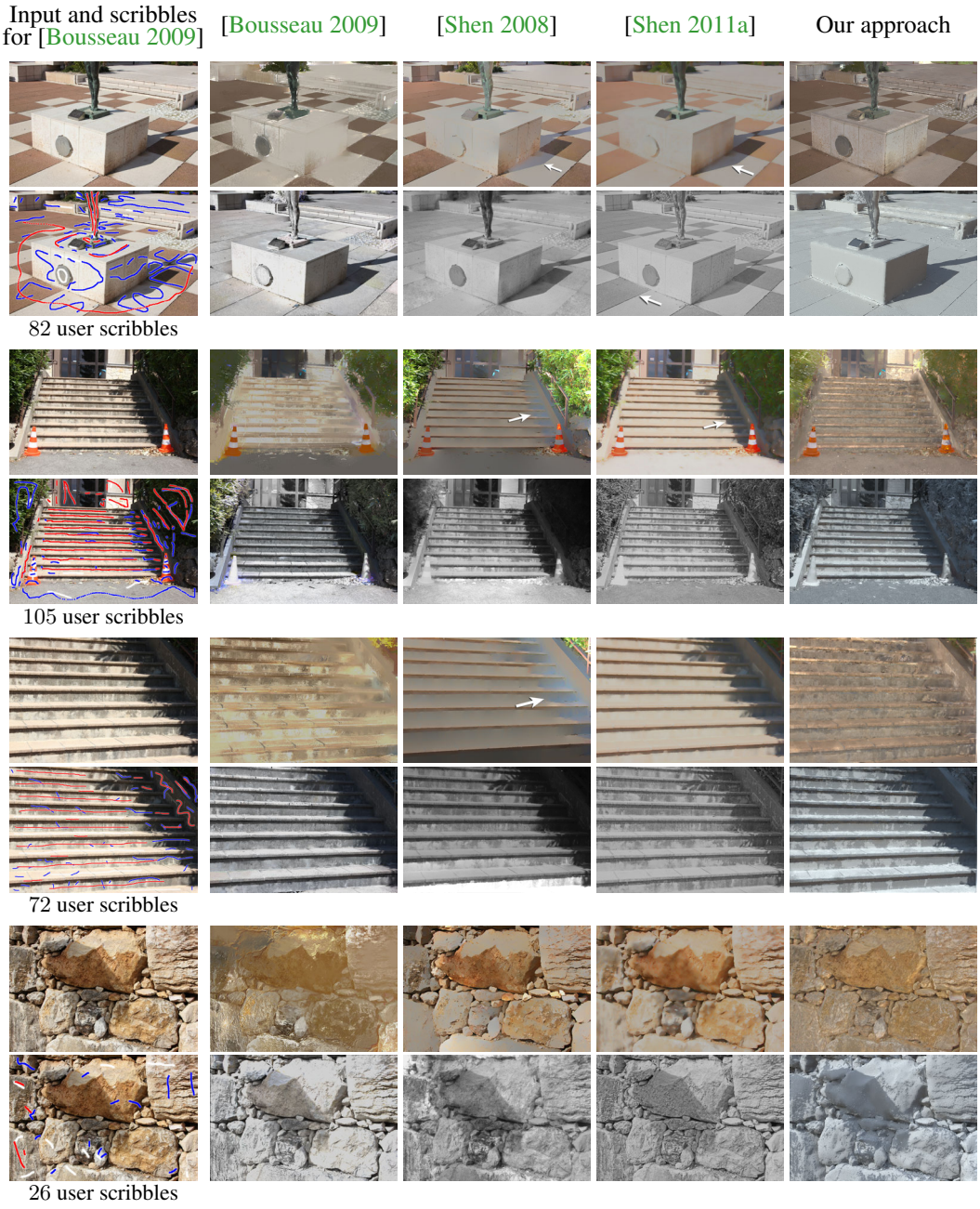


Figure 3.15: *Comparison of our method with a user-assisted method [Bousseau 2009] and two automatic algorithms [Shen 2008, Shen 2011a]. In the first column, the user scribbles used for the method of Bousseau et al. [Bousseau 2009] are shown under the input image. In the next columns, the first row contains the estimated reflectance while the second one corresponds to the total illumination. Our results are shown in the last column. Our multiview approach outperforms single-image automatic algorithms and achieves results of comparable quality to the user-assisted approach with significantly less user intervention. White arrows point to residual reflectance or shading variations. The brightness has been adjusted for comparison.*

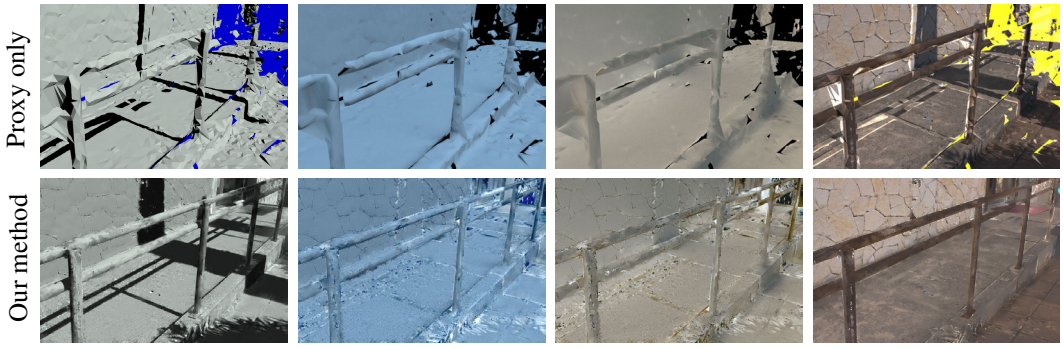


Figure 3.16: *Comparison between the decomposition estimated directly from the geometric proxy (first row), and our results (second row). We obtain the proxy reflectance by dividing the input image by the sum of the illumination components. Holes and inaccuracies in the proxy result in artifacts and residual shadows in the reflectance image.*

In Figure 3.16 we show a comparison to the result obtained simply by using the proxy to compute sun, sky and indirect illumination using PBRT, and then inverting Equation 3.5 to obtain reflectance. This approach produces illumination layers with many holes due to the incomplete proxy. In addition, the sun illumination is completely erroneous (Figure 3.16, top left), due to the lack of reconstruction of the surrounding objects. In contrast, our method (Figure 3.16, bottom row) correctly captures these sun shadows, and removes them in the reflectance layer as well.

3.6.3 Applications

We illustrate applications of our decomposition in Figure 3.1. We first alter the reflectance of the ground to insert a graffiti while maintaining consistent shadows (b). We then add a virtual object in the scene with consistent lighting and shadows (c). We used PBRT to render the dinosaur surrounded by the captured environment map, and its shadow cast on a horizontal plane. Finally we simulate a sunset by changing the color and intensity of each illumination component separately; in addition, our decomposition allows us to blur shadows without affecting the reflectance of the scene (d). All these manipulations can be performed easily in image-editing software with layer support; the accompanying video shows how we created the images in Figure 3.1b-d with Adobe Photoshop.

3.6.4 Discussion

Since we were interested in separating sun lighting from other sources, we have not shown overcast scenes. There is no fundamental reason that our method would not work with overcast scenes; the decomposition would simply rely on the $\mathbf{S}_{\text{sky}}$ and $\mathbf{S}_{\text{ind}}$ values and ignore sun illumination.

Our method tolerates holes in the proxy geometry if the hole is sufficiently distant from the 3D points where sky and indirect illumination are computed. In such configurations, rays that do not hit the proxy will hit the environment map and provide a plausible color. However, in Figure 3.14 (fifth row), the near bush over the stairs was not reconstructed while being too close to be properly captured in the environment map. As a result, rays emitted from the stairs reach the sky instead of the bush and yield a residual shadow in the indirect lighting.

Our method can fail if the initial guess for sun visibility is completely wrong. In Figure 3.14 (bottom row), we show a case where a spurious object appeared in a single view, and was thus not reconstructed at all, resulting in this type of failure. Similarly, if objects have very dark reflectance, or specularities, the PMVS reconstruction procedure does not provide a sufficient number of points, resulting in errors. Similarly to other intrinsic image methods, we have assumed that the scene is Lambertian. Incorporating a specular layer in the intrinsic image formulation is a challenging avenue for future work.

Our optimization for sun visibility exploits the redundant information provided by points of the same reflectance that are in sun light *and* shadow. Nevertheless our method can also handle reflectances that are only in light *or* shadow. Most often in such cases the proxy initialization will result in the correct answer. The only case which could potentially cause errors is when other reflectance curves incorrectly intersect with or influence the curve of this material. We have carefully designed our algorithm to avoid this situation, by making each curve vote only for the region in which it is confident, by processing PMVS points with drastically different orientations separately, and by clustering similar candidate reflectance curves. The results were satisfactory for our examples and the values shown in Table 3.1.

A drawback of the method described in this chapter is the need for the reflective sphere to capture the environment map. Some user interaction is also required for calibration. In Chapter 4, we discuss how to further simplify the capture and calibration process.

3.7 Conclusion

We have presented a method to estimate rich intrinsic images for outdoor scenes. In addition to reflectance, our algorithm generates a separate image for the sun, sky and indirect illumination. Our method relies on a lightweight capture (10-31 photographs in the scenes shown here) to estimate a coarse geometric representation of the scene. This geometric information allows us to estimate illumination terms over a sparse 3D sampling of the scene. We then introduce an optimization algorithm to refine the inaccurate estimations of sun visibility. While incomplete, we demonstrate that this sparse information provides the necessary constraints for an image-guided propagation algorithm that recovers the reflectance and illumination components at each pixel of the input photographs. Our intrinsic image

decomposition allows users of image manipulation tools to perform consistent editing of material and lighting in photographs.

The research work presented in this chapter has led to a technology transfer agreement with Autodesk, and an industrial collaboration which aims to make our intrinsic image decomposition and its applications accessible to customers. As part of this collaboration, we simplify the capture and calibration process of our approach in Chapter 4.

CHAPTER 4

Outdoor lighting extraction from a few photographs

The appearance of a scene largely depends on the lighting it receives. Pictures taken during a sunset or on an overcast day exhibit drastically different moods. Extracting the lighting incident to a scene is a required step towards enabling relighting a photograph, or plausibly inserting synthetic objects.

In Chapter 3, we presented an approach to separate *the effects of lighting and materials* on the appearance in a scene. In this chapter[1], we instead focus on extracting *lighting incident to an outdoor scene* (i.e., incident radiance from all directions) semi-automatically from the captured photographs. Our approach requires only two user clicks and has little overhead in the capture process. We leverage the multiple views of the scene captured at a single time of day and automatically reconstructed geometry, in order to approximate the three components of outdoor lighting:

- **directional sunlight**: we model the sun as a directional light source which can produce high-frequency, sharp cast shadows; we estimate its direction automatically by combining cues from the reconstructed geometry and captured photographs, and we design a method to estimate its radiance from simple user indications (two clicks) instead of using a photographer's grey card as in Chapter 3;

- **low-frequency distant radiance**: we automatically reconstruct an environment map which approximates distant radiance coming from the sky and distant, non-reconstructed surfaces, by extrapolating the information captured in the input photographs; this replaces the environment map which was assembled from HDR pictures of a reflective sphere and manually aligned with the reconstructed scene in Chapter 3;

- **near-field indirect lighting**: we use proxy geometry textured with radiance values from the input photographs in order to approximate indirect illumination, as in Section 3.3.

[1] The method described in this chapter results from joint work with Jorge Lopez-Moreno at REVES / INRIA Sophia-Antipolis, who particularly contributed to Section 4.3 on estimating the sun direction.

This results in a comprehensive method for estimating all the components of outdoor lighting. The estimated incident lighting can then be used as input to the rich intrinsic decomposition described in Chapter 3, which required user intervention to capture an environment map with a chrome sphere, to align the sun and sky dome with the reconstructed geometry, and to take photographs of a gray card to calibrate the illuminants. By simplifying the capture and calibration steps, we remove the most constraining aspects of our decomposition method in order to make it accessible to casual photographers, in the context of a technology transfer agreement with Autodesk.

4.1 Previous work

Several methods have been developed to estimate components of incident lighting in a scene. We review these approaches here.

A light probe is a calibration object of known size and shape, with known reflection properties; placing such a light probe in a scene during capture allows the extraction of some information about incident lighting. In particular, Debevec et al. [Debevec 1998] accurately measure the radiance incident to a physical point in the scene by capturing high-dynamic range images of a mirrored sphere. This results in an environment map which represents incoming radiance from all directions, and can be used to render synthetic objects inserted in the scene. The process of measuring incident radiance with a light probe is now part of the standard workflow in visual effects, animation and games industry, but requires special equipment (the mirrored sphere) and careful capture.

A few approaches simplify the capture process by extrapolating the data contained in captured photographs. In the work of Yu and Malik [Yu 1998], a set of photographs of the horizon are taken, and a sky model is fitted to the sky pixels observed in the captured images. Although they do not need a chrome sphere, they still use special equipment in the form of neutral lens filters in order to measure sun radiance, and assume a geometric model of the scene is available. Lalonde et al. also fit a sky model in order to generate a plausible environment map for each frame in a webcam sequence [Lalonde 2009], or from a single image after estimating the sun position [Lalonde 2011]. In contrast, our method leverages multiple views of the scene which can be quickly captured at a single time of day.

Other approaches assume that the scene is illuminated by a limited number of light sources, and try to recover their position. A local analysis of the surface and image derivatives is used to estimate the direction of a single light source in [Pentland 1982, Brooks 1985]. Alternatively, occluding contours of a single object [Horn 1986, Nillius 2001] or the texturing [Koenderink 2003, Varma 2004] provide cues about where the light is coming from. Lalonde et al. [Lalonde 2011] combine a set of visual cues (shadows over planar surfaces, sky gradients, etc.) in order to estimate the position of the sun in outdoor scenes. Lopez-Moreno et al. [Lopez-Moreno 2010] detect multiple light sources in the scene: they estimate the number of sources, their dominant directions and the rel-

ative intensities based on a perceptual framework. In contrast to these methods, we aim to recover all components of outdoor lighting: this includes a directional component (the sun), but also significant contribution from the sky and indirect lighting in all directions.

Lastly, knowledge of the scene geometry can help with recovering information about the incident lighting. Haber et al. [Haber 2009] estimate distant lighting in photographs of a photo collection and model it with spherical harmonics; however, they assume the reconstructed geometry is complete and includes all surfaces which cast shadows on the scene. Panagopoulos et al. [Panagopoulos 2009] use a probabilistic mixture framework and coarse 3D models provided by the user to estimate and identify shadow casters and light direction in a single image. Our method leverages automatically reconstructed geometry, and handles sparse and inaccurate reconstructions.

In this chapter, we aim to estimate the complete lighting incident to an outdoor scene only from a few photographs and minimal user interaction. We develop a comprehensive approach which estimates all components of outdoor lighting: the direction of the sun and its color, the radiance from the sky and distant surfaces towards all incoming directions, and near-field indirect lighting which represents interreflections from nearby surfaces. Our approach leverages multiple views and automatically reconstructed 3D geometry. It does not require special equipment for capture, and uses photographs taken at a single time of day for low capture overhead.

4.2 Overview

We represent the lighting incident to an outdoor scene with three components: a directional light source, the sun, defined by its direction and radiance; near-field indirect illumination, generated by a proxy textured with radiance values; and an environment map representing the sky dome and distant indirect lighting, which corresponds to the light reflected by non-reconstructed surfaces.

Instead of requesting user input to specify the sun direction as in Chapter 3, we propose in Section 4.3 a method to estimate it automatically by combining cues from reconstructed geometry and input images. First, we find a rough estimation of the sun direction based on the observation that the sun is likely to be in the direction of the brightest normals. Then, we refine the estimated sun direction by automatically aligning a virtual shadow cast by the reconstructed geometry with shadows detected in the input image. We define a set of descriptors which measure the overlap of the two shadow masks and the alignment of the shadow edges.

In Section 4.4, we construct a synthetic environment map which approximates the distant radiance from all incoming directions. Given a few photographs of the scene from different viewpoints, we detect sky pixels and fit a parametric sky model using the estimated sun direction. Our method extrapolates the information present in the input images,

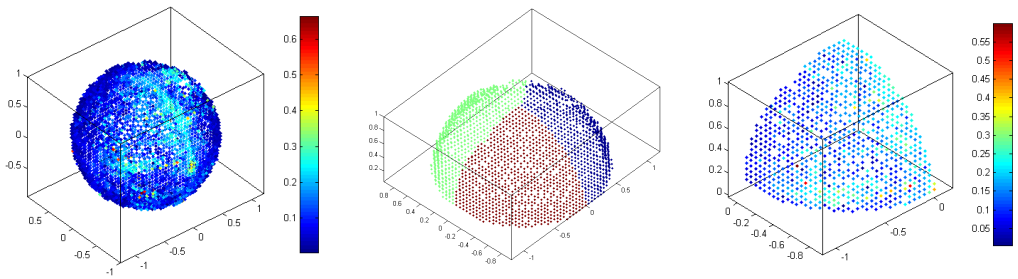
and yields a complete environment map which does not contain holes and does not require a reflective sphere.

The synthetic environment map is consistent with the radiance values captured in the input photographs, which allows us to simplify the illuminant calibration process. Instead of using two photographs of a grey card to correct for the chrome sphere transfer function and estimate the sun radiance, we design a user-assisted method which only requires simple indications (e.g., in lit and shadowed regions). We ask users to specify two pixels with the same reflectance but different illuminations, which allows for the estimation of sun radiance. This process is described in Section 4.5.

4.3 Estimating sun direction

In this section, we estimate the sun direction automatically by combining cues from the reconstructed geometry (i.e., an oriented 3D point cloud and a very approximate mesh), and from one photograph in which a cast shadow is clearly visible. The sun position is defined by its azimuth (ϕ_s) and zenith (θ_s) angles. The proposed method consists of two parts: a coarse estimation based on the orientation of 3D points and their luminance, and a refinement step based on matching shadows detected in one image with those generated by the reconstructed geometry.

4.3.1 Coarse estimation based on luminance



(a) Average luminance per orientation bin

(b) Clusters after K-means

(c) Cluster with highest luminance

Figure 4.1: *(a) Normals from the reconstructed 3D points after binning, and their luminance values (color-mapped); note that normals pointing down will be discarded. (b) Clusters corresponding to three energy levels, after K-means. (c) Cluster with the highest average luminance. The color shows the level of luminance per orientation. Although no particular direction stands out within this cluster, a weighted average of all its orientation bins yields a coarse estimate of the sun direction.*

In the first part of the algorithm, we estimate a coarse sun direction, based on the luminance of reconstructed 3D points and their normals.

We first gather the outgoing radiance of reconstructed 3D points, by averaging the color of corresponding pixels in all images where they are visible. We then compute their

luminance and use it as a coarse approximation of their irradiance; this approximation assumes that the reflectance is constant, but we found that it is sufficient to obtain a coarse sun direction when many points and orientations are available. We bin the normals and assign the average luminance to each bin (Figure 4.1a).

We cluster the orientation bins with a K-means approach depending on their luminance level and angular distance, similar in spirit to [Lopez-Moreno 2010]. We build feature vectors which combine average luminance, azimuth angle, and zenith angle, for each orientation bin. As we are looking for the sun direction, we discard all the normals pointing down. We then apply K-means (with $K = 3$) to identify three clusters which intuitively corresponds to three energy levels (direct light, shadows, and uncertain), as illustrated in Figure 4.1b.

We compute the weighted average of the normals in the cluster with the highest energy level (Figure 4.1c), using the per-bin luminance as a weighting factor. The resulting normal is likely to point towards the sun, and will be used as the initial guess $(\phi_s^{\text{init}}, \theta_s^{\text{init}})$ for the second part of our method. Although we have introduced an uncertainty in this initial solution by considering the reflectance as uniform, in our experiments this approach was sufficient to find at least the right quadrant $(\pm\pi/4)$ of the solution, providing sufficient initialization for the subsequent shadow-based refinement.

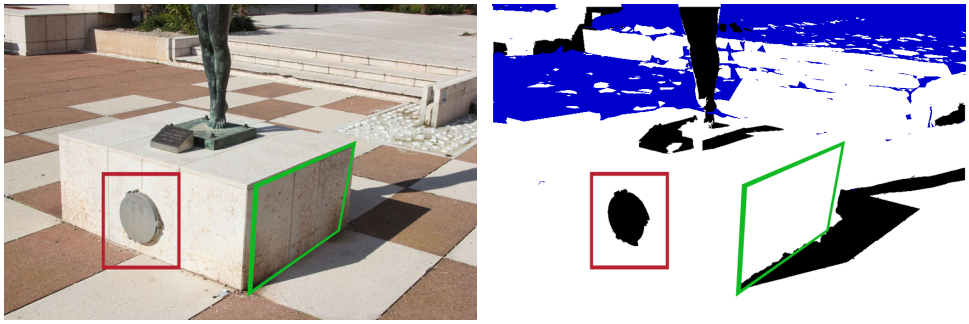
4.3.2 Shadow-based refinement

Once we have an initial guess, we apply an optimization approach to refine the estimation of sun direction. We use an image from the dataset where a cast shadow is clearly visible; we assume that the geometry of both the shadow caster and the receiving surface is approximately reconstructed in the proxy geometry.

We estimate the shadow areas in the selected image by using the automatic pairing technique described in [Guo 2011]. This yields a *detected shadow mask* s_{detected}. This estimate is not fully reliable and false positives and negatives are to be expected: regions with dark reflectance might be wrongly considered as shadows, while vertical surfaces receiving significant indirect lighting might not be detected as shadows (Figure 4.2). After the shadow detection, we mask out all the pixels which correspond to non-reconstructed regions in the proxy; these pixels will be ignored in our optimization.

For each sun position, we generate virtual shadows using the reconstructed geometry, as in Section 3.2.3 (Figure 3.5), yielding a *virtual shadow mask* s_{virtual}. Our strategy then consists in automatically comparing this virtual shadow mask, for a set of azimuth and zenith angles around the initial guess, with the detected shadow mask s_{detected} estimated with [Guo 2011].

We define a set of descriptors which measure how the two masks agree, based on area of shadow overlap and orientation of shadow edges.



Input image Result of the shadow detector

Figure 4.2: *Example of common errors in shadow detection. Framed in red: region of dark reflectance, detected as shadow. Framed in green: shadow not detected due to indirect light. After the image-based shadow detection, pixels corresponding to holes in the proxy are masked out (blue pixels).*

4.3.2.1 Shadow overlap area

The first descriptor compares the virtual shadow mask s_{virtual}, created by casting shadows and the detected shadow mask s_{detected} pixel-per-pixel. This descriptor measures the percentage of pixels which do not agree well:

$$D_{\text{overlap}} = \sum_{p \in \mathcal{P}} \frac{(s_{\text{virtual}}(p) - s_{\text{detected}}(p))^2}{|\mathcal{P}|} \tag{4.1}$$

where $\mathcal{P}$ is the set of pixels corresponding to regions with reconstructed geometry.

This descriptor reaches a local minimum at the correct solution (see Figure 4.4, left). However this minimum is non-global, as both the proxy shadow and the detected shadow may miss shadowed areas (due to incomplete reconstructed geometry, or to poor detection of shadows in the image) or contain false positives (due to noise in the proxy, or surfaces with dark reflectance).

4.3.2.2 Orientation of shadow edges

We design a second descriptor which encourages shadow edges to have a similar orientation in the virtual and detected shadow masks.

We first extract the contours of shadows in the virtual and detected shadow masks using Canny's filter [Canny 1986]. Then, we establish pairs of pixels along the edges of shadows which share a similar orientation within a given distance K in screen space. For each pixel (x, y) on a contour of the detected shadow mask, we compute a distance function to each pixel (x', y') in the contours of the virtual shadow mask. This distance incorporates the 2D position of the two points, and the local orientation of the contours at both points. We

define this distance function d_{matching} as follows:

$$d_{\text{matching}} = \sqrt{(x-x')^2 + (y-y')^2 + \left(\frac{2K}{\pi}(\psi - \psi')\right)^2} \tag{4.2}$$

where ψ and ψ' are the orientation (angle in $[0, 2\pi]$) of the detected and virtual shadow's edge pixels in screen plane respectively, computed as in [Hong 1998]. The difference $(\psi - \psi')$ is computed with the interior angle. The $(2K/\pi)$ factor balances the penalties due to orientation changes and Euclidean distance, so that points with an orientation difference above $\pi/2$ (right angle) yield a distance $d_{\text{matching}} \geq K$.

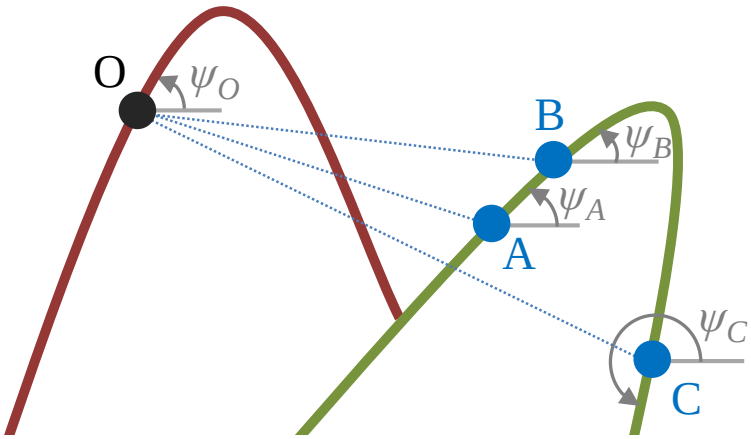


Figure 4.3: *Illustration of matching pixels on shadow edges. For each pixel along the contours of the detected shadow mask (in this example, point O on the red curve), we aim to locate the closest pixel on a contour of the virtual shadow mask (green curve), based on 2D distance and edge orientation. Out of the three pixels A, B, C in this example, the most likely matching will be A because it is closest both in 2D space and orientation.*

For each pixel (x, y) on a contour of the detected shadow mask, we keep the matching pixel (x', y') with the lowest distance d_{matching}; this is illustrated for one pixel in Figure 4.3. This results in a set of pairs of matching pixels which are close and have a similar local shadow edge orientation.

A large number of matchings with a small distance d_{matching} indicate that edges of the virtual and detected shadow masks are well aligned. From this observation, we adopt a simple multiscale analysis to increase robustness by combining two search radii K and $K/3$, and we assemble a descriptor from the following data:

- $nbPairs$: number of pairs matched with distance below search radius K,
- $averageDistance$: average distance of these pairs,
- $nbPairsSmallRadius$: number of pairs matched with distance below a smaller search radius $K/3$,
- $averageDistanceSmallRadius$: average distance of these pairs.

We combine these two average distances, using the number of pairs as a confidence measure: the higher the number of matching pairs, the more confident we are that a low average is not produced by chance. The smaller search radius $K/3$ is more restrictive and yields a better alignment of shadows near the solution, whereas radius K yields more matching pairs and is more robust to imperfect detected and virtual shadows. We also normalize all these values to the $[0, 1]$ range by dividing by the maximum number of pairs $maxNbPairs$ and the maximum average value $maxAverageDistance$ detected in the whole search space. This yields the following descriptor:

$$D_{\text{orient}} = \frac{nbPairs}{maxNbPairs} * \left(1 - \frac{averageDistance}{maxAverageDistance}\right) + \frac{nbPairsSmallRadius}{maxNbPairsSmallRadius} * \left(1 - \frac{averageDistanceSmallRadius}{maxAverageDistanceSmallRadius}\right) \tag{4.3}$$

4.3.2.3 Optimization

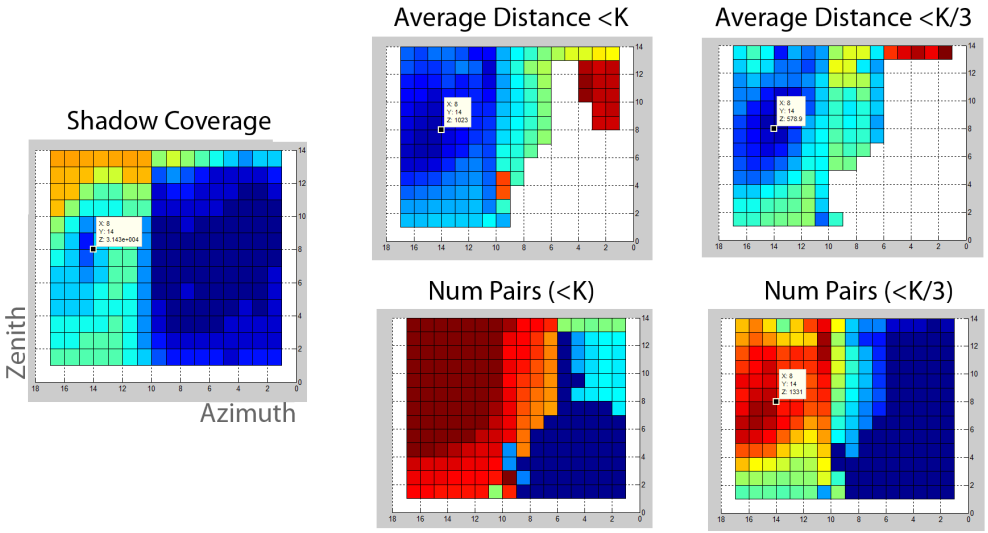


Figure 4.4: *Example of descriptor responses in azimuth-zenith space. Responses are color-coded (blue: low values; red: large values). The highlighted location corresponds to the global maximum of the combined descriptors (Equation 4.4).*

Figure 4.4 shows the responses of individual descriptors for different zenith and azimuth angles. In order to find the sun direction, we look for the angles which maximize a function that combines the two descriptors:

$$\arg\max_{\theta_s,\phi_s} \left(D_{\text{orient}} * (1 - D_{\text{overlap}}) + F(\theta_s)\right) \tag{4.4}$$

where $F(\theta_s)$ is a regularization term which penalizes extreme sun positions close to the ground level ($\theta_s = \pi/2$), since any small protuberance or noise in the proxy mesh produces

elongated shadows which might yield false positives.

We follow a brute force optimization approach around the initial guess obtained from Section 4.3.1. We limit our search space to one quarter $(\phi_s^{\text{init}} \pm \pi/4, \theta_s^{\text{init}} \pm \pi/4)$. Additionally, the sun zenith angle is bound to the $[0, \pi/2]$ range, as we are looking for a sun position above the ground level. We take a fixed step size of $\pi/32$.

4.3.3 Results

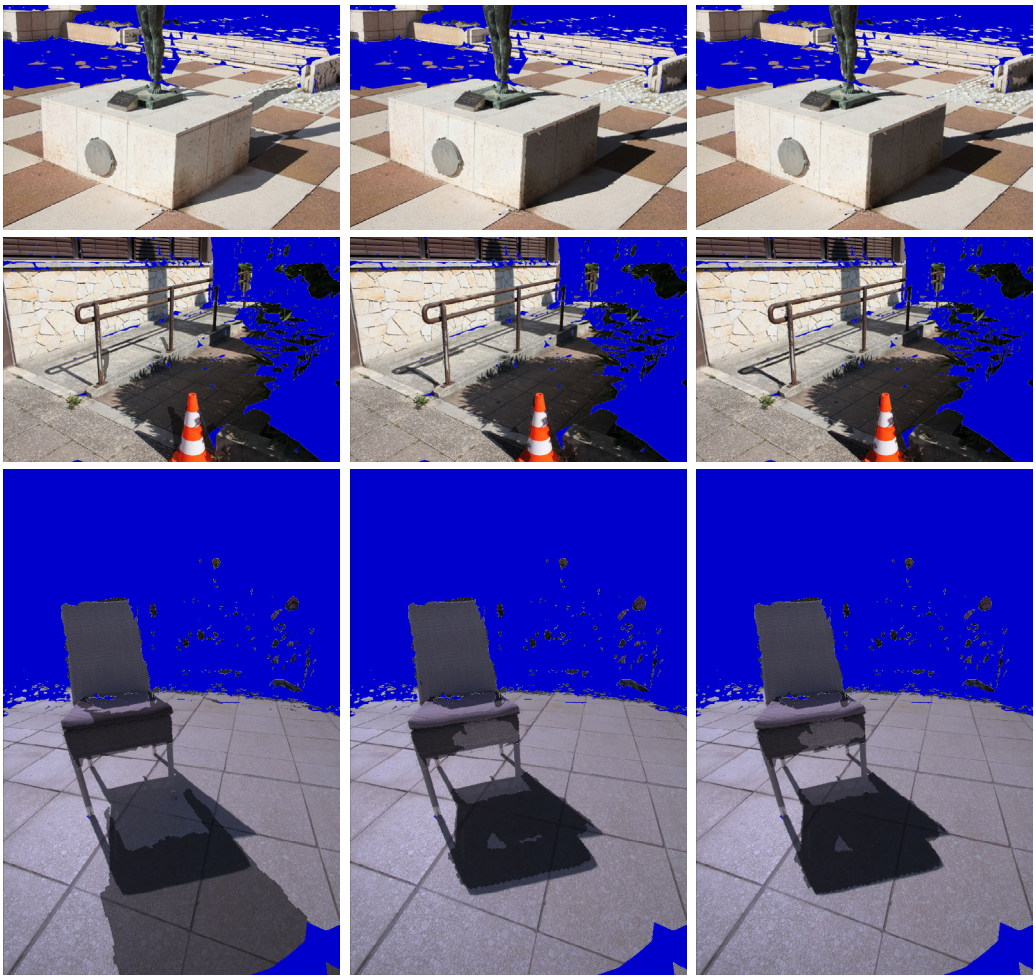


Figure 4.5: *Results of the automatic estimation of sun direction. As in Figure 3.5, we show the virtual shadow cast by the reconstructed proxy overlaid with the input photograph. The ground truth sun direction has been set manually using the method described in Section 3.2.3.*

Figure 4.5 shows the results of sun direction estimation, after coarse estimation and shadow-based refinement, on three scenes. The final error compared to ground truth is less than 2.5 degrees on all scenes, as reported in Table 4.1.

	Coarse estimation	Shadow-based refinement
Statue	24.5	<0.1
Ramp	29.9	2.4
Chair	14.3	1.2

Table 4.1: *Error in sun direction estimation, in degrees.*

4.4 Gathering distant illumination

In Chapter 3, we captured an environment map which represents the radiance incoming from all directions. This is necessary to account for *distant* indirect illumination, which comes from directions corresponding to holes in the reconstructed geometry, and for sky illumination. The capture process involved taking photographs of a chrome sphere placed near the center of the scene, from two viewpoints and with multiple exposures to build an HDR environment map; the user then marked sky and non-sky pixels in the environment map. In this section, we show that this manual process can be avoided by fitting a parametric sky model to the sky pixels observed in the input photographs.

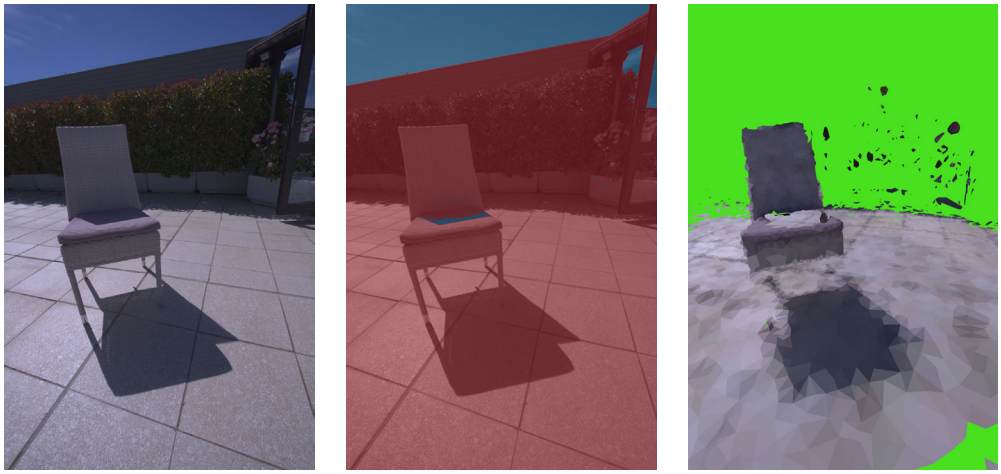
4.4.1 Extracting partial environment maps from input photographs

From each input photograph which contains a portion of sky, we extract two *partial environment maps* which represent distant radiance. We consider that all pixels corresponding to non-reconstructed regions in the proxy geometry represent "distant radiance".

Given one input image (Figure 4.6a), we first detect sky pixels. We currently use a simple thresholding strategy which gives reasonable results on our clear-sky scenes (Figure 4.6b, blue pixels), but other sky detectors such as the approach proposed by Hoiem et al. [Hoiem 2005] could be used to handle more complex cases.

We then identify pixels corresponding to distant radiance (Figure 4.6c, green pixels) by rendering the reconstructed proxy geometry from the viewpoint of the input photograph, using ray-tracing. Pixels corresponding to rays which intersect the proxy are ignored, since the intersected geometry will contribute to near-field indirect illumination (see Section 3.3).

We use the color of the pixels corresponding to distant radiance in the input photograph to partially fill two environment maps aligned with the reconstructed scene: Figure 4.6d represents a partial sky environment map extracted from the photograph, and Figure 4.6e corresponds to the partial distant indirect environment map. The sky/non-sky pixel classification (Figure 4.6b) is used to decide which environment map each pixel contributes to: sky pixels corresponding to distant radiance are transferred into the partial sky environment map, while non-sky pixels corresponding to distant radiance contribute to the partial distant indirect environment map.



(a) Input image (b) Detected sky pixels (c) Reconstructed proxy

(d) Partial sky environment map (e) Partial distant indirect environment map

Figure 4.6: *From one input image (a), sky pixels are detected (b, blue pixels) and the reconstructed geometry is rendered from the viewpoint of the photograph to label pixels corresponding to distant radiance (c, green pixels). These pixels are then used to construct two partial environment maps, in latitude-longitude format, which represent the incoming radiance due to skylight and distant indirect lighting.*

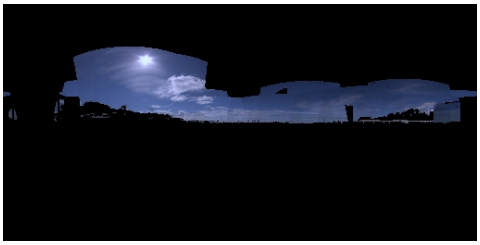
4.4.2 Fitting a parametric sky model

Although each input photograph only observes a small portion of the sky (or no sky at all) due to its limited field of view, combining the partial environment maps extracted from several input images leads to *average environment maps* which describe a larger portion of distant radiance (Figure 4.7a-b).

In order to fill the remaining holes, we fit a parametric sky model to the non-saturated pixels in the average sky environment map (Figure 4.7a). We use the physically-inspired sky model introduced by Perez et al. [Perez 1993], which defines the luminance L_p of a sky element relative to another arbitrary sky element, such as zenith luminance L_z:

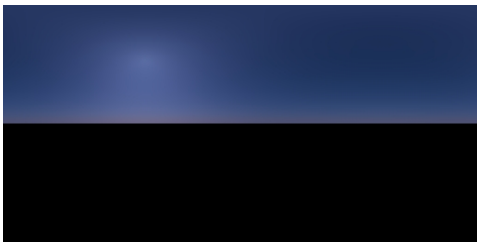
$$L_p = f(\theta_p, \gamma_p)\,\frac{L_z}{f(0, \theta_s)} \tag{4.5}$$

where θ_p is the zenith angle of the sky element, γ_p its angle with respect to the sun, θ_s the sun zenith angle, and $f(\ )$ a function which depends on five additional coefficients representing the atmospheric conditions. Preetham et al. [Preetham 1999] show that these



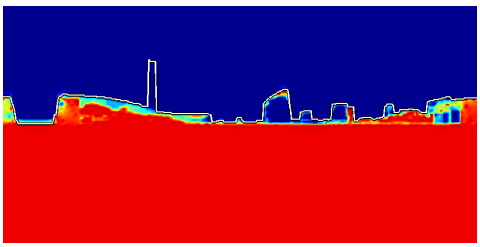
(a) Average sky environment map

(b) Average distant indirect environment map



(c) Parametric sky model after fitting

(d) Inpainted distant indirect environment map



(e) Blending weights for sky and indirect radiance

(f) Final environment map

Figure 4.7: *Combining partial environment maps constructed from several photographs yields average environment maps which cover a larger set of incoming light directions (a-b). We use the average sky environment map to fit a parametric sky model (c), and fill holes in the average distant indirect environment map with diffusion (d). We then estimate per-pixel blending weights β (e, color-coded) which describes the probability that radiance comes from the sky (blue pixels) or distant indirect lighting (red pixels); the white curve defines the limit above which all pixels correspond to sky radiance (see text for details). The environment map resulting from our approach approximates distant radiance coming from all directions (f).*

five weather coefficients can be approximated with a single parameter, the turbidity t, and they apply the sky model in xyY color space.

Since the sun zenith angle is known from Section 4.3, we can deduce θ_p and γ_p for each sky element. Fitting the parametric sky model then consists in estimating four parameters which minimize the difference between the average sky environment map and the synthesized sky:

$$\underset{t,\mathbf{k}^x,\mathbf{k}^y,\mathbf{k}^Y}{\arg\min} \sum_{c\in\{x,y,Y\}} \sum_{p\in\mathcal{P}} \left(\mathbf{k}^c\, f(\theta_p,\gamma_p,t) - \mathbf{A}_p^c\right)^2 \tag{4.6}$$

where $\mathcal{P}$ is the set of non-empty pixels in the average sky environment map $\mathbf{A}$ (converted to xyY color space), and $\mathbf{k}$ is an unknown scale factor which incorporates the sky zenith color and the denominator in Equation 4.5.

We solve this non-linear minimization problem iteratively using Matlab's `fminsearch` function: at each iteration, we generate a synthetic sky given the current turbidity, then find $\mathbf{k}$ by solving a linear system. The initial turbidity is set to $t = 3.5$. After the fitting, we generate a sky environment map using the estimated parameters (Figure 4.7c).

4.4.3 Assembling the final environment map

Similarly, we extrapolate the average distant indirect environment map (Figure 4.7b) by applying diffusion [Perona 1990] to fill-in holes below the horizon (Figure 4.7d). We combine this extrapolated environment map with the synthesized sky after fitting, using blending with per-pixel weights, which we describe next.

Some directions of incident radiance appear both in partial sky environment maps and partial distant indirect environment maps; this occurs when 3D reconstruction fails to recover the geometry of some nearby surfaces. In such cases, we account for both sky and indirect radiance at the corresponding pixels in the final environment map, by mixing their contributions. More formally, if a particular radiance direction appears in n_{sky} partial sky environment maps with an average color $\mathbf{L}_{\text{sky}}$, and in n_{ind} partial distant indirect environment maps with an average color $\mathbf{L}_{\text{ind}}$, we set the color in the final environment map as:

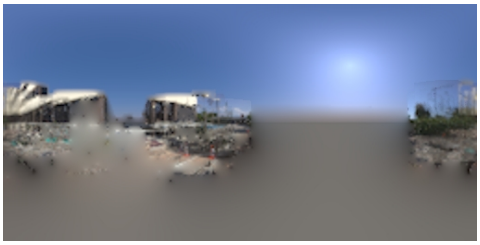
$$\beta\, \mathbf{L}_{\text{ind}} + (1 - \beta)\, \mathbf{L}_{\text{sky}} \tag{4.7}$$

with blending weight $\beta = n_{\text{ind}} \,/\, (n_{\text{ind}} + n_{\text{sky}})$. For directions where both $n_{\text{ind}} = 0$ and $n_{\text{sky}} = 0$, this quantity β is undefined. We instead infer it by applying diffusion from the neighboring pixels where it is defined (Figure 4.7e). We further constrain incident radiance in the lower hemisphere to come from distant indirect lighting (i.e., $\beta = 1$). Lastly, we estimate a curve which separates sky and indirect regions of the environment map, by locating the highest row where $n_{\text{ind}} > 0$, for each column of the environment map in latitude-longitude representation; after smoothing this curve, we enforce $\beta = 0$ at all pixels above this curve (Figure 4.7e, white curve). This process allows automatic estimation of the contribution of sky and indirect radiance for each incident direction, instead of asking the users to manually segment the environment map as in Figure 3.4.

After combining the inpainted distant indirect environment map with the synthesized sky model, we add in the clouds from the average sky environment map by replacing the corresponding pixels in the final environment map. This results in a synthetic environment map constructed from only a few photographs (Figure 4.7f).

We also show a comparison with a captured environment map for the Ramp scene, in Figure 4.8. For this particular scene, the assumption that the environment map captures *distant* radiance does not hold, because some objects (in particular the building façade and the tall bush) are very close from the chrome sphere location; these objects occupy large areas in the captured environment map. In contrast, our synthetic environment map only

(a) Captured environment map



(b) Synthesized environment map

Figure 4.8: *Captured and synthesized environment maps for the Ramp scene. Note that the captured environment map (Section 3.2.1) is shown before color correction for the chrome sphere. The synthesized environment map represents radiance from the sky and from non-reconstructed distant surfaces; it does not include the nearby buildings, which appear in the reconstructed proxy geometry and contribute to near-field indirect lighting.*

contains distant radiance; reconstructed surfaces do not appear in the synthetic environment map because they produce near-field indirect illumination which is modeled by the colored geometric proxy. Lastly, some directions of the radiance are never observed in the input photographs, which results in large holes in the synthetic environment; in particular, the bush is missing. Taking additional pictures to cover these regions at the time of capture would prevent these holes.

4.5 User-assisted illuminant calibration

In Section 3.2.3, we described how two photographs of a grey card can be used to correct for the color transfer function of the chrome sphere used to capture an environment map, and to estimate the sun radiance. We show here how to remove grey card calibration.

Applying the method described in Section 4.4 yields an environment map which contains radiance values coherent with the photographs of the scene, since it uses pixels from the input images to fit a sky model. As a result, there is no need to calibrate the environment map (i.e., $\mathbf{K} = (1, 1, 1)$), and only the sun radiance $\mathbf{L}_{\text{sun}}$ remains to be estimated.

We start from Equation 3.8 and derive the ratios for two points:

$$\frac{\mathbf{I}^{(2)}}{\mathbf{I}^{(1)}} = \frac{\mathbf{R}^{(2)}}{\mathbf{R}^{(1)}} * \frac{(v^{(2)}_{\text{sun}} \cos\theta^{(2)}_{\text{sun}} \mathbf{L}_{\text{sun}} + \mathbf{S}^{(2)}_{\text{sky}} + \mathbf{S}^{(2)}_{\text{ind}})}{(v^{(1)}_{\text{sun}} \cos\theta^{(1)}_{\text{sun}} \mathbf{L}_{\text{sun}} + \mathbf{S}^{(1)}_{\text{sky}} + \mathbf{S}^{(1)}_{\text{ind}})} \tag{4.8}$$

When the two points share the same reflectance, the reflectance ratio $\mathbf{R}^{(2)} / \mathbf{R}^{(1)}$ vanishes and we can rewrite the equation as:

$$\mathbf{L}_{\text{sun}} (\mathbf{I}^{(2)} v^{(1)}_{\text{sun}} \cos\theta^{(1)}_{\text{sun}} - \mathbf{I}^{(1)} v^{(2)}_{\text{sun}} \cos\theta^{(2)}_{\text{sun}}) = \mathbf{I}^{(1)}(\mathbf{S}^{(2)}_{\text{sky}} + \mathbf{S}^{(2)}_{\text{ind}}) - \mathbf{I}^{(2)}(\mathbf{S}^{(1)}_{\text{sky}} + \mathbf{S}^{(1)}_{\text{ind}}) \tag{4.9}$$

where $\mathbf{L}_{\text{sun}}$ is the only unknown. Therefore, a pair of points which share the same reflectance is enough to estimate the sun radiance.

Input image and correspondences for calibration | Reflectance | Total illumination

Figure 4.9: *Preliminary results of the decomposition on the Ramp scene. Top row: results with the captured environment map and user-assisted calibration from Chapter 3. Bottom row: results with the synthesized environment map and 2-click calibration without using the grey card; correspondences marked by the user are shown on the left image.*

In practice, we let users indicate correspondences in one picture of the scene. They are asked to mark pixels in at least two regions which share the same reflectance. The values $v_{\text{sun}}^{(i)}$, $\theta_{\text{sun}}^{(i)}$, $\mathbf{S}_{\text{sky}}^{(i)}$ and $\mathbf{S}_{\text{ind}}^{(i)}$ are estimated using the proxy in the marked regions. When multiple pixels are selected, all the quantities are averaged over each connected component.

4.6 Decomposition results

We have applied the rich intrinsic image decomposition method described in Chapter 3, after automatically extracting the sun direction and the synthetic environment map, and running the user-assisted illuminant calibration. We used RAW images for all the input photographs instead of the HDR pictures as in Chapter 3; this simplifies the capture process and did not negatively affect the results in our experiments.

Figure 4.9 provides a visual comparison between the method from Chapter 3 and the new method on the Ramp scene. Although the shadows are not as well separated from the reflectance layer, the decomposition is fully automatic except for the two clicks to mark the correspondences for calibration (Figure 4.9, bottom-left). Also, the parametric sky model was fitted to a very limited portion of the environment, because the sky was visible in only few pictures. Results should improve with additional pictures of the missing portions of the scene.

We also captured a new scene, without a grey card nor a chrome sphere (Figure 4.10). For this scene, the PMVS reconstruction fails to reconstruct the chair pillow and legs; there are also no reconstructed points in the background, where the bush casts a shadow on the

(a) Input image (b) Reflectance (c) Total illumination (d) Sun illumination at reconstructed points

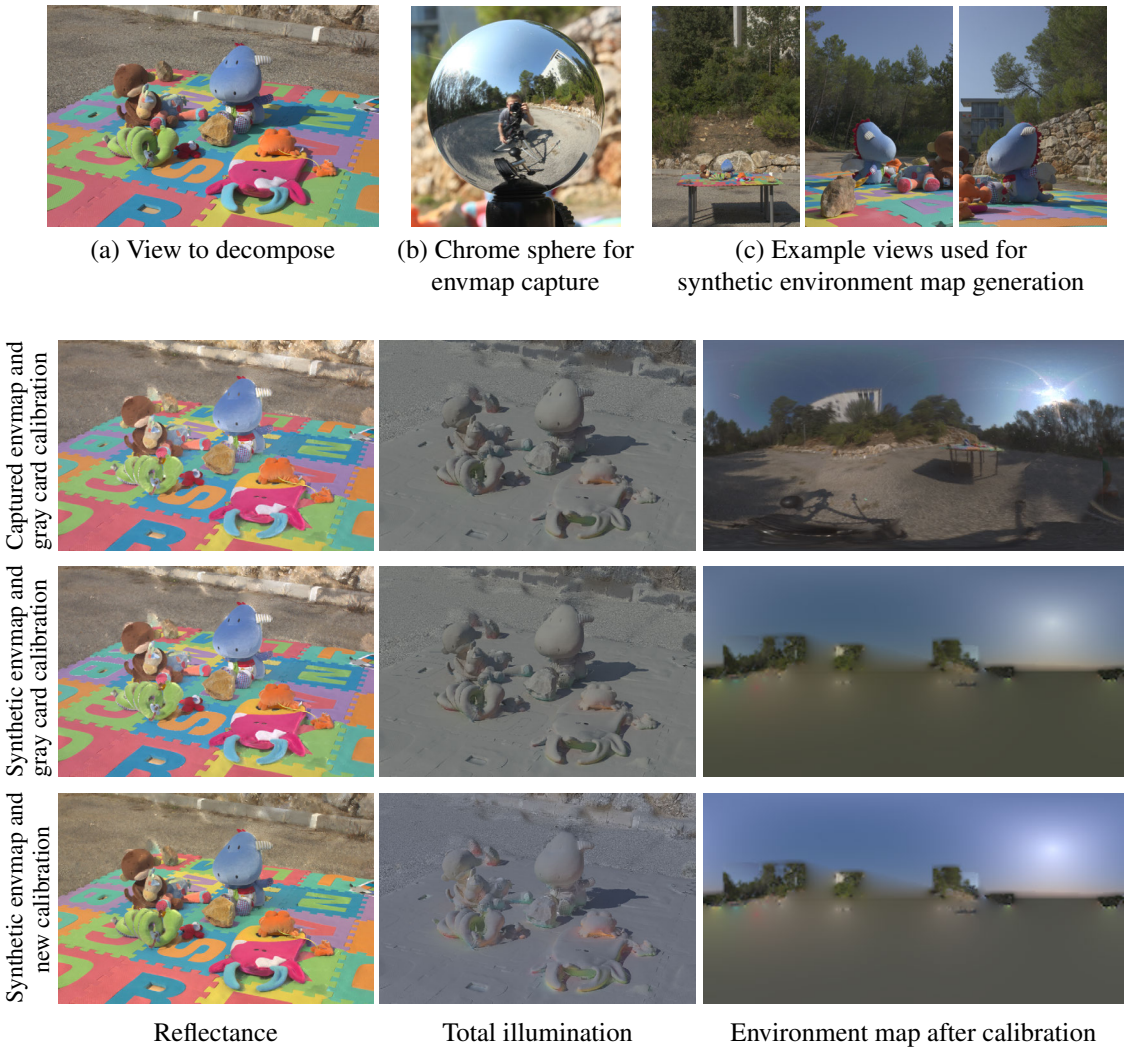
Figure 4.10: *Preliminary results of the decomposition on the Chair scene. Although the chair's shadow is properly detected (d), a color shift remains in the extracted reflectance (b) due to incomplete and inaccurate reconstructed geometry. Note that the background bush and the ground close to it were not reconstructed (d).*

ground. Still, our method reconstructs an environment map from only a few photographs (Figure 4.7d) and properly estimates the sun visibility (shadows) on reconstructed portions of the ground and chair (Figure 4.10d). We believe the results will improve with a better reconstruction, in particular the color shift in the reflectance below the chair.

In order to compare the proposed method with the one described in Chapter 3, we captured a new outdoor scene consisting of toys placed on a table. We captured images of a chrome sphere and a grey card placed in the scene, and took additional pictures of the surroundings in order to synthesize an environment map. Figure 4.11 compares the results obtained when using the environment map captured with the chrome sphere, or that obtained with the method described in Section 4.4. It also compares the two illuminant calibration processes: using a gray card (Section 3.2.3) or correspondences marked by the user (Section 4.5). Although results are visually similar, extracting an environment map directly from the input images and using pairs of points marked by the user simplify the capture process, since it does not require placing a chrome sphere or a grey card in the scene.

We also show results of our rich intrinsic decomposition for three views of this new scene, in Figure 4.12. Note that the cameras were registered and the geometry was reconstructed using the Autodesk reconstruction pipeline; the chrome sphere and grey target were not used in these results.

Lastly, the faster and lighter capture process allows us to treat scenes that were difficult for the method described in Chapter 3, such as urban scenes. We show in Figure 4.13 results of our rich intrinsic image decomposition on a complex scene, captured within a few minutes in a busy city. This suggests promising directions for applications in urban environments.



(a) View to decompose (b) Chrome sphere for envmap capture (c) Example views used for synthetic environment map generation

Figure 4.11: *Results of our decomposition on an image of the OutdoorToys scene (a). We reconstruct an environment using either HDR photographs of a chrome sphere (b), or by leveraging the information contained in photographs of the surroundings (c) as described in Section 4.4. We calibrate the illuminants using either a grey card (top and middle row) as described in Section 3.2.3, or the user-assisted calibration (bottom row) described in Section 4.5. Results of all three methods are visually similar, but the generation of a synthetic environment map and user-assisted calibration simplify the capture since it does not require the placement of a chrome sphere or a grey card in the scene.*

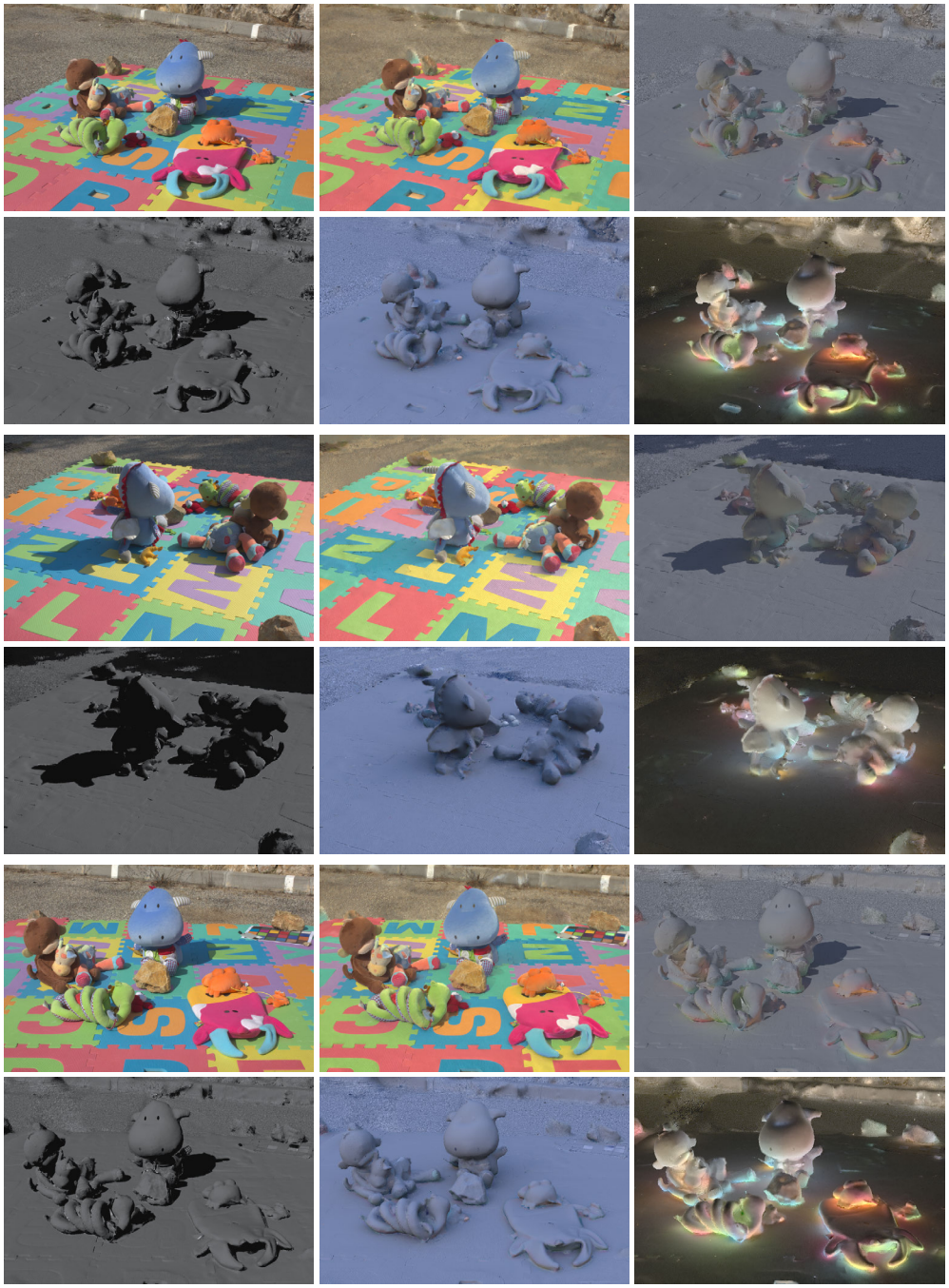


Figure 4.12: *Results of our rich intrinsic decomposition on three views of the OutdoorToys scene. We created a synthetic environment map using the method described in Section 4.4, and calibrated the sun radiance using our new method,i.e., without using a grey card (Section 4.5). We adjusted the brightness of images for illustration purposes (the scaling values were:* 1 *for the input, sky illumination and indirect illumination;* 0.2 *for the total illumination and sun illumination).*

Figure 4.13: *Results of our rich intrinsic decomposition on three views of a complex scene: the Masséna Museum, in Nice. The capture process required neither a photographer's grey card, nor a chrome sphere.*

4.7 Conclusion and future work

We have presented a method for simplifying the capture and calibration process of our rich intrinsic image decomposition. Our preliminary results show that the sun direction can be reliably estimated from cast shadows, that an approximate environment map can be constructed from just a few photographs, and that the calibration based on a grey card can be replaced by simple user intervention consisting of two clicks. We have developed a pipeline which can handle the reconstruction data provided by Autodesk.

In future work, we would like to fully automate the calibration process: instead of requesting users to specify regions of constant reflectance, we will investigate detecting paired regions with an approach inspired by [Guo 2011], constrained by our geometry-based estimation of the low-frequency components of the illumination (Chapter 5). Such

automatic calibration will make our approach practical for processing a large number of scenes.

The outdoor lighting recovered with our approach could be used to realistically insert synthetic objects in the captured scene, with plausible illumination, even for glossy objects. The simplified capture and calibration process enables easy extraction of lighting in photographs, which can be used to recover illumination-free 3D models of captured scenes. Combined with free-viewpoint image-based rendering [Chaurasia 2011], it opens the way to navigating in 3D and controlling the lighting in outdoor scenes captured at a single time of day.

CHAPTER 5

Coherent Intrinsic Images from Photo Collections

Photo-sharing websites such as Flickr© and Picasa© contain millions of photographs of famous landmarks captured under different viewpoints and illumination conditions. Photo collections of less famous places are also becoming available thanks to initiatives like the collaborative game PhotoCity [Tuite 2011]. The wide availability of photos on the internet has been exploited for many computer graphics applications including scene completion [Hays 2007] and virtual tourism [Snavely 2006]; however, the variation of illumination in a collection has often been seen as a nuisance that is distracting during navigation or, at best, an interesting source of visual diversity. Inspired by existing work on time-lapse sequences [Weiss 2001, Matsushita 2004a], we consider these variations as a rich source of information to compute intrinsic images.

In this chapter[1], we exploit the rich information provided by multiple viewpoints and illuminations in an image collection to process complex scenes without user assistance, nor precise and complete geometry. Furthermore, we enforce that the decomposition be *coherent*, which means that the reflectance of a scene point should be the same in all images.

We process pairs of points visible in several images in order to guide the decomposition process. We consider the ratio of radiance between two points, which is equal to the ratio of reflectance if the points share the same illumination. A contribution of this chapter is to identify pairs of points that are likely to have similar illumination across most conditions. For this, we leverage sparse 3D information from multi-view stereo as well as a simple statistical criterion on the distribution of the observed ratios. These ratios give us a set of equations relating the reflectance of pairs of sparse scene points, and consequently of sparse pixels where the scene points project in the input images. To infer the reflectance and illumination for all the pixels, we build on image-guided propagation [Levin 2008, Bousseau 2009]. We augment it with a term to force the estimated reflectance of a given 3D point to be the same in all the images in which it is visible. This yields a large sparse linear system, which we solve in an interleaved manner. By enforcing coherence in the reflectance layer we obtain a common "reflectance space" for all input views, while we extract the color variations proper to each image in the illumination layer.

[1]The work described in this chapter has been accepted to SIGGRAPH Asia 2012 [Laffont 2012b]. It has been developed in part during a research visit at MIT, with Frédo Durand and Sylvain Paris (Adobe).

Our automatic estimation of coherent intrinsic image decompositions from photo collections relies on the following contributions:

- a method to robustly identify reliable reflectance constraints between pairs of pixels, based on multi-view stereo and a statistical criterion;
- an optimization approach which uses the constraints within and across images to perform an intrinsic image decomposition with *coherent* reflectance in all views of a scene.

We ran our method on 9 different scenes, including a synthetic benchmark with ground truth values, which allows for a comparison to several previous methods. We use our intrinsic images for three different applications, including a novel image-based illumination transfer between photographs captured from different viewpoints. Our coherent reflectance layers enable stable transitions between views by applying a single illumination condition to all images.

5.1 Overview

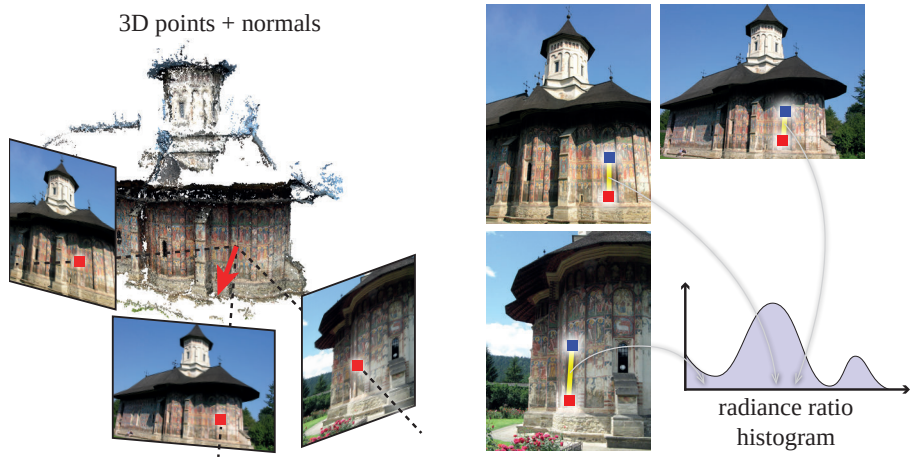
We take as input a collection of photographs $\{\mathbf{I}_i\}$ of a given scene captured from different viewpoints and under varying illumination. We seek to decompose each input image into an illumination layer $\mathbf{S}_i$ and a reflectance layer $\mathbf{R}_i$ so that, for each pixel p and each color channel c:

$$\mathbf{I}_{ic}(p) = \mathbf{S}_{ic}(p)\,\mathbf{R}_{ic}(p) \tag{5.1}$$

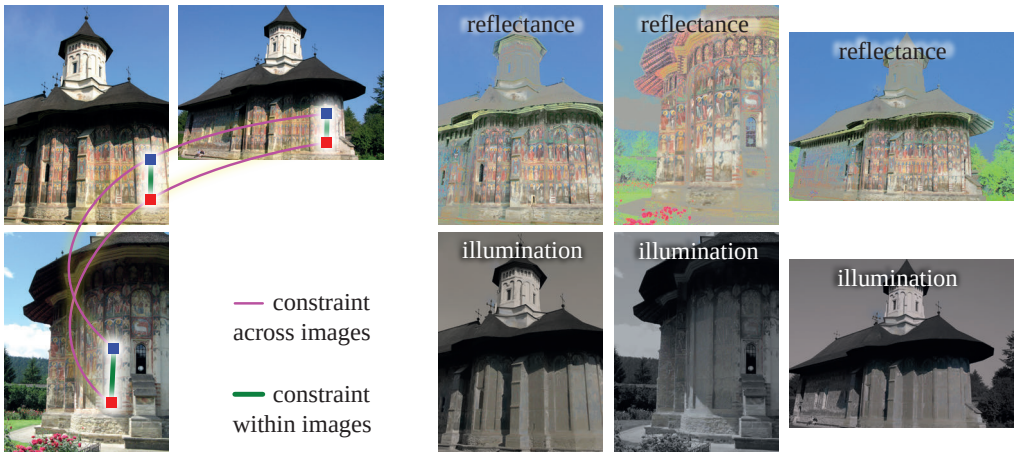
Furthermore, whereas the illumination is expected to change from image to image, we assume that the scene is mostly Lambertian so that the reflectance of a 3D point is constant across images. In the following, we drop the color channel subscript c and assume per-channel operations, unless stated explicitly.

In order to leverage the multiple illumination conditions, we need to relate scene points in different images. For this, we apply standard multiview-stereo [Furukawa 2009b] (Figure 5.1(a)), which produces an oriented point cloud of the scene and estimates for each point the list of images where it appears. For ease of notation, we make 3D projection implicit and denote the observed radiance of point $\mathbf{p}$ and the color of the pixel where it projects in image i as $\mathbf{I}_i(\mathbf{p})$.

We next infer ratios of reflectance between pairs of 3D points (Figure 5.1(b)). For a pair of points $(\mathbf{p}, \mathbf{q})$, we consider the distribution of ratios of pixel radiance $\mathbf{I}_i(\mathbf{p}) \,/\, \mathbf{I}_i(\mathbf{q})$ in all the images where both points are visible. The ratio of reflectance is equal to the median ratio of radiance if the two points have the same illumination in most lighting conditions. A contribution of our work is to identify pairs of points that share the same illumination based on geometric criteria and on the distribution of radiance ratios.



(a) off-the-shelf multiview stereo (b) reflectance ratio inference



(c) least-squares propagation within and across images (d) coherent reflectance and individual illumination layers

Figure 5.1: *Our method leverages the heterogeneity of photo collections to automatically decompose photographs of a scene into reflectance and illumination layers. We first reconstruct a 3D point cloud of the scene using automatic algorithms (a). We then derive constraints on the relative reflectance of pairs of 3D points, based on their appearance in multiple images (b). We enforce coherent reflectance of 3D points across all images and propagate the decomposition to all pixels (c). The extracted reflectance layers are coherent across all views, while the illumination layers capture the shading and shadow variations proper to each picture (d).*

Our last step solves for the illumination layer at each image based on a linear least squares formulation. It includes the constraints on reflectance ratios (depicted as green edges in Figure 5.1(c)), an image-guided interpolation term inspired by Levin et al. [Levin 2008] and Bousseau et al. [Bousseau 2009], and coherency constraints which force reflectance to be the same in all images (magenta edges in Figure 5.1(c)).

5.2 Reflectance ratios

Our method relies on reflectance ratios inferred from the multiple illumination conditions provided as input. In order to relate points in different images, we reconstruct a sparse set of 3D points and normals, and introduce a statistical criterion to reliably infer reflectance ratios.

5.2.1 Relations on reflectance between pairs of points

If two points $\mathbf{p}$ and $\mathbf{q}$ have the same normal $\vec{n}$ and receive the same incoming radiance, then the variations of the observed radiances $\mathbf{I}$ are only due to the variations of the scene reflectance $\mathbf{R}$.

Assuming Lambertian surfaces, the radiance $\mathbf{I}$ towards the camera at each non-emissive point $\mathbf{p}$ is given by the following equation:

$$\mathbf{I}(\mathbf{p}) \;=\; \mathbf{R}(\mathbf{p}) \int_{\Omega} \mathbf{L}(\mathbf{p}, \vec{\omega})\; (-\vec{\omega} \cdot \vec{n}(\mathbf{p}))\; d\vec{\omega} \tag{5.2}$$

where $\mathbf{L}(\mathbf{p}, \vec{\omega})$ is the incoming radiance arriving at $\mathbf{p}$ from direction $\vec{\omega}$, $\vec{n}(\mathbf{p})$ is the normal at $\mathbf{p}$, and Ω is the hemisphere centered at $\vec{n}(\mathbf{p})$.

Given a pair of points $\mathbf{p}$ and $\mathbf{q}$ with the same normal $\vec{n}$, we can express the ratio of radiance between the two points as

$$\frac{\mathbf{I}(\mathbf{q})}{\mathbf{I}(\mathbf{p})} = \frac{\mathbf{R}(\mathbf{q})}{\mathbf{R}(\mathbf{p})} \frac{\int_{\Omega} \mathbf{L}(\mathbf{q}, \vec{\omega})\; (-\vec{\omega} \cdot \vec{n})\; d\vec{\omega}}{\int_{\Omega} \mathbf{L}(\mathbf{p}, \vec{\omega})\; (-\vec{\omega} \cdot \vec{n})\; d\vec{\omega}}. \tag{5.3}$$

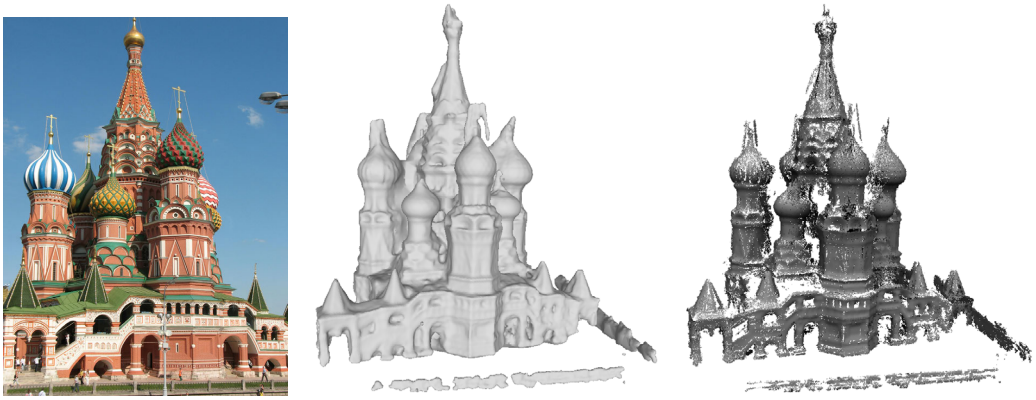
If the incoming radiance $\mathbf{L}$ is identical for both points, then the ratio of reflectances $\mathbf{R}(\mathbf{q})$ / $\mathbf{R}(\mathbf{p})$ is equal to the ratio of radiances $\mathbf{I}(\mathbf{q})$ / $\mathbf{I}(\mathbf{p})$. From multiview stereo we have a normal estimate for each point, and it is straightforward to find points with similar normals. We next find an image where lighting conditions at $\mathbf{p}$ and $\mathbf{q}$ match. For points $\mathbf{p}$ and $\mathbf{q}$ which are close, the likelihood that a shadow boundary falls between them is low. Thus for most images in which these points are visible, the radiance ratio is equal to the reflectance ratio. However, lighting may still not match in a few images. Inspired by the work of Weiss [Weiss 2001] and Matsushita et al. [Matsushita 2004a] in the context of timelapse sequences, we use the median operator as a robust estimator to deal with such

rare cases:

$$\frac{\mathbf{R}(\mathbf{q})}{\mathbf{R}(\mathbf{p})} = \underset{i \in \mathcal{I}(\mathbf{p},\mathbf{q})}{\text{median}} \left(\frac{\mathbf{I}_i(\mathbf{q})}{\mathbf{I}_i(\mathbf{p})} \right) \tag{5.4}$$

where the median is taken only over the images of the set $\mathcal{I}(\mathbf{p}, \mathbf{q}) \subset \{\mathbf{I}_i\}$ in which both $\mathbf{p}$ and $\mathbf{q}$ are visible. This equation allows us to estimate the ratio of reflectances at $\mathbf{p}$ and $\mathbf{q}$ based on the observation of their radiances in a sequence of images with varying lighting; in Section 5.3.1, this reflectance ratio will be incorporated as a constraint which guides the intrinsic decomposition.

Ambient occlusion Our derivation so far assumes that the illumination depends only on the normal orientation and is independent of the location. However, for scenes with strong concavities, differences in visibility might cause two points with similar normals to have different illumination on average, because one of them might be in shadow more often. We compensate for this by evaluating the ambient occlusion factor $\alpha(\mathbf{p})$, that is, the proportion of the hemisphere visible from $\mathbf{p}$. Ambient occlusion is computed by using a standard Poisson mesh reconstruction (in practice the one in MeshLab) to create a geometric proxy of the scene, and casting rays from the 3D points in the upper hemisphere around the normal. Figure 5.2 shows an example of reconstructed proxy and ambient occlusion estimated at 3D points.



(a) View of St. Basil (b) Approximate proxy (c) Ambient occlusion at 3D points

Figure 5.2: *Ambient occlusion estimation on the St.Basil scene (a). An approximate proxy created with Poisson reconstruction (b) is used to estimate ambient occlusion at sparse 3D points (c).*

When $\mathbf{p}$ and $\mathbf{q}$ are both in shadow, Equation 5.3 becomes:

$$\frac{\mathbf{I}(\mathbf{q})}{\mathbf{I}(\mathbf{p})} = \frac{\mathbf{R}(\mathbf{q})}{\mathbf{R}(\mathbf{p})} \frac{\alpha(\mathbf{q})}{\alpha(\mathbf{p})} \tag{5.5}$$

We account for the discrepancy in visibility by multiplying the ratio $\mathbf{I}(\mathbf{q})/\mathbf{I}(\mathbf{p})$ by $\alpha(\mathbf{p})/\alpha(\mathbf{q})$, in order to correct the reflectance ratio estimated in Equation 5.4. With this approach, we correct the case where both points are often in shadow, which happens in regions with concavities. When only one of the points is in shadow and the other receives

sunlight in most images, the pair is not reliable and will be discarded as explained in Section 5.2.2. Lastly, in the case where both points receive sunlight in most images, the ambient occlusion ratio $\frac{\alpha(\mathbf{p})}{\alpha(\mathbf{q})}$ is likely to be close to 1 since this rarely happens in concavities; in such a case, the correction does not affect the estimated reflectance ratio.

The estimation of ambient occlusion is robust to inaccurate geometry, since it averages the contribution of incoming light from all directions of the hemisphere. Models reconstructed from our point clouds are typically very coarse, but our results show that this is sufficient for correcting the estimated reflectance ratios.

5.2.2 Selection of constrained pairs

Given the set of 3D points, we need to select a tractable number of pairs whose median radiance ratio is likely to be a good estimate of the reflectance ratio. Based on the above discussion, we start with a set of geometric factors (normals and 3D distance), and add a simple statistical criterion on the observed ratios. We make things tractable by selectively subsampling the valid constraints.

Geometric criterion. For each 3D point, we select a set of candidate pairs that follow the geometric assumptions in Section 5.2.1. In most cases, the two points of a pair should be nearby and have similar normals. However, we also wish to obtain a good spatial coverage so that all regions of the point cloud are inter-connected; thus, we also select fewer pairs consisting of points which are further apart or with varying orientations. Our approach consists in sampling candidate pairs carefully by controlling the distribution of their spatial extent and orientation discrepancy. Note that this step only selects *candidate pairs* of points, on which constraints *might* be applied; unreliable pairs will be detected and discarded in the next step of the algorithm.

We define the distance $d_{\vec{n}}$ on normal orientation between two points $\mathbf{p}$ and $\mathbf{q}$ from the dot product between their normals

$$d_{\vec{n}}(\mathbf{p},\mathbf{q}) = \left|1 - \vec{n}(\mathbf{p}) \cdot \vec{n}(\mathbf{q})\right| . \tag{5.6}$$

We set $d_{3\mathrm{D}}(\mathbf{p},\mathbf{q})$ to be the Euclidean 3D distance, representing the spatial proximity of two points.

Our goal is to select N candidate pairs of points so that $d_{\vec{n}}$ and $d_{3\mathrm{D}}$ follow normal distributions $\mathcal{N}(\sigma_{\vec{n}})$ and $\mathcal{N}(\sigma_{3\mathrm{D}})$. The parameter $\sigma_{\vec{n}}$ accounts for surfaces with low curvature and inaccuracy in the normals estimated from multiview stereo. We set $\sigma_{\vec{n}} = 0.3$ for all our results, and set $\sigma_{3\mathrm{D}}$ to 20% of the spatial extent of each scene.

For a given point $\mathbf{p}$, we sample the density functions in two steps. First we select a subset of points according to $\mathcal{N}(\sigma_{3\mathrm{D}})$, and then we sample this subset according to $\mathcal{N}(\sigma_{\vec{n}})$.

In both cases, the major difficulty resides in properly accounting for the non-uniform distribution of the distance $d \in \{d_{\vec{n}}, d_{3D}\}$ in the point cloud generated by multiview stereo. We account for these non-uniform distributions with the following algorithm:

Algorithm 1 Sampling according to 3D distances or normals

1. Estimate the density of distances $f_{\text{original}}(d(\mathbf{p},\mathbf{q}))$ of all points $\mathbf{q}$ to the current point $\mathbf{p}$. We use the Matlab `ksdensity` function, which computes a probability density estimate of distances to $\mathbf{p}$ from a set of samples $d(\mathbf{p},\mathbf{q})$ by accumulating normal kernel functions centered on each sample.

2. Assign to each point $\mathbf{q}$ a sampling probability based on the desired distribution $\mathcal{N}(\sigma)$ and the density of distances f_{original}:

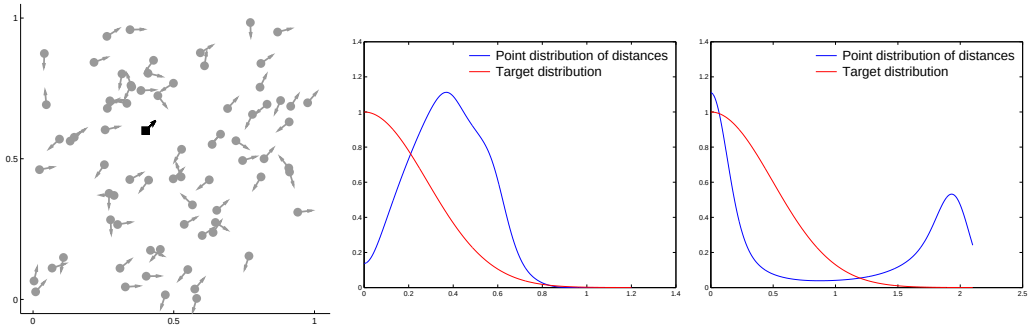
$$\Pr(\mathbf{q}) \;=\; \exp\left(-\frac{d(\mathbf{p},\mathbf{q})^2}{2\sigma^2}\right) \,/\, f_{\text{original}}(d(\mathbf{p},\mathbf{q}))$$

3. Select a subset of points according to probabilities $\Pr(\mathbf{q})$ using inversion sampling.

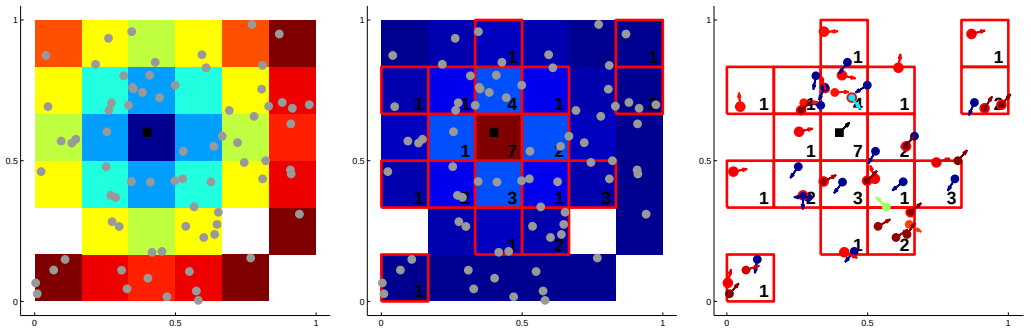
In practice, we accelerate the sampling of $\mathcal{N}(\sigma_{3D})$ by first embedding the point cloud in a 3D grid (with 10^3 non-empty cells on average). We then apply Algorithm 1 to the grid cells instead of the points, ignoring empty cells and computing d_{3D} between the centers of the cells. As a result of this first sampling we obtain a list of cells and the number of points that we need to choose in each cell to obtain a total of N pairs. We then apply Algorithm 1 according to $\mathcal{N}(\sigma_{\vec{n}})$, with the caveat that we only consider points from the cells which have been obtained in the first step, and we apply inversion sampling independently in each cell to select the proper number of points. We illustrate this process in Figure 5.3 and provide reference Matlab code.

Our sampling strategy ensures a good distribution of pairs of points, with many "short distance" pairs around the point and a few "longer distance" pairs. We also experimented with a simple threshold that selects the pairs with the highest score based on $d_{\vec{n}}$ and d_{3D}, but this naive strategy tends to only select short pairs with identical normals, which yields a weakly connected graph of constraints and results in isolated regions in the final optimization.

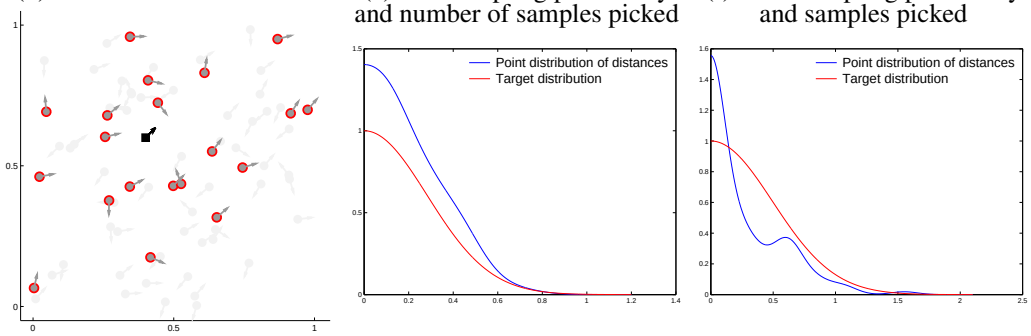
In our experiments, the sampling strategy we designed produces a well-connected graph of candidate pairs. However, we did not try to enforce connectivity of this graph, because our subsequent statistical criterion discards a large number of unreliable pairs. We used 30 candidate pairs per point in all our examples, and keep at most 1.5 million candidate pairs per scene.



(a) Initial point cloud (b) Initial distribution of d_{3D} (c) Initial distribution of $d_{\vec{n}}$



(d) Distance to reference cell (e) Cell sampling probability and number of samples picked (f) Point sampling probability and samples picked



(g) Final sampled points (h) Final distribution of d_{3D} (i) Final distribution of $d_{\vec{n}}$

Figure 5.3: *2D Illustration of our algorithm for sampling candidate pairs for a single point. (a) Given an oriented point cloud, we wish to select N points so that their distances d_{3D} and $d_{\vec{n}}$ to a reference point (black square) follow normal distributions. (b-c) The point cloud is irregularly sampled, and the distribution of distances of all points (blue curves) is very different from the target normal distributions (red curves). (d) We first embed the point cloud in a grid and compute the Euclidean distance d_{3D} to the cell containing the reference point: the distance is color-coded (blue: small distance; dark red: large distance). (e) We infer a sampling probability for each cell based on d_{3D} as described in Algorithm 1; this sampling probability is shown color-coded for each cell (blue: low sampling probability; dark red: large sampling probability). From these probabilities, we draw N samples to choose the number of points to select in each cell, shown as black numbers in the corresponding highlighted cells. We discard all points contained in cells for which no sample has been drawn. (f) For all the points within sampled cells, we infer a sampling probability based on $d_{\vec{n}}$ (shown color-coded; blue: low sampling probability; dark red: large sampling probability). We draw samples in each cell from these probabilities; the number of samples drawn in each cell corresponds to the result of (e). (g) The final samples are distributed so that many points are nearby and have similar normals compared to the reference point, while a few are further away or with different normals to produce a well-connected graph of constraints. (h-i) The distribution of distances of sampled points (blue curves) is closer to the desired normal distributions (red curves). We provide the Matlab sampling code used to generate this figure with the following parameters: 150 points in the point cloud and 35 samples drawn, for the illustrations (a, d-g); 500000 points in the point cloud, and 100 samples drawn, for the distributions estimated with the Matlab* `ksdensity` *function (b-c, h-i).*

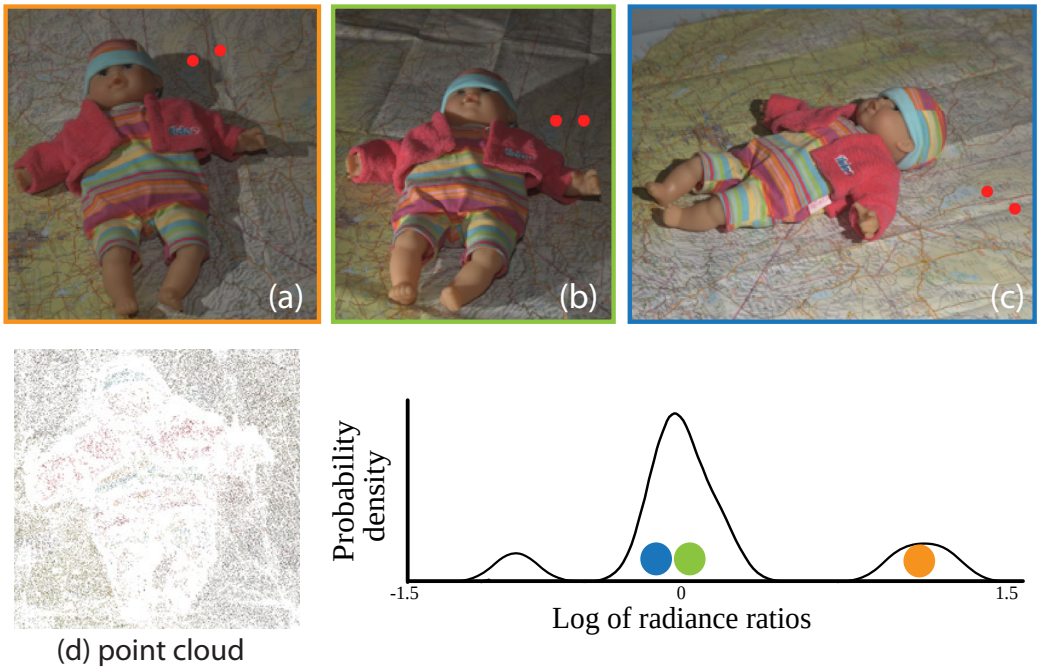


Figure 5.4: *Analysis of the distribution of radiance ratio (red channel, log scale) between two 3D points (red dots) with similar normals, under varying viewpoints and lighting. The probability distribution function has a dominant lobe, corresponding to (b) and (c) where both points receive approximately the same incoming radiance. In (a), the light is visible from only one of the points and the corresponding radiance ratio falls in a side lobe. (d) shows the point cloud for image (a).*

Photometric statistical criterion. Each candidate pair $(\mathbf{p}, \mathbf{q})$ can be observed in a subset of input images $\mathcal{I}(\mathbf{p}, \mathbf{q})$. Figure 5.4 illustrates the probability density function (PDF) of the ratio of radiances of a pair over multiple images with different lighting. When the two points fulfill our assumptions, the distribution has a dominant lobe well captured by the median operator. In such a case, the reflectance ratio of the pair can be estimated with the median (Equation 5.4). However when the two points receive different incoming radiance in more than 50% of the images, the distribution is spread and not necessarily centered at the median. We detect and reject such unreliable pairs of 3D points, by counting the observations of their radiance ratio which are far from the median value. The observation of pair $(\mathbf{p}, \mathbf{q})$ in image j is considered far from the median if

$$\left| \log\left(\frac{\mathbf{I}_j(\mathbf{q})}{\mathbf{I}_j(\mathbf{p})}\right) - \underset{i \in \mathcal{I}(\mathbf{p},\mathbf{q})}{\text{median}} \log\left(\frac{\mathbf{I}_i(\mathbf{q})}{\mathbf{I}_i(\mathbf{p})}\right) \right| > 0.15 \tag{5.7}$$

in at least one channel. We consider a pair to be unreliable if it has less than 50% of the radiance ratio values close to the median, or if it is visible in less than 5 images (too few observations). Candidate pairs that are considered reliable will be used to constrain the intrinsic image decomposition (Section 5.3.1).

What would happen for a cube? In theory, the faces of a cube are not connected by pairwise constraints because they have orthogonal normals. However, the solution is also influenced by an image-guided smoothness prior (Section 5.3.2) and a coherence term that enforces the reflectance to remain the same under different illumination conditions (Section 5.3.3). The faces may also be indirectly connected via other objects in the scene. In our experiments, these additional constraints were enough to obtain plausible decompositions even on cube-like scenes (Temple in Figure 5.10, first row; RizziHaus in Figure 5.12, third row).

5.3 Multi-Image Guided Decomposition

We now have a sparse set of constraints on the ratio of reflectance at pairs of 3D points. To obtain values everywhere, we formulate an energy function over the RGB illumination $\mathbf{S}$ at each pixel of each image. Our energy includes data terms on the reflectance ratios, an image-guided interpolation term, and a set of constraints that enforce the coherence of the reflectance between multiple images. This results in a large sparse linear least square system, which we solve in a staggered fashion.

5.3.1 Pairwise reflectance constraints

Given the ratio between the reflectances of pixels corresponding to points $\mathbf{p}$ and $\mathbf{q}$ in Equation 5.4, we deduce ratios $\mathbf{Q}_j(\mathbf{p},\mathbf{q})$ between the illumination of the corresponding pixels in image j:

$$\mathbf{Q}_j(\mathbf{p},\mathbf{q}) = \frac{\mathbf{S}_j(\mathbf{p})}{\mathbf{S}_j(\mathbf{q})} = \frac{\mathbf{I}_j(\mathbf{p})}{\mathbf{I}_j(\mathbf{q})}\frac{\mathbf{R}(\mathbf{q})}{\mathbf{R}(\mathbf{p})} \tag{5.8a}$$

$$= \frac{\mathbf{I}_j(\mathbf{p})}{\mathbf{I}_j(\mathbf{q})} \underset{i\in\mathcal{I}(\mathbf{p},\mathbf{q})}{\text{median}} \left(\frac{\mathbf{I}_i(\mathbf{q})}{\mathbf{I}_i(\mathbf{p})}\right) \tag{5.8b}$$

where $\mathbf{S}_j$ is the illumination layer of image j. This formula lets us write a constraint on the unknown illumination values of pixels corresponding to $\mathbf{p}$ and $\mathbf{q}$ in image j:

$$\mathbf{Q}_j(\mathbf{p},\mathbf{q})^{\frac{1}{2}}\,\mathbf{S}_j(\mathbf{q}) \;=\; \mathbf{Q}_j(\mathbf{p},\mathbf{q})^{-\frac{1}{2}}\,\mathbf{S}_j(\mathbf{p}) \tag{5.9}$$

We combine the contribution of all the constrained pairs selected in Section 5.2.2, in all the images where they are visible, and express these constraints in a least-squares sense to get the energy $E_{\text{constraints}}$:

$$E_{\text{constraints}} = \sum_{(\mathbf{p},\mathbf{q})} \sum_{j\in\mathcal{I}(\mathbf{p},\mathbf{q})} \left[\mathbf{Q}_j(\mathbf{p},\mathbf{q})^{\frac{1}{2}}\,\mathbf{S}_j(\mathbf{q}) - \mathbf{Q}_j(\mathbf{p},\mathbf{q})^{-\frac{1}{2}}\,\mathbf{S}_j(\mathbf{p})\right]^2 \tag{5.10}$$

In practice, we have one such term for each RGB channel.

5.3.2 Smoothness

We build our smoothness prior on the intrinsic images algorithm of Bousseau et al. [Bousseau 2009] that was designed to propagate sparse user indications for separating reflectance and illumination in a single image, and on the closely related *Matting Laplacian* introduced by Levin et al. [Levin 2008] for scribble-based matting. The former assumes a linear relationship between the unknowns and the image channels and the latter an affine relationship. We experimented with both, and while the intrinsic image prior captures variations of illumination at a long distance from the constrained pixels, we found that the matting prior yields smoother illumination in regions with varying reflectance, especially in our context where many pixels are constrained. We show a comparison between these priors in Section 5.4.2.1.

Our prior translates into a local energy for each pixel neighborhood that relates the color at a pixel x with the illumination value in each channel $\mathbf{S}_{jc}(x)$ using an affine model:

$$\sum_{c\in\{r,g,b\}} \sum_{y\in\mathcal{W}_j^x} \left(\mathbf{S}_{jc}(y) - \mathbf{a}_{jc}^x \cdot \mathbf{I}_j(y) - b_{jc}^x\right)^2 + \epsilon\,(\mathbf{a}_{jc}^x)^2 \tag{5.11}$$

where $\mathcal{W}_j^x$ is a 3×3 window centered on x, $\mathbf{a}_j^x$ and b_j^x are the unknown parameters of the affine model, constant over the window, and $\epsilon = 10^{-6}$ is a parameter controlling the regularization $(\mathbf{a}_j^x)^2$ that favors smooth solutions. Levin et al. [Levin 2008] showed that $\mathbf{a}_j^x$ and b_j^x can be expressed as functions of $\mathbf{S}_j$ and removed from the system. Then, summing over all pixels and all images yields an energy that only depends on the illumination, and can be expressed in matrix form:

$$E_{\text{smoothness}} = \sum_{c\in\{r,g,b\}} \sum_j \hat{\mathbf{S}}_{jc}^{\mathsf{T}} M_{jc} \hat{\mathbf{S}}_{jc} \tag{5.12}$$

where the vectors $\hat{\mathbf{S}}_j$ stack the unknown illumination values in image j and the matrices M_j encode the smoothness prior over each pixel neighborhood in this image (see the paper by Levin et al. [Levin 2008] for the complete derivation).

We found that it is beneficial to add a grayscale regularization for scenes with small concavities in shadow. Because these areas often have no (or very few) reconstructed 3D points, they are influenced by their surrounding lit areas and illumination tends to be overestimated. For such scenes, we add the term below to favor illumination values close to the image luminance:

$$\sum_x \sum_{c\in\{\text{r,g,b}\}} \left(\mathbf{S}_{jc}(x) - \frac{1}{3}\left[\mathbf{I}_{j\text{r}}(x) + \mathbf{I}_{j\text{g}}(x) + \mathbf{I}_{j\text{b}}(x)\right]\right)^2 \tag{5.13}$$

We use a small weight (10^{-3}) so that this term affects only regions with no other constraints. We show in Section 5.4.2.1 that although results are satisfactory without it, this term helps further improve the decomposition.

5.3.3 Coherent reflectance

For photo collections, it is important to ensure that the intrinsic image decomposition is coherent across images from the collection. We impose additional constraints across images by enforcing the reflectance of a 3D point to be constant over all views where it appears.

Consider the case where a given point $\mathbf{p}$ is visible in two images $\mathbf{I}_m$ and $\mathbf{I}_n$. For each such pair (m, n) of images we want to force the pixels corresponding to $\mathbf{p}$ to have the same reflectance, and thus infer a constraint on their illumination:

$$\begin{aligned} \mathbf{R}_m(\mathbf{p}) = \mathbf{R}_n(\mathbf{p}) &\Rightarrow \frac{\mathbf{I}_m(\mathbf{p})}{\mathbf{S}_m(\mathbf{p})} = \frac{\mathbf{I}_n(\mathbf{p})}{\mathbf{S}_n(\mathbf{p})} \\ &\Rightarrow \mathbf{I}_m(\mathbf{p})\,\mathbf{S}_n(\mathbf{p}) = \mathbf{I}_n(\mathbf{p})\,\mathbf{S}_m(\mathbf{p}) \end{aligned} \tag{5.14}$$

We denote $\mathcal{I}(\mathbf{p}) \subset \{\mathbf{I}_i\}$ the subset of images where the point $\mathbf{p}$ is visible. Summing the contribution of every pair of images where a point appears gives us an additional energy term $E_{\text{coherence}}$ that encourages coherent reflectance across images:

$$E_{\text{coherence}} = \sum_{\mathbf{p}} \sum_{m \in \mathcal{I}(\mathbf{p})} \sum_{\substack{n \in \mathcal{I}(\mathbf{p}) \\ n > m}} \left(\mathbf{I}_m(\mathbf{p})\,\mathbf{S}_n(\mathbf{p}) - \mathbf{I}_n(\mathbf{p})\,\mathbf{S}_m(\mathbf{p})\right)^2 \tag{5.15}$$

This term generates a large number of constraints. We found that applying them only at the points selected in Section 5.2.2 yields equivalent results while reducing the complexity of the system. In addition, we describe an efficient solver in Section 5.3.4.

5.3.4 Solving the system

We combine the energy terms defined above with weights $w_{\text{constraints}} = 1$, $w_{\text{smoothness}} = 1$ and $w_{\text{coherence}} = 10$, fixed for all our results. Minimizing this global energy translates into solving a sparse linear system where the unknowns are the illumination values at each pixel of each image. We obtain the reflectance at each pixel by dividing the input images by the estimated illuminations.

Our system is large because it includes unknowns for all the pixels of all the images to decompose. To make things tractable, we use an iterative approach akin to a blockwise Gauss-Seidel solver, where each iteration solves for the illumination of one image with the values in all the other images fixed. The advantage of this approach is that we can reduce Equation 5.15 to a single term per point $\mathbf{p}$, at each iteration. To show this, we first write the energy $E^k_{\text{coherence}}(m, \mathbf{p})$ for point $\mathbf{p}$ in image m at iteration k:

$$E^k_{\text{coherence}}(m,\mathbf{p}) = \sum_{\substack{n\in\mathcal{I}(\mathbf{p})\\ n<m}} \left(\mathbf{I}_m(\mathbf{p})\,\mathbf{S}^k_n(\mathbf{p}) - \mathbf{I}_n(\mathbf{p})\,\mathbf{S}^k_m(\mathbf{p})\right)^2 + \sum_{\substack{n\in\mathcal{I}(\mathbf{p})\\ n>m}} \left(\mathbf{I}_m(\mathbf{p})\,\mathbf{S}^{(k-1)}_n(\mathbf{p}) - \mathbf{I}_n(\mathbf{p})\,\mathbf{S}^k_m(\mathbf{p})\right)^2 \tag{5.16}$$

In this energy, the only variable is $\mathbf{S}^k_m(\mathbf{p})$, everything else is fixed. Since all the terms in Equation 5.16 are quadratic functions depending on the same variables, the energy can be rewritten as a single least-squares term, plus a constant which does not depend on $\mathbf{S}^k_m(\mathbf{p})$:

$$\left(\sum_{\substack{n\in\mathcal{I}\\ n\neq m}} \mathbf{I}_n^2\right)\left(\mathbf{S}^k_m - \frac{\mathbf{I}_m\left(\sum_{\substack{n\in\mathcal{I}\\ n\neq m}} \mathbf{I}_n\,\mathbf{S}^{\tilde{k}}_n\right)}{\sum_{\substack{n\in\mathcal{I}\\ n\neq m}} \mathbf{I}_n^2}\right)^2 + \text{constant} \tag{5.17}$$

where for clarity, we use the notation $\mathbf{S}^{\tilde{k}}_n = \mathbf{S}^k_n(\mathbf{p})$ when $n < m$ and $\mathbf{S}^{(k+1)}_n(\mathbf{p})$ when $n > m$, and omit the dependency on $\mathbf{p}$. Equation 5.17 expresses the inter-images constraints on $\mathbf{S}^k_m(\mathbf{p})$ as a single least-squares term, which shows that these constraints are tractable even though there is a quadratic number of them. Further, when we derive this term to obtain the corresponding linear equation used in our solver, the factor $(\sum_{\substack{n\in\mathcal{I}\\ n\neq m}} \mathbf{I}_n^2)$ and the denominator cancel out, ensuring that our system does not become unstable with small values of $\sum_{\substack{n\in\mathcal{I}\\ n\neq m}} \mathbf{I}_n^2$.

To initialize this iterative optimization, we compute an initial guess of the illumination in each image with an optimization where we only use the single-image terms $E_{\text{constraints}}$, $E_{\text{smoothness}}$, and the grayscale regularization. The energy decreases quickly during the first few iterations of the optimization process, then converges to a plateau value.

5.4 Implementation and Results

3D Reconstruction. We first apply bundle adjustment [Wu 2011] to estimate the parameters of the cameras and patch-based multi-view stereo [Furukawa 2009b] to generate a 3D point cloud of the scene. For each point, this algorithm also estimates the list of photographs where it appears. We compute normals over this point cloud using the PCA approach of Hoppe et al. [Hoppe 1992]. We used 103 images per scene on average to perform reconstruction, but this varies significantly depending on the scene (e.g., we used 11 viewpoints to reconstruct the "Doll" scene).

Point cloud resampling. Multi-view reconstruction processes full-sized photographs, while we apply our decomposition on smaller images for efficiency. Multiple nearby 3D points may project to the same pixels on the resized images. We downsample the point cloud so that at most one 3D point projects at each pixel in each image, using a greedy algorithm which keeps in priority points that are visible in most images. To do so, we visit every pixel of every image, creating the point set as we proceed. At a given pixel, we first test if a point of the set already projects to it. If not, we choose the point which is visible in the largest number of images, and we add it to the set. We limit the point cloud to 200k points since we did not observe a significant quality improvement with more points. We also discard points that project on strong edges because their radiance tends to result from a mixture of reflectances that varies among images: we discard a point if the variance of the radiance in adjacent pixels is greater than 4×10^{-3}.

Performance. The average running time of our method is 90 minutes for the 9 scenes in this chapter, on a 2.80Ghz Xeon machine with 8 cores. Our unoptimized single-core Matlab implementation of the sampling algorithm (Section 5.2.2) takes 52 minutes on average, and the selection of reliable constraints takes less than a minute. Each iteration of the optimization takes 6 minutes on average; we use Matlab's backslash operator to solve for each image within one iteration. We could greatly speed up our method by parallelization and a multi-scale approach.

5.4.1 Intrinsic Decompositions

We demonstrate our coherent intrinsic image decomposition on three types of data. First, we apply our method to synthetic data which allows an evaluation with respect to ground truth, and thorough comparison to the results which were kindly provided by authors of previous work. We then show results of our method for scenes that we capture indoors, in which we have placed cameras and lights around objects in a room. We finally apply our method to online photo collections.

5.4.1.1 Synthetic dataset

We evaluate our method on a synthetic dataset that we have created. This dataset consists of realistic renderings of a complex outdoor scene, from multiple viewpoints and under varying illuminations, along with a ground truth decomposition for each image. We use a diffuse model of the St. Basil Cathedral because it contains complex geometric details and a colorful spatially varying reflectance, in addition to occluded areas that are challenging for our approach (Section 5.2.1).

(a) Synthetic environment map (b) Captured environment map

Figure 5.5: *Examples of sky environment maps used for two renderings shown in Figure 5.6. (a) Synthetic sky created with a parametric sky model [Preetham 1999], around 1pm on a summer day. (b) Captured sky at dusk time (source: CGSkies).*

Dataset construction. We render the scene and compute ground truth illumination using path-tracing. We use a physically-based sun and sky model [Preetham 1999] for daylight, and captured environment maps for sunset/sunrise and night conditions (Figure 5.5). We rendered 30 different viewpoints over the course of three days (in summer, autumn and winter).

We use a modified version of PBRT [Pharr 2010] to obtain ground truth reflectance and illumination (as irradiance) at each pixel. There is one caveat, however: in Section 2.3.1, we showed that the intrinsic illumination is equivalent to the irradiance under the assumption that the radiance towards the camera is constant over each pixel. This does not necessary hold when a pixel straddles a reflectance edge, or on curved surfaces. Without loss of generality, let us analyze the case of a pixel $\mathbf{I}_{\text{pixel}}$ which observes two radiances $\mathbf{I}_1$ and $\mathbf{I}_2$, received in equal proportions:

$$\begin{aligned} \mathbf{I}_{\text{pixel}} &= (\mathbf{I}_1 + \mathbf{I}_2) \,/\, 2 \\ \mathbf{R}_{\text{pixel}} * \mathbf{S}_{\text{pixel}} &= (\mathbf{R}_1 * \mathbf{S}_1 + \mathbf{R}_2 * \mathbf{S}_2) \,/\, 2 \end{aligned} \tag{5.18}$$

We can distinguish three cases:

- when the reflectance is constant over the two regions, $\mathbf{R}_{\text{pixel}} = \mathbf{R}_1 = \mathbf{R}_2$ and the pixel illumination is a blend of the individual irradiances: $\mathbf{S}_{\text{pixel}} = (\mathbf{S}_1 + \mathbf{S}_2) \,/\, 2$
- when the illumination is constant over the two regions, $\mathbf{S}_{\text{pixel}} = \mathbf{S}_1 = \mathbf{S}_2$ and the pixel reflectance is a blend of the individual reflectances: $\mathbf{R}_{\text{pixel}} = (\mathbf{R}_1 + \mathbf{R}_2) \,/\, 2$
- when *both* the reflectance and the illumination change, we cannot identify separately $\mathbf{R}_{\text{pixel}}$ and $\mathbf{S}_{\text{pixel}}$ from Equation 5.18. This case occurs in particular at occlusion boundaries, because object changes may involve orientation and material variations. Averaging the reflectance and illumination separately (i.e., $\mathbf{S}_{\text{pixel}} = (\mathbf{S}_1 + \mathbf{S}_2) \,/\, 2$ and $\mathbf{R}_{\text{pixel}} = (\mathbf{R}_1 + \mathbf{R}_2) \,/\, 2$) would lead to an invalid intrinsic decomposition, because:

$$(\mathbf{R}_1 * \mathbf{S}_1 + \mathbf{R}_2 * \mathbf{S}_2) \,/\, 2 \neq ((\mathbf{S}_1 + \mathbf{S}_2) \,/\, 2) * ((\mathbf{R}_1 + \mathbf{R}_2) \,/\, 2). \tag{5.19}$$

Informed with this analysis, we create *ground truth* illumination images by averaging the irradiance of primary rays within each pixel. We then create ground truth reflectance images by dividing the rendering with the ground truth illumination: $\mathbf{R} = \mathbf{I}/\mathbf{S}$.

We select three representative images of the scene, which are illustrated in Figure 5.6. The first view exhibits mostly monochromatic illumination, and few shadows. The second view corresponds to a morning appearance, with slightly colored lighting and significant shadows. In the third view, we used a dusk sky and added point light sources on the ground to simulate urban lights; this results in colored lighting, with blue shadows cast on the cathedral towers. We provided the authors of single-image intrinsic decomposition methods with the input images (in two versions: gamma-encoded JPEG, and linear EXR) and the associated ground truth.

Sampling the geometry. In order to apply our method, we sample the 3D model to generate a 3D point cloud. We could not apply our standard reconstruction pipeline on the synthetic dataset because structure-from-motion fails to estimate the camera parameters; this also allows us to evaluate the performance of our algorithm independently of the quality of multi-view reconstruction. We cast rays from the camera centers to each pixel of the evaluation images Figure 5.6. We create a sparse oriented point cloud from the intersection of these rays with the geometry. We uniformly downsample the point cloud to 100k points for the first evaluation (this corresponds to 1 reconstructed point every 16 non-sky pixels, on average), and we later study the influence of the point density on the decomposition results.

Quantitative evaluation and comparisons. We evaluate the results of our approach, as well as several state-of-the-art single-image automatic and user-assisted methods, all kindly provided by the authors of the previous work. We also implemented an extension of [Weiss 2001] to multiple views: inspired by [Liu 2008], we first align all images with the viewpoint of the photograph to decompose. While Liu et al. use a global homography or warp a mesh based on detected feature points, we avoid mesh folding by exploiting the reconstructed sparse geometry. We construct a height field for each image and use it to project all images on the viewpoint to decompose. We then compute the intrinsic decomposition as in [Liu 2008], ignoring pixels from views where they are not visible or occluded by the geometry.

We estimate the Local Mean Squared Error (LMSE) of each method with respect to ground truth, as in [Grosse 2009], and average it over the three comparison views. We report the error in Table 5.1 and plot it in Figure 5.7. We also provide a visual comparison of all methods and ground truth for one view in Figure 5.8.

For this benchmark, our approach produces results that closely match ground truth and quantitatively outperforms all the tested methods. In particular, we successfully decompose the night picture while automatic methods fail to handle the yellow spot lights and

blue shadows; only the multi-image approach (extended [Weiss 2001]), and to a certain extent the user-assisted approach of [Bousseau 2009] are able to extract some of the colored illumination. Our method extracts most of colored texture from the illumination layer in this challenging case; it also disambiguates regions of same reflectance chromaticity, such as the white balconies and grey brick arches, whereas single-image approaches estimate those as similar reflectances and different illuminations. In addition, our method also produces coherent reflectance between all views (shown in accompanying materials) despite the drastic change of lighting.

	Image 1	Image 2	Image 3	**Average**
[Shen 2011a]	0.0422	0.0432	0.0441	0.0432
[Garces 2012]	0.0346	0.0417	0.0408	0.0390
[Bousseau 2009]	0.0305	0.0344	0.0500	0.0383
[Zhao 2012]	0.0275	0.0226	0.0343	0.0281
[Shen 2011b]	0.0200	0.0158	0.0262	0.0207
Extended [Weiss 2001]	0.0201	0.0216	0.0199	0.0205
Ours	0.0159	0.0119	0.0116	0.0131

Table 5.1: *Local Mean Squared Error of several intrinsic image decomposition methods, on the three comparison images of our synthetic dataset. The average error is plotted on Figure 5.7.*

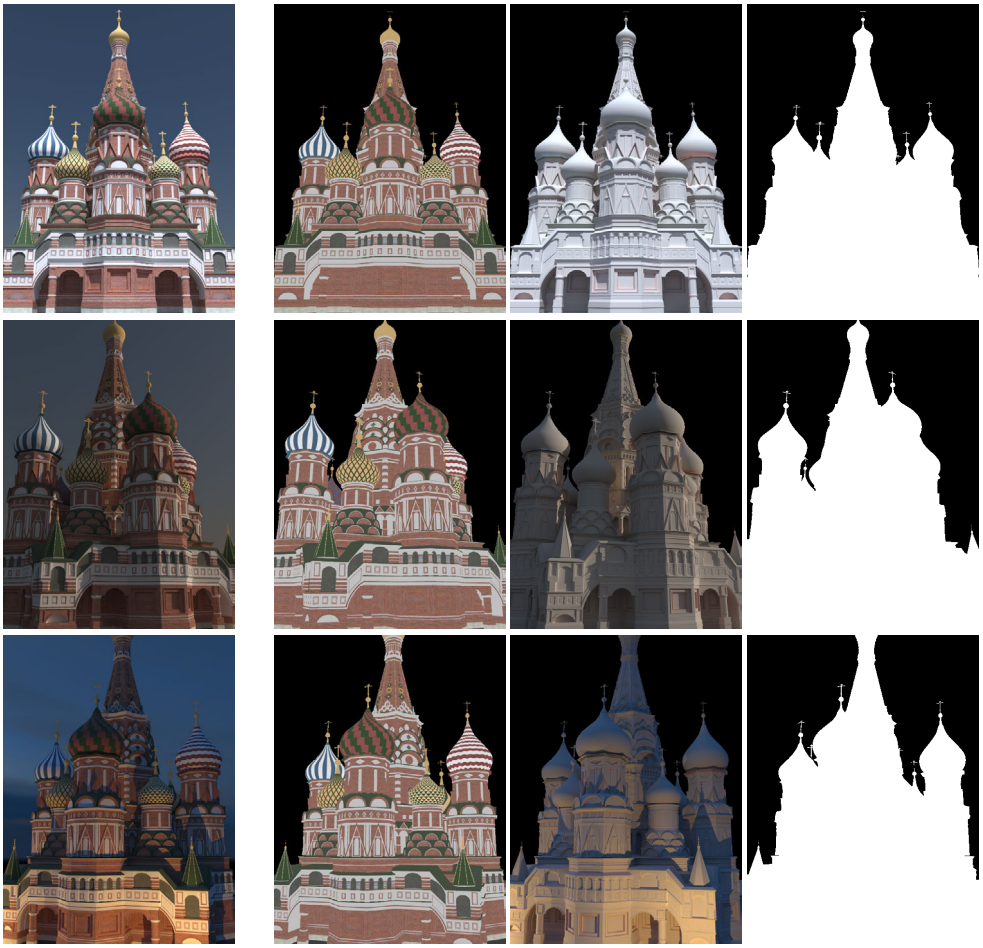


Figure 5.6: *Renderings of the synthetic StBasil scene. We provided authors of concurrent methods with the input images on the left, as well as the ground truth reflectance and illumination images (second and third column). All the results are then evaluated on non-sky pixels (right column).*

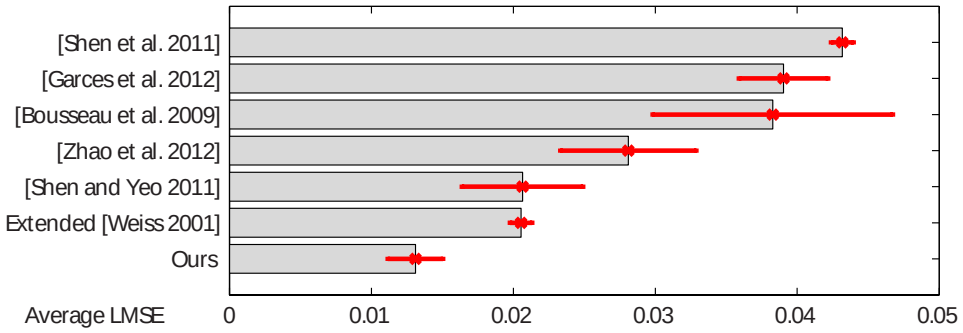
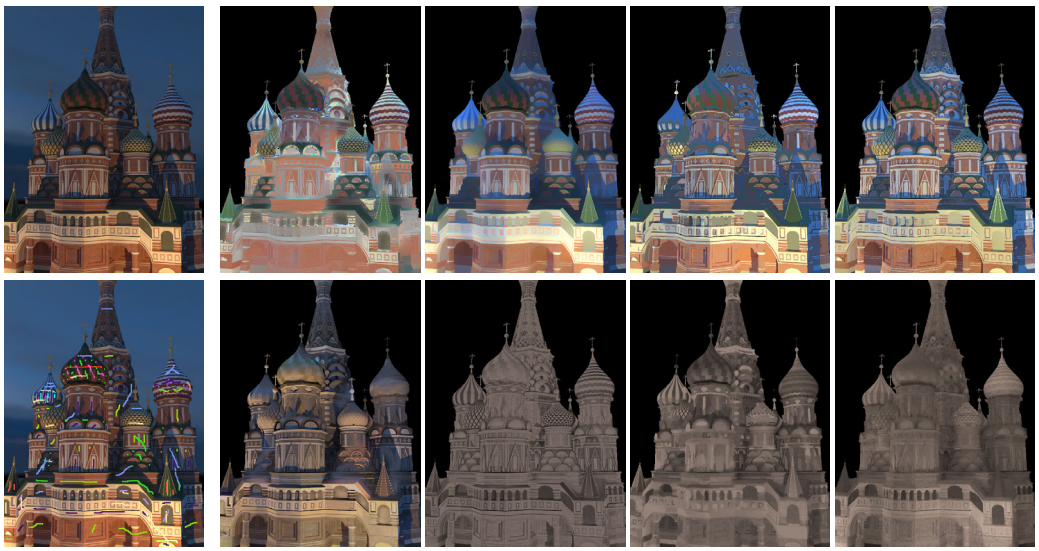
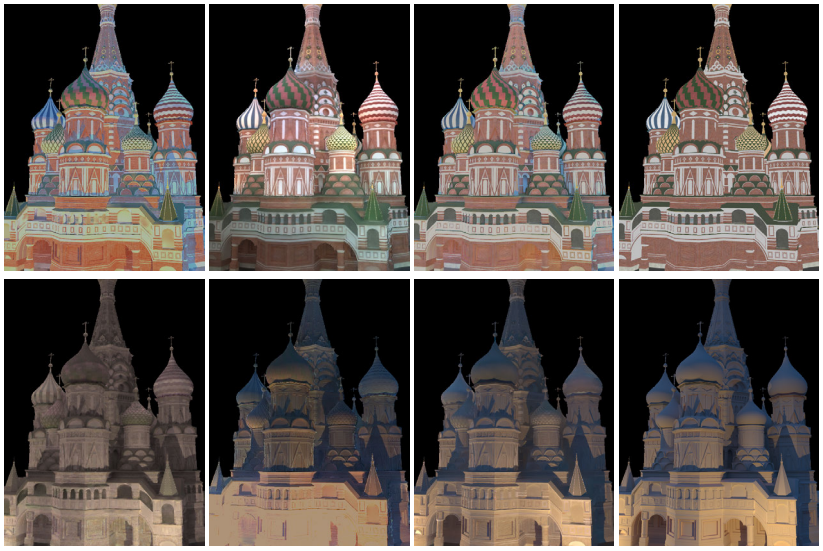


Figure 5.7: *Numerical evaluation of seven intrinsic decomposition methods on our synthetic dataset. Grey bars indicate the LMS error averaged over all comparison images, while red bars illustrate the standard deviation of LMSE across the three comparison images.*



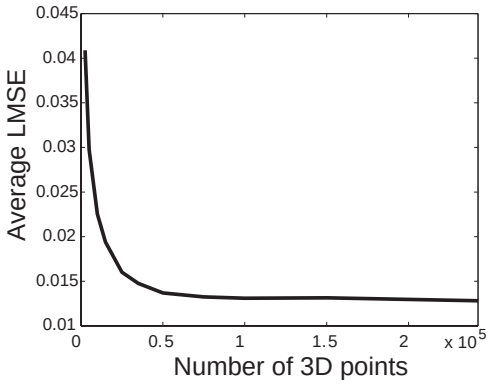
Input and scribbles for [Bousseau 2009] | [Bousseau 2009] | [Shen 2011a] | [Garces 2012] | [Zhao 2012]



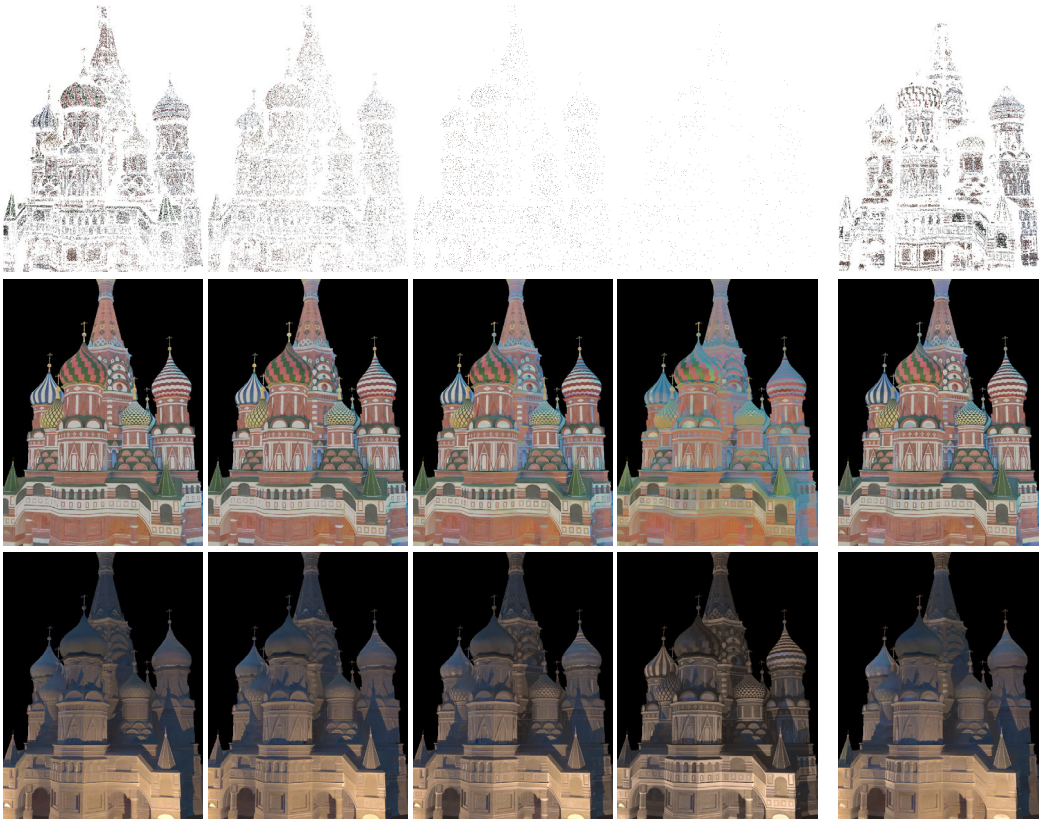
[Shen 2011b] | Extended [Weiss 2001] | Our results | Ground truth

Figure 5.8: *Comparison to existing methods and ground truth on one view of our synthetic dataset. Reflectance and illumination images have been scaled to best match ground truth; sky pixels have been removed. For each decomposition, the top row contains the reflectance and the bottom row shows the illumination. The error of each method is reported in Figure 5.7.*

Influence of reconstruction density We study the robustness of our algorithm by varying the number of 3D points in the sparse point cloud. We uniformly subsample the original point cloud and select between 2500 and 240k points. We show the resulting LMS error in the side graph: our approach still outperforms the best single-image based technique when only 15000 points are used (i.e., one 3D point every 110 non-sky pixels on average).



We perform another experiment and vary the number and distribution of 3D points. Instead of sampling the geometry, we specify ground truth camera parameters to replace the output of structure-from-motion (since those techniques fail on our synthetic images)



Sampling with 240k points (LMSE 0.013) | Sampling with 50k points (LMSE 0.014) | Sampling with 15k points (LMSE 0.019) | Sampling with 2.5k points (LMSE 0.041) | PMVS reconstruction (LMSE 0.014)

Figure 5.9: *Influence of the point cloud density and reconstruction method. Top row: constrained 3D points and their estimated reflectance. Middle row: estimated reflectance. Bottom row: estimated illumination. For each setting, we report the LMS error on this view.*

then apply PMVS to reconstruct the oriented point cloud. Our decomposition using this reconstruction yields an average LMSE of 0.01564, still significantly lower than all the approaches compared on Table 5.1. This slight decrease in performance could be due to inaccurate estimation of 3D position and normals, and differences in the spatial distribution of reconstructed points. In particular, PMVS tends to yield an irregularly sampled point cloud, where some regions are densely reconstructed while others contain large holes (see Figure 5.9, top right).

We show a visual comparison of the results with different reconstructions on Figure 5.9, and illustrate the density of the 3D point cloud in the top row. When fewer 3D points are reconstructed, only few pairwise and coherence constraints can be imposed and the decomposition mostly relies on our image-guided smoothness prior.

5.4.1.2 Captured scenes

We set up two indoors scenes containing small objects and use two light sources: a camera-mounted flash with low intensity, which adds ambient light in shadows (fill light), while a remote-controlled flash produces strong lighting from a separate direction. This setup allows us to validate our algorithm on real photographs, while avoiding the difficulties inherent in internet photo collections, such as the use of different camera settings or contrast and color manipulation that affect the validity of our assumptions. We shoot RAW photographs, from which we recover linear images.

Figure 5.10 shows our decomposition for the "Doll" and "Temple" scenes. We used 11 and 10 viewpoints respectively, and 7 different lighting conditions. Both scenes contain colored reflectances (cloth of the baby doll, texture of the tabletop) and strong hard shadows that are successfully decomposed by our method. Non-lambertian components of the reflectance (such as the specularities on the tablecloth) are assigned to the illumination layer, since coherency constraints enforce similar reflectance across images.

We also provide a visual comparison to previous work on a similar "Doll" scene, shown in Figure 5.11. Our results are on par with the best previous decompositions, and we handle the complex background textures and strong cast shadows which did not appear in the image used in previous work.

Figure 5.10: *Results of our decomposition on scenes captured with a flash ("Temple" and "Doll"). Note that the colored illumination on the doll is due mainly to indirect light, and the reflectance is coherent across views.*

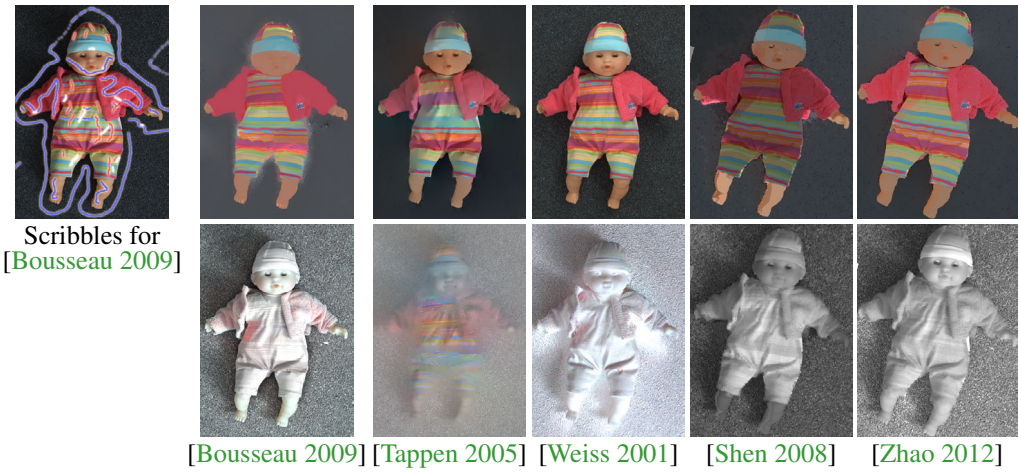


Figure 5.11: *Results of existing single-image methods on pictures of a doll. We captured our own version of a similar doll from different viewpoints with a moving light source (flash) and show our results in Figure 5.10. Although our input photographs are more challenging due to the background texture and shadows cast on the doll, our automatic method successfully recovers a smooth illumination layer and a shading-free reflectance layer.*

5.4.1.3 Internet photo collections

The last set of results we show is on internet photo collections of famous landmarks; we chose challenging scenes with interesting lighting and shadowing effects. We download images from Flickr© or Photosynth©, avoiding pictures that have been overly edited. We use 45 images on average to compute the pairwise reflectance ratios (Equation 5.4), and perform the intrinsic decomposition on around 10 images per dataset. Table 5.2 lists the number of images used for each scene.

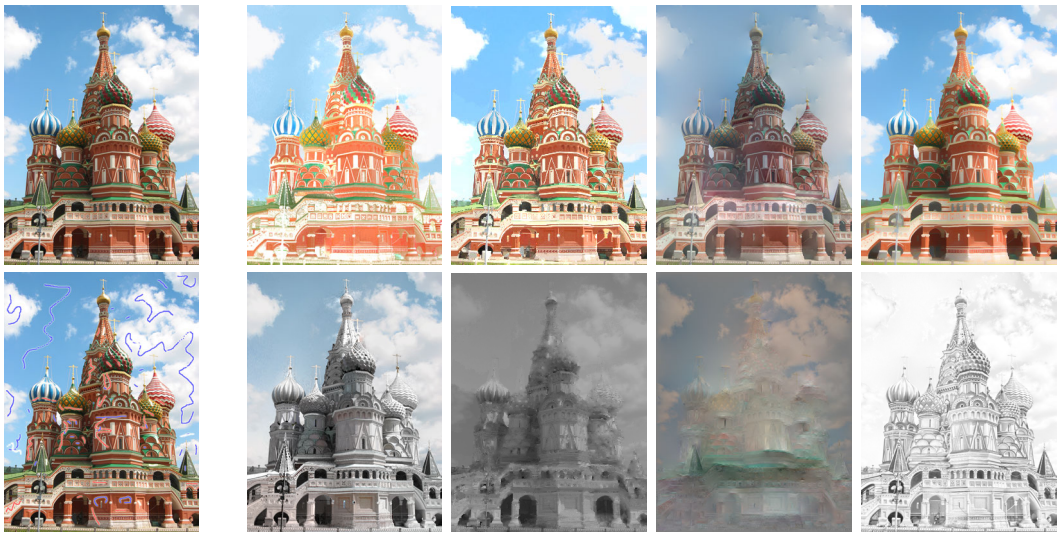
We correct radial distortion using camera parameters estimated from scene reconstruction. We assume a gamma correction of 2.2 which is common for jpeg images. However, noise and non-linearities in the camera response can generate unreliable pixels which have very low values in some channels; in such cases we recover reliable information from other channels when available.

Results on downloaded scenes. Figure 5.12 illustrates our results on several scenes, namely St. Basil, Knossos, RizziHaus, and Moldovita. Our method successfully decomposes the input image sets into intrinsic images, despite the complex spatially-varying reflectance and strong cast shadows.

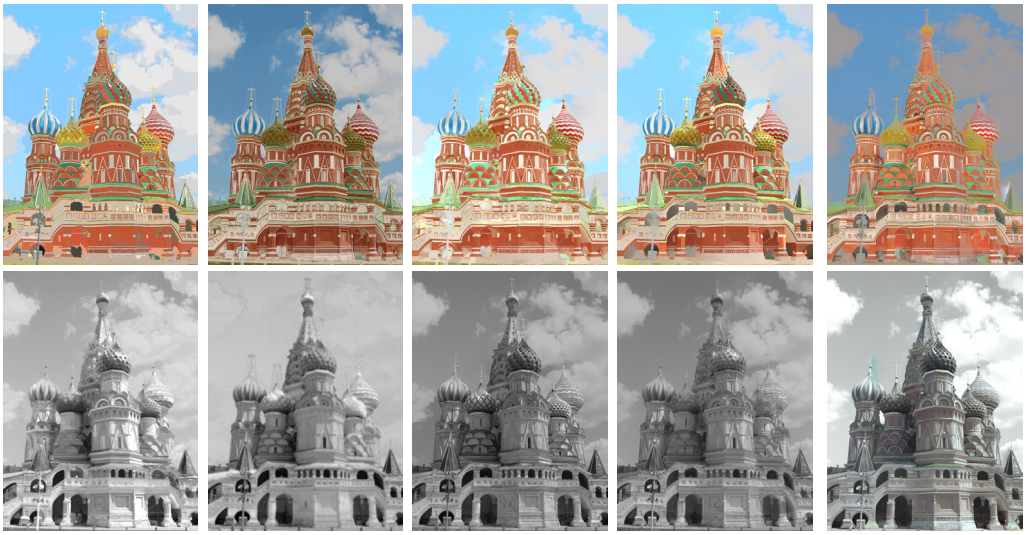
Comparison on St. Basil. In Figure 5.13 we present a side-by-side comparison with several existing single-image methods on a real picture of the St. Basil cathedral. We provide coherent reflectance (compare our result to Figure 5.12, top row), which was not in the scope of single-image approaches. Our reflectance layer is comparable in quality to the best previous work. Enforcing coherent reflectance across views results in some residual color in the illumination layer; this is discussed in Section 5.4.2. However, color residuals are attenuated in other views (see Figure 5.12, top row). Also, note that the illumination is completely monochromatic in the results of some of the other methods.

Input images Reflectance Illumination

Figure 5.12: *Results of our method on internet photocollections. First row: another view of the StBasil scene. The reflectance we extract is coherent with the one shown in Figure 5.13. Second row: Knossos. Third row: RizziHaus. The thin specular objects which cast shadows on the façade are a challenging case for multi-view stereo. Our method is able to extract their shadows despite the lack of a complete and accurate 3D reconstruction. Bottom row: Moldovita. Our decomposition successfully separates the complex paintings and recovers a smooth illumination.*



Input and scribbles for [Bousseau 2009] | [Bousseau 2009] | [Shen 2008] | [Tappen 2005] | [Shen 2011a]



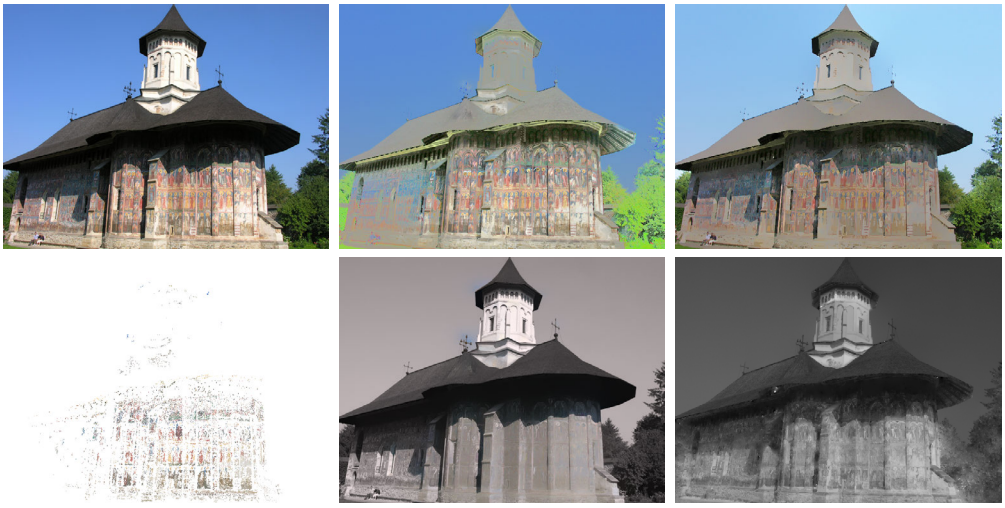
[Garces 2012] | [Gehler 2011] | [Shen 2011b] | [Zhao 2012] | Our decomposition

Figure 5.13: *Comparison between our approach and existing single-image methods on a picture from an online collection. For each decomposition, the top row contains the reflectance and the bottom row shows the illumination. Note that the method by Bousseau et al. [Bousseau 2009] is user-assisted. All results have been provided by the respective authors or come from [Garces 2012].*

Comparison to single-image methods. In Figure 5.14b, our decomposition assigns a similar reflectance to the steeple and roof of the monastery because very few 3D points are reconstructed in these areas (see Figure 5.14a, bottom). In the absence of 3D points, the decomposition lacks pairwise and coherence constraints to disambiguate reflectance and illumination and relies mostly on the smoothness prior. In contrast, our algorithm successfully disambiguates the complex texture on the lower facade where sparse 3D information is available. Additional pictures of the steeple and roof can be used to improve the 3D reconstruction and our decomposition.

A common assumption of single image methods is to constrain pixels with similar chrominance to share similar reflectance, which is likely to produce the same greyish reflectance as ours over the white steeple and dark roof. For example, the method by Zhao et al. [Zhao 2012] produces a similar greyish reflectance as ours on the steeple and roof, while their reflectance contains residual blue shadows and their illumination contains residual texture on the facade (Figure 5.14c). In addition, while this assumption on chrominance works well in many situations, it fails in the presence of colored lighting. Our method does not rely on such assumption and handles mixed lighting conditions well; compared to existing approaches, our method leverages geometric and multi-lighting information to disambiguate complex reflectance-illumination patterns.

Comparison to an user-assisted approach. In Figure 5.15, we show additional comparisons to the user-assisted approach of Bousseau et al. [Bousseau 2009]. Our automatic approach produces comparable results, and superior results in cases such as the painted façade of Moldovita (first row) or the underexposed shadow of Florence (third row).



(a) Input image and constrained points (b) Our decomposition (c) Result from [Zhao 2012]

Figure 5.14: *Comparison to a single-image method on the Moldovita scene. Our approach successfully separates the complex painted texture from the smooth illumination (b), in regions which are well reconstructed (a). In the absence of 3D points (e.g., steeple and rool, left part of the façade), our decomposition relies on the image-guided smoothness prior. In comparison, the method by Zhao et al. [Zhao 2012] shares similar artifacts on the steeple and roof due to their assumption on chrominance, but does not extract the shadow cast on the façade (c).*

Scribbles for [Bousseau 2009] | [Bousseau 2009] reflectance | [Bousseau 2009] illumination | Our reflectance | Our illumination

Figure 5.15: *Comparison to a user-assisted approach on three scenes: Moldovita, Manarola, Florence. Input scribbles were kindly provided by the authors. The coherence constraints ensure that the reflectance is similar in every view and allows the recovery of reflectance values even in shadowed areas where the single image approach of Bousseau et al. [Bousseau 2009] produces noisy results. In addition, we recover a smoother illumination in textured planar regions.*

5.4.2 Analysis and Limitations

5.4.2.1 Analysis

	Synth.	Captured		Internet Photo Collections					
Scene	1	2	3	4	5	6	7	8	9
N_{d}	6	5	10	9	11	8	11	8	17
N_{r}	30	32	48	56	60	34	61	28	53
P_{rec}	100k	467k	1.3M	1M	888k	2.0M	1.4M	591k	552k
P_{sel}	68k	200k	199k	200k	196k	192k	199k	200k	196k
C_{cand}	1.4M	1.5M	1.5M	1.3M	1.4M	1.3M	1.5M	1.5M	1.5M
C_{pair}	260k	724k	709k	197k	392k	241k	155k	272k	192k
C_{coher}	39k	105k	142k	66k	65k	57k	46k	54k	53k

Table 5.2: *N_d Number of images to decompose for each scene, N_r number of images for reflectance ratio estimation (Equation 5.4), P_{rec} number of reconstructed 3D points, P_{sel} number of points after downsampling, C_{cand} number of candidate pairs for reflectance constraints before applying the statistical criterion, C_{pair} number of reliable pairwise constraints, C_{coher} number of coherency constraints. 1: Synthetic St. Basil; 2: Doll; 3: Temple; 4: St. Basil; 5: Knossos; 6: Moldovita; 7: Florence; 8: RizziHaus; 9: Manarola.*

We show the number of constraints estimated for each scene in Table 5.2. The size of the downsampled point cloud P_{sel} and the number of candidate pairs for reflectance constraints C_{cand} are approximately the same for all captured and downloaded scenes. However, on average 52% of the pairs are discarded for captured scenes, and 83% for downloaded scenes. Moving from a single-camera, controlled capture setting to online photo collections introduces errors due to different cameras, temporal extent (e.g., repainted façades), and image editing. Our robust statistical criterion detects some of these errors and discards the corresponding pairs.

Influence of pairwise constraints. Figure 5.16 illustrates the importance of our pairwise constraints for disambiguating reflectance and illumination. In regions with complex texture, they allow us to recover smooth illumination (Figure 5.16b). Disabling these constraints introduces strong texture residuals in the illumination layer (Figure 5.16a).

Influence of coherency constraints. In Figure 5.17, we first show the decomposition for a single image without the coherence term $E_{\text{coherence}}$, and then the result with coherence constraints to all other images. This image is hard to decompose since it contains mixed lighting conditions, i.e., the blue sky is dominant in the shadow while the bright sun is dominant elsewhere. As a result, the reflectance without coherence constraints contains a residual shadow, which is removed when coherence constraints are added. In addition, enforcing a coherent reflectance across images enables new applications which combine multiple views, such as model texturing (Section 5.5.2) and image-based illumination transfer (Section 5.5.3).

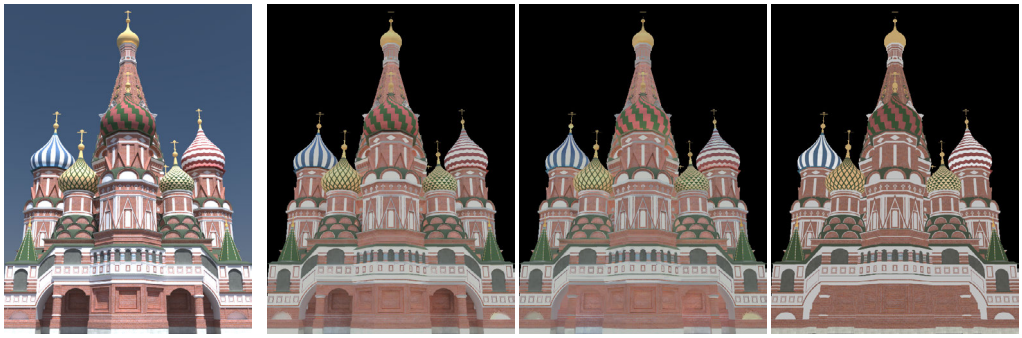
(a) Illumination without pairwise constraints (b) Illumination with pairwise constraints

(c) Input (d) Without pairwise constraints (e) With pairwise constraints

Figure 5.16: *Influence of the pairwise relative constraints on another image of the Doll scene. (a) Without pairwise reflectance constraints, texture cannot be successfully separated from lighting and the resulting illumination layer contains large texture variations. (b) Enabling these constraints allows us to recover a smooth illumination on the tablecloth, despite the complexity of its texture. Similar observations can be made on the painted façade of Moldovita (bottom).*

Input image Reflectance without coherence Reflectance with coherence

Figure 5.17: *Comparison between the decomposition, before and after multi-view coherence in the Florence scene. The coherence constraints between multiple views allow our method to recover a coherent reflectance even under mixed lighting conditions such as this bright sun with blue shadows.*



(a) Input rendering (b) Reflectance without correction (c) Reflectance with correction (d) Ground truth reflectance

Figure 5.18: *Effect of compensating for ambient occlusion on the decomposition of a synthetic image (a). Without special treatment, the reflectance under the arches appears darker (b) because these regions systematically receive less illumination. Correcting the pairwise reflectance constraints by compensating for ambient occlusion (Section 5.2.1) yields a reflectance (c) closer to ground truth (d).*

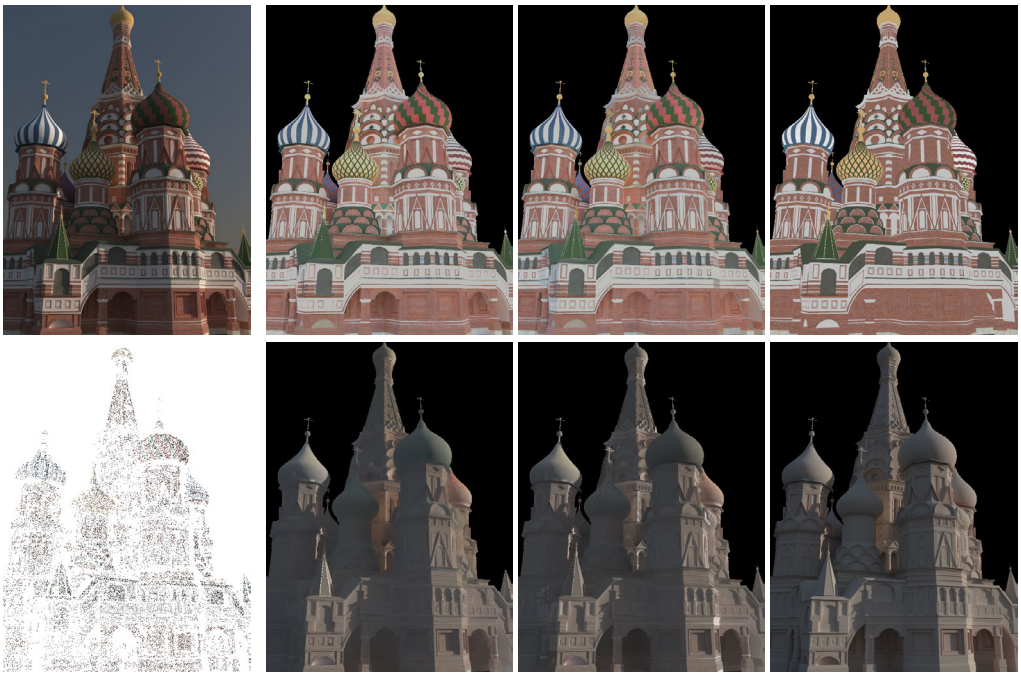
Influence of ambient occlusion. Figure 5.18 shows the effect of correcting pairwise reflectance constraints with the ratios of ambient occlusion (Section 5.2.1). This correction yields a better estimation of reflectance in regions which are systematically in shadow, such as the arches in the synthetic example. While we applied it on all scenes, this correction is optional; disabling it has little effect on scenes without concavities (such as RizziHaus, Figure 5.12, third row).

Influence of grayscale regularization. Figure 5.19 shows an example where the grayscale regularization term introduced in Section 5.3.2 improves the results of our decomposition, in highly occluded regions which are not well reconstructed.

Influence of smoothness prior. Bousseau et al. [Bousseau 2009] applied their linear model channel per channel to estimate colored illumination, which is critical to handle nighttime pictures and mixed sun/sky lighting. In Section 5.3.2, we relaxed this model by adding a per-window constant term, leading to an affine model similar to that of Levin et al. [Levin 2008]. We found that this prior captures well smooth illumination on complex textured surfaces, such as the tablecloth on the Doll scene. On such a scene, the prior from Bousseau et al. produces texture residuals in the illumination layer (Figure 5.20).

5.4.2.2 Limitations

We designed our method to estimate coherent reflectance over multiple views of a scene. However, images in photo collections are often captured with different cameras and can be post-processed with different gamma and saturation settings. Since we enforce coherent



Input image and constrained points | Without regularization | With regularization | Ground truth

Figure 5.19: *While our method produces a high quality decomposition in most regions of the image, adding the grayscale regularization further improves the results in regions with ambient occlusion. The regularization helps to capture the shadowing effects in areas where only few 3D points are reconstructed, such as the arches in the lower part of the cathedral.*

Input image | With prior of [Bousseau 2009] | With our smoothness prior

Figure 5.20: *Influence of the smoothing prior. We compare results of our decomposition using the image-guided prior of [Bousseau 2009] (middle), with ours based on the Matting prior of [Levin 2008]. Our prior better disambiguates texture from shading in complex regions and recovers a smoother illumination layer.*

reflectance, residues of these variations are sometimes visible in our illumination component, (e.g., Figure 5.13). We argue that *some* reflectance residues in the illumination are acceptable as long as reflectance is plausible and coherent. For example they will be recombined with a coherent (thus similar) reflectance layer when transferring lighting, in Section 5.5.3. An automated way to identify and correct for camera responses and image transformations is a promising direction for future work. We expect such corrections to remove the remaining artifacts in our intrinsic image decompositions.

We rely on multi-view stereo for correspondences between views. Consequently in poorly reconstructed regions (such as very dark regions; e.g., just below the roof in Figure 5.12, bottom), we rely only on the smoothness energy for our decomposition. Since no correspondences exist between views, reflectance in these regions is not coherent across images. If such regions are systematically darker in all views, this is fine for lighting transfer because low illumination values mask the reflectance discrepancy. However, since reflectance is computed by dividing the input image with illumination, very small illumination values can result as very bright pixels in the reflectance. Thin features are also problematic since radiance is blended in the input images. This could be treated with a change of scale, i.e., using close up photos.

5.5 Applications

5.5.1 Image editing

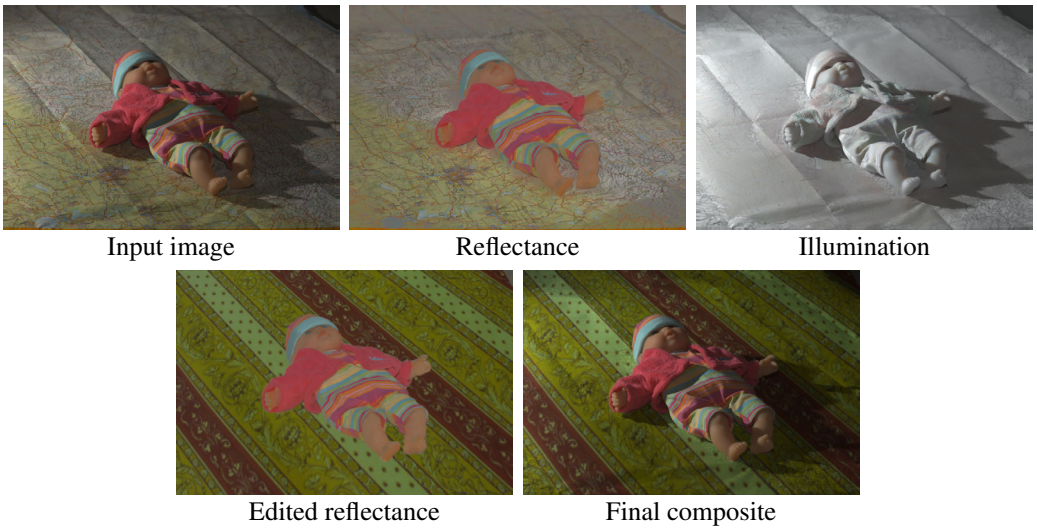
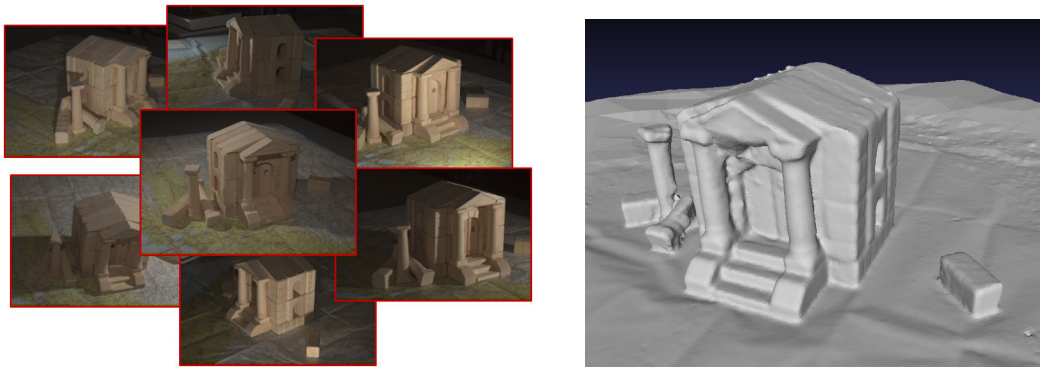


Figure 5.21: *We use our method to replace the texture of the tablecloth while preserving the illumination variations such as folds and shadows.*

Our intrinsic decomposition separates the effects of reflectance and illumination in photographs, thus allowing lighting-aware manipulations in image editing software. In Figure 5.21, we modify the reflectance layer and recombine it with the original illumination layer, resulting in an image with new material colors but coherent illumination.

Note that in this simple editing example, we ignored indirect lighting: altering the materials colors should affect the illumination on other parts of the scene, because of inter-reflections. Our automatic decomposition provides the input required in [Carroll 2011] to tackle this issue.

5.5.2 Texturing



(a) Input photographs (b) Reconstructed 3D mesh

(c) Textured with projected radiance (d) Textured with projected reflectance

Figure 5.22: *From a set of multi-view photographs (a), we automatically reconstruct a 3D mesh (b). Applying projective texture mapping with the input photographs yields a textured mesh which contains shadows from the original images (c). In contrast, applying texture mapping with the coherent reflectance obtained with our method yields an illumination-free model of the object (d). The resulting model contains no shading or shadows, and can be integrated into virtual environments.*

Our decomposition yields reflectance images which are coherent over all views of a scene. These images do not contain shading or shadows, and can be exploited to facilitate texturing and compositing. In Figure 5.22, we reconstruct a 3D model and automatically produce illumination-free textures using projective texture mapping.

5.5.3 Lighting transfer

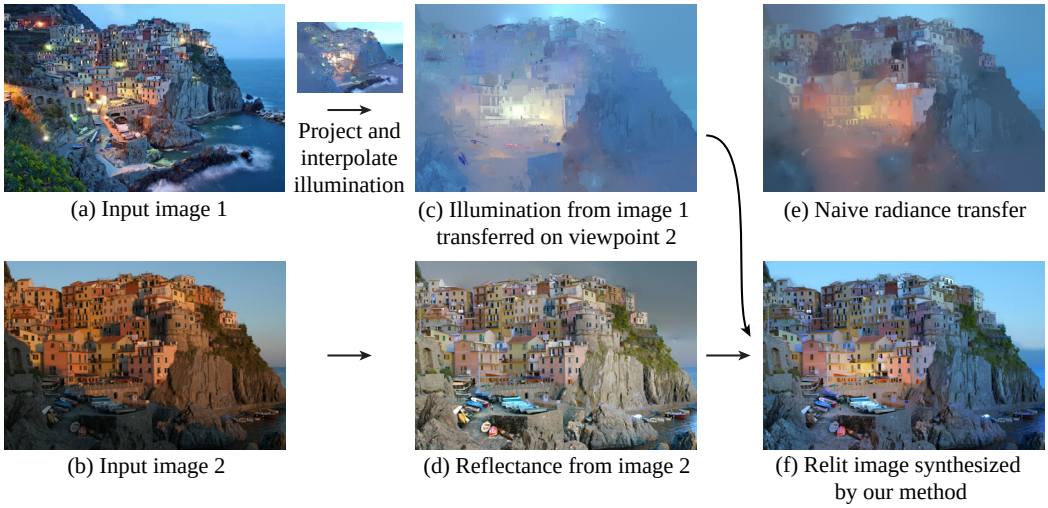


Figure 5.23: *Given two views of the same scene under different lighting conditions (a,b), we transfer the illumination from one view into the other view (c). We then multiply the transferred illumination by the reflectance layer (d) to synthesize the relit image (f). Transferring the radiance directly fails to preserve the fine details of the reflectance (e).*

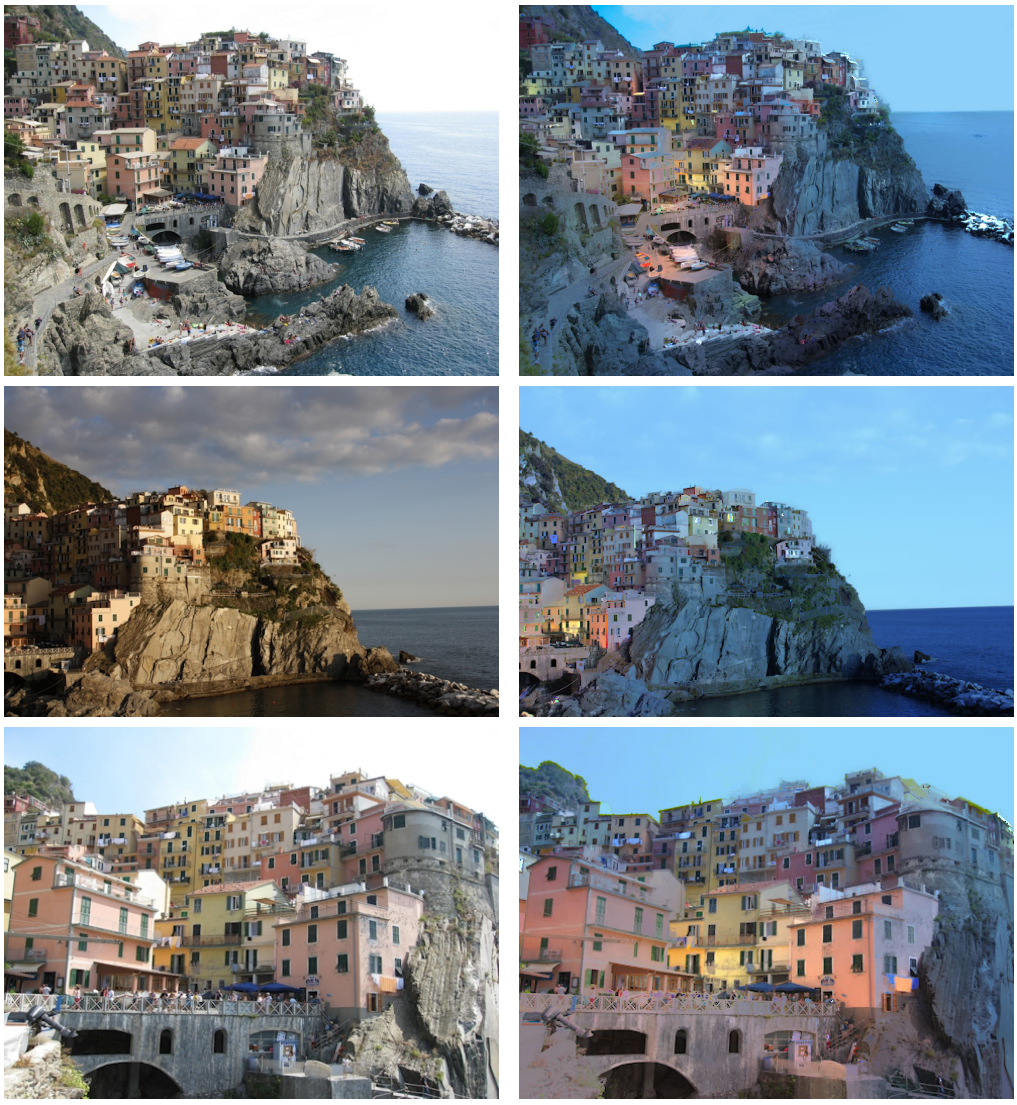
As a novel application of our multi-view decomposition, we transfer illumination between two pictures of a scene taken from different viewpoints under different lighting conditions. We use the 3D point cloud as a set of sparse correspondences for which the illumination is known in the two images. We then propagate the illumination of one image to the other image using the smoothness prior of Section 5.3.2. In areas visible only in the target view, the propagation interpolates the illumination values from the surrounding points visible in both images. We finally generate a radiance image by multiplying the reflectance layer with the transferred illumination layer. Since multi-view stereo does not produce 3D points in sky regions, we use the sky detector of Hoiem et al. [Hoiem 2005], correct the segmentation if necessary, and apply standard histogram transfer on sky pixels.

In Figure 5.23(e) we compare our illumination transfer with direct transfer of radiance. Propagating the radiance produces smooth color variations in-between the correspondences. In contrast, our combination of transferred illumination with the target reflectance preserves fine details.

We apply our approach to harmonize lighting for multiple viewpoints (figure 5.24), and produce illumination-coherent view transitions for applications such as Photo-Tourism [Snavely 2006]. In our accompanying video we show image-based view transitions [Roberts 2009] with harmonized photographs. Our method produces stable transitions between views, despite strong shadows in the original images that could not be handled by simple color compensation [Snavely 2008].

We also transfer all illumination conditions to a single viewpoint (Figure 5.25). This results in artificial images which show the scene from a fixed view, under a variety of different appearances, and allows us to create artificial *timelapse* sequences of a scene.

(a) Source lighting



(b) Source viewpoints (c) Relit images

Figure 5.24: *We use our lighting transfer to harmonize the illumination over multiple images.*

Figure 5.25: *We transfer the lighting of several input images to a single viewpoint. The variations in scene appearance can be assembled into an artificial "timelapse" sequence (best viewed in video).*

5.6 Conclusion

We introduced a method to compute coherent intrinsic image decompositions from photo collections. Such collections contain multiple lighting conditions and can be used to automatically calibrate camera viewpoints and reconstruct a 3D point cloud. We leverage this additional information to *automatically* compute coherent intrinsic decompositions over the different views in the collection. We demonstrated how sparse 3D information allows automatic correspondences to be established, and how multiple lighting conditions are effectively used to compute the decomposition. Our automatic solution shows that the use of coherence constraints can improve the extracted reflectance significantly, and that we can produce coherent reflectance even for images with extremely different lighting conditions, such as night and day.

We introduced a complex synthetic benchmark with ground truth, and compared our method to several previous approaches. Our approach outperforms previous methods numerically on the synthetic benchmark and is comparable visually in most cases. In addition, our method ensures that the reflectance layers are coherent among the images. We presented results on a total of 9 scenes and have automatically computed intrinsic image decompositions for a total of 85 images. We include those decompositions in the accompanying materials.

Our coherent intrinsic images enable lighting transfer and illumination-coherent transitions between views. It allows the creation of artificial images with combinations of viewpoint and lighting which did not exist in the original photo collection.

CHAPTER 6

Conclusion and Future Work

Lighting is a key element in all photographs, but is difficult to control in many scenes. Manipulating the lighting after pictures have been taken is a difficult task, which requires skilled artists. Our work has focused on extracting lighting from photographs to enable advanced image editing.

This thesis describes a new approach to tackling the intrinsic image decomposition problem, which aims to separate photographs into independent reflectance and illumination layers. We show that multiple views of the scene provide relevant information for guiding the decomposition. We leverage the geometric information provided by automatic 3D reconstruction in order to constrain the decomposition, and image-guided propagation algorithms to interpolate information in non-reconstructed or unreliable regions. We describe two classes of methods which exploit multiple photographs, either captured at a single time of day (Chapters 3 and 4) or gathered from photo collections with varying lighting (Chapter 5).

Decoupling the reflectance from the illumination enables advanced image editing, such as manipulating material colors while preserving coherent lighting, inserting virtual objects, and relighting photographs. We demonstrate such manipulations in image editing software, simply by modifying and recombining independent layers generated by our methods. Our multi-view decomposition further allows for constructing illumination-free textures for 3D models, and for transferring lighting between photographs of a scene. This opens the way for image-based applications which simplify digital content creation and manipulation.

Our work on multi-view intrinsic image decomposition opens a number of potential research directions. We classify these directions in two categories, based on their high-level goal. The first concentrates on the capture and simplifying the manipulation of scenes, while the second focuses on understanding and exploring the appearance of a given scene under real-world conditions.

6.1 Improving capture and manipulation of scenes

The ultimate goal is to make digital content creation as quick and as easy as possible, for casual users. This can involve capturing an object or scene with simple equipment,

modifying it with intuitive tools which do not require artistic training, and inserting it into virtual environments in a plausible manner. In future work, we would like to leverage the diverse and heterogeneous information which can be acquired about real-world scenes, in order to simplify their capture and manipulation.

The method presented in Chapter 3 proposes advanced image manipulations made possible by separating reflectance and illumination layers. However, such editing is performed in image space and independently for each view. In future work, we would like to pursue the development of solutions which will simplify advanced manipulation of *all the observations and representations of a scene at once*, rather than isolated images. In particular, edits could be transferred from one image to multiple captured views, to all frames of a dynamic video, and to 3D models assembled from the input images or scanned.

From multiple observations of a scene, a lot of useful information can be extracted about the geometry, the materials, and the lighting. While current state-of-the-art 3D reconstruction methods often yield incomplete and inaccurate geometry, combining the extracted 3D information with captured photographs enables interesting applications such as virtual tourism [Snavely 2006] or free-viewpoint navigation [Eisemann 2008, Chaurasia 2011]. Compared to existing image-based rendering techniques, our approach provides additional information, in the form of view-dependent intrinsic decompositions. In the future, we are interested in designing a new representation of captured scenes, which combines all the extracted data in a way suitable for image-based rendering techniques, and which enables relighting virtual environments.

The intrinsic image model exploited in this work assumes Lambertian surfaces (as shown in Section 2.3.1), which reflect the same amount of light in all directions. In order to handle the wide variety of materials present in real-world scenes, this model should be generalized to treat effects such as specular highlights and reflections. Photographs with multiple viewpoints capture the radiance from physical points of the scene towards different directions, and could be used to extract such view-dependent effects. An interesting direction for future work will be the extraction of Lambertian and view-dependent components of the reflectance, in the spirit of the lighting separation we proposed in Chapter 3.

Our separation of reflectance and illumination could potentially facilitate tasks which rely on color constancy, such as segmenting images or matching photographs with drastically different lighting conditions. In addition, the illumination layer captures subtle variations of shading which are due to local orientation changes. This can be used to extract geometric details, as shown in recent work [Luo 2012]. A 3D reconstruction pipeline could re-inject the result of intrinsic image decomposition into the pipeline in order to refine the estimated geometry.

We have shown in this thesis that knowledge of partial geometry, reconstructed with multi-view stereo, can help with the problem of intrinsic decomposition. Recent advances in computer vision and machine learning, as well as new capture devices such as cheap depth sensors (e.g., Kinect) and light-field cameras, now provide additional information

about the scene. An interesting future challenge will consist in identifying which problems can be simplified thanks to this extra information, and in designing robust algorithms to compensate for the unreliability of captured data.

6.2 Exploring the space of scene appearance

Community photo and video collections depict scenes under a variety of viewpoints, lighting and weather conditions, at different periods of time, and with various imaging devices. Taken together, these photos and videos span *the space of Scene Appearance*, defined as the space of images of a given scene under real-world conditions. Garg et al. studied the dimensionality of this space under a few limiting assumptions [Garg 2009]. So far, existing work focused on 3D navigation to explore this space: Snavely et al. proposed transitions between images based on a sparse 3D model of the scene [Snavely 2006], and chose to browse photographs with similar illumination [Snavely 2008].

Based on the multiple observations of the scene, i.e., sparse points in the large space of scene appearance, an interesting challenge will consist in developing new methods to synthesize plausible images in other regions of this space. As a first step, the work we described in Chapter 5 allows lighting transfer between images, by first decomposing input photographs into intrinsic images. This enables the exploration of a larger subspace of scene appearance, as the synthesized pictures exhibit viewpoint/lighting combinations which did not exist in the original collection. In future work, we would like to identify and transfer other properties which affect the appearance of the scene.

Figure 5.25 shows several images of a scene with different lighting conditions transferred to a single viewpoint, i.e., a *virtual timelapse* created with our approach. Even though the scene is static and the viewpoint is fixed, these images exhibit some variety in appearance and mood, due to the changes in lighting. Prior work has focused on defining *scene attributes* which describe the appearance of arbitrary scenes, based on their materials, surface properties, lighting, and functions [Patterson 2012]. In contrast, we would like to find attributes which discriminate the appearance of a given scene in real-world conditions. Such attributes will enable the classification of photographs based on their appearance and the identification of images of interest based on specific criteria. In addition, we will leverage these attributes to allow users to browse a photo collection based on intuitive controls which are mapped to meaningful directions in the space of scene appearance, such as named sliders which represent the degree of "sunniness" or the time of day.

In summary, this direction of future work aims to better understand and allow the exploration of the space of scene appearance. We will develop new image-based solutions which allow users to not only browse photographs from the original collection, but also easily visualize the scene with new combinations of plausible viewpoints, time, lighting and weather conditions. The ultimate goal is to allow users to freely navigate in a scene and manipulate its appearance, using a collection of photographs as input.

6.3 Concluding remarks

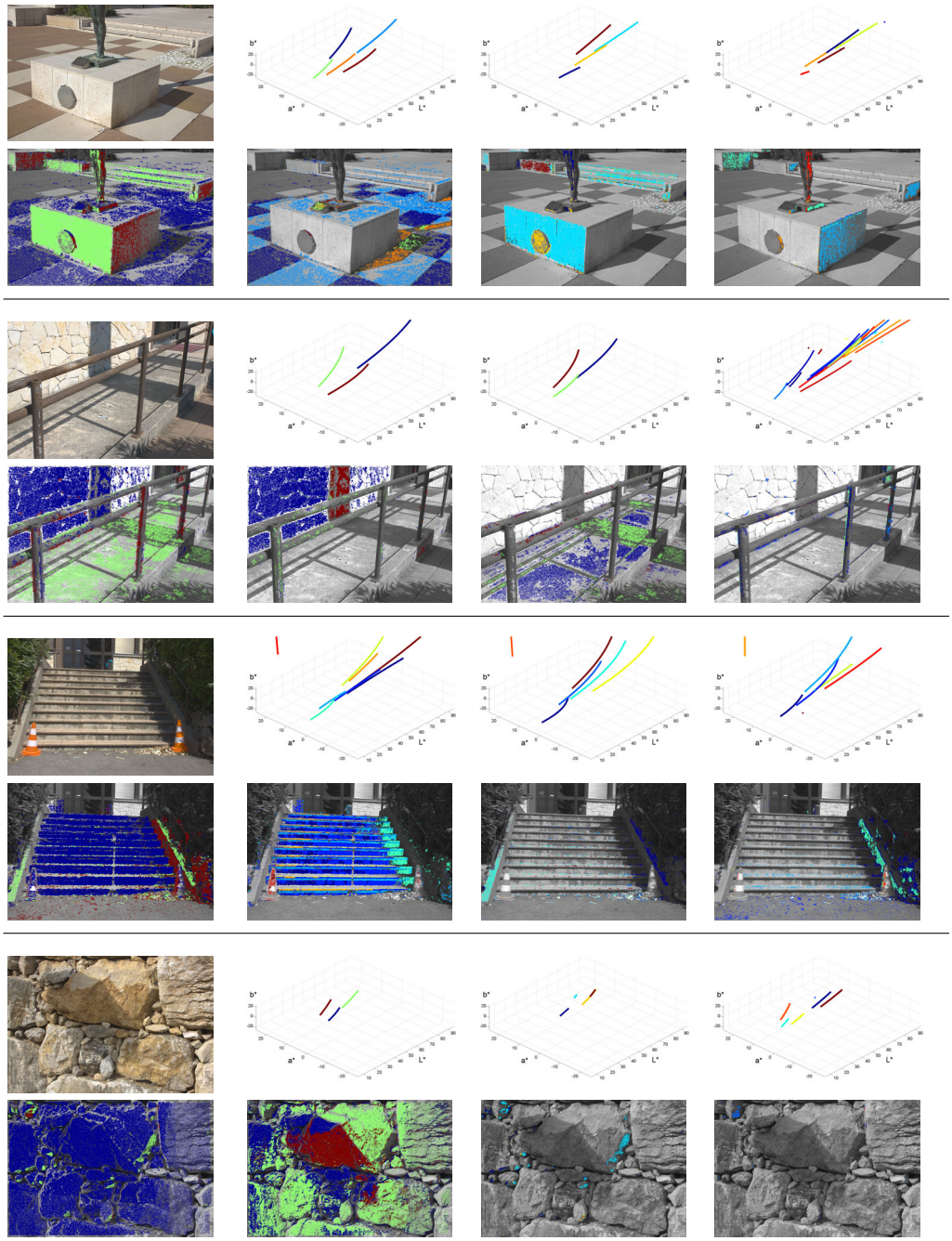
We have presented a new approach to extracting material and lighting information in a scene, using sparse geometry automatically reconstructed from multiple images. We hope that this thesis is a step forward towards empowering users with tools for advanced image editing, and manipulation of photo collections. Ultimately, our goal is to make digital content creation more accessible, which will increase the richness of virtual environments while enabling the manipulation of their appearance.

APPENDIX A

Appendix: Rich decomposition results

We show results of the rich intrinsic decomposition described in Chapter 3 on four scenes, for a total of 14 views. For each viewpoint, we provide the input (rectified) image, estimated reflectance, as well as the constraints and estimated total illumination, sun illumination, sky illumination and indirect illumination.

We also show examples of the candidate reflectance curves after clustering, in all of our scenes.



(a) Groups of normal orientation | (b) Curve clusters, orientation 1 | (c) Curve clusters, orientation 2 | (d) Curve clusters, orientation 3

Table A.1: *Clustering and candidate curves. PMVS points are separated into three groups according to the orientation of their normals (a, bottom). Within each group, PMVS points are clustered (b-d, bottom) based on their candidate reflectance curves. Each cluster is then represented by a single representative curve (b-d, top). The clustering parameters were fixed for all scenes.*

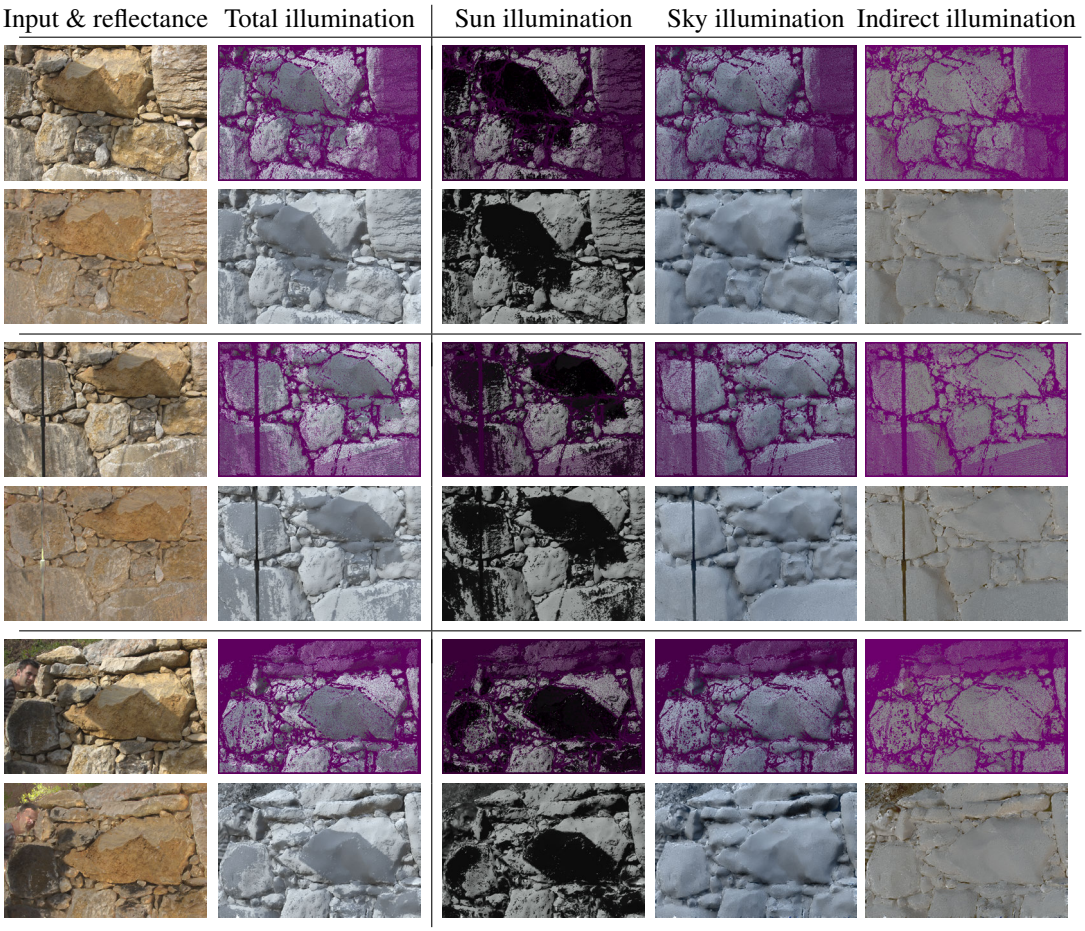


Table A.2: *Constrained pixels and results of our decomposition on the Rocks scene. This scene did not require inpainting after the illumination separation. The linear images have been scaled for display, before applying gamma correction with* $\gamma = 1/2.2$*; the scaling values were:* 0.7 *for the input,* 1.8 *for the reflectance,* 0.12 *for total illumination and sun illumination,* 0.2 *for sky illumination,* 0.5 *for indirect illumination.*

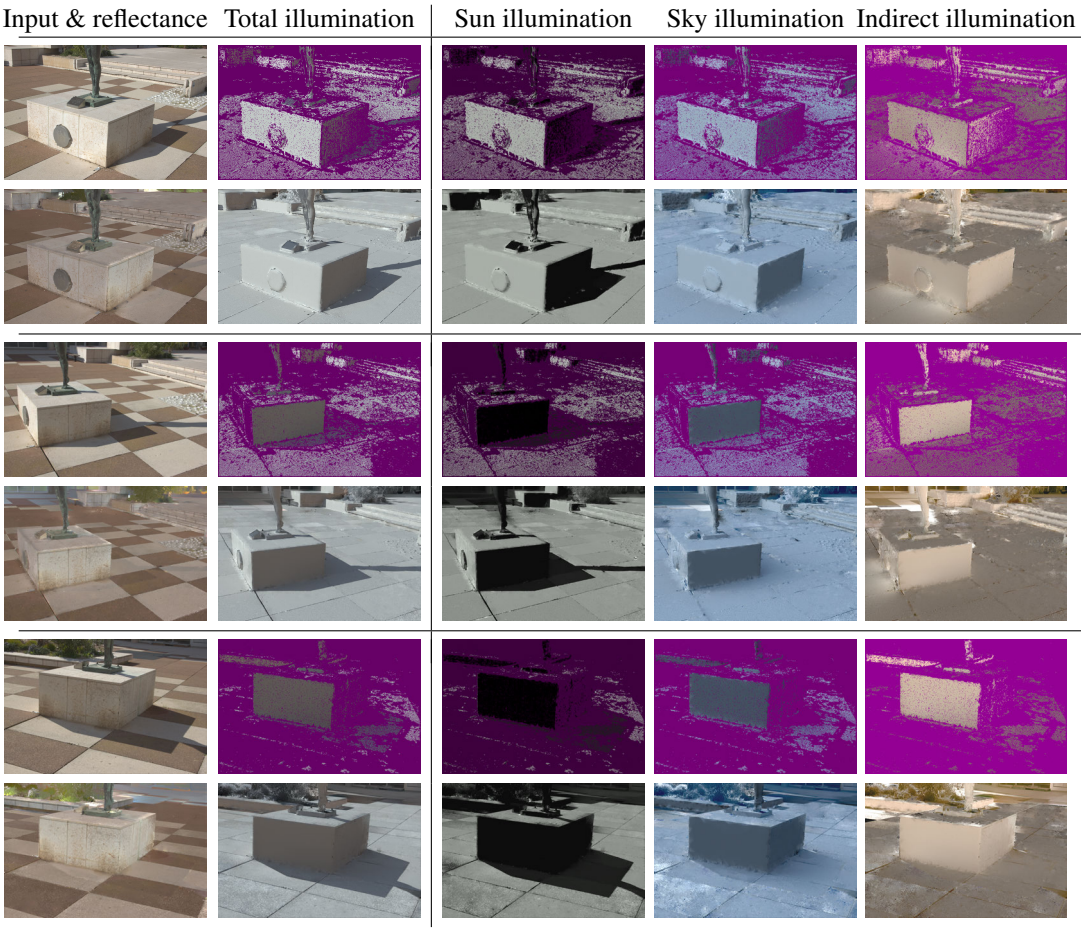


Table A.3: *Constrained pixels and results of our decomposition on the Statue scene after final inpainting. The linear images have been scaled for display, before applying gamma correction with* $\gamma = 1/2.2$*; the scaling values were:* 0.6 *for the input,* 1 *for the reflectance,* 0.15 *for total illumination and sun illumination,* 0.6 *for sky illumination,* 1 *for indirect illumination.*

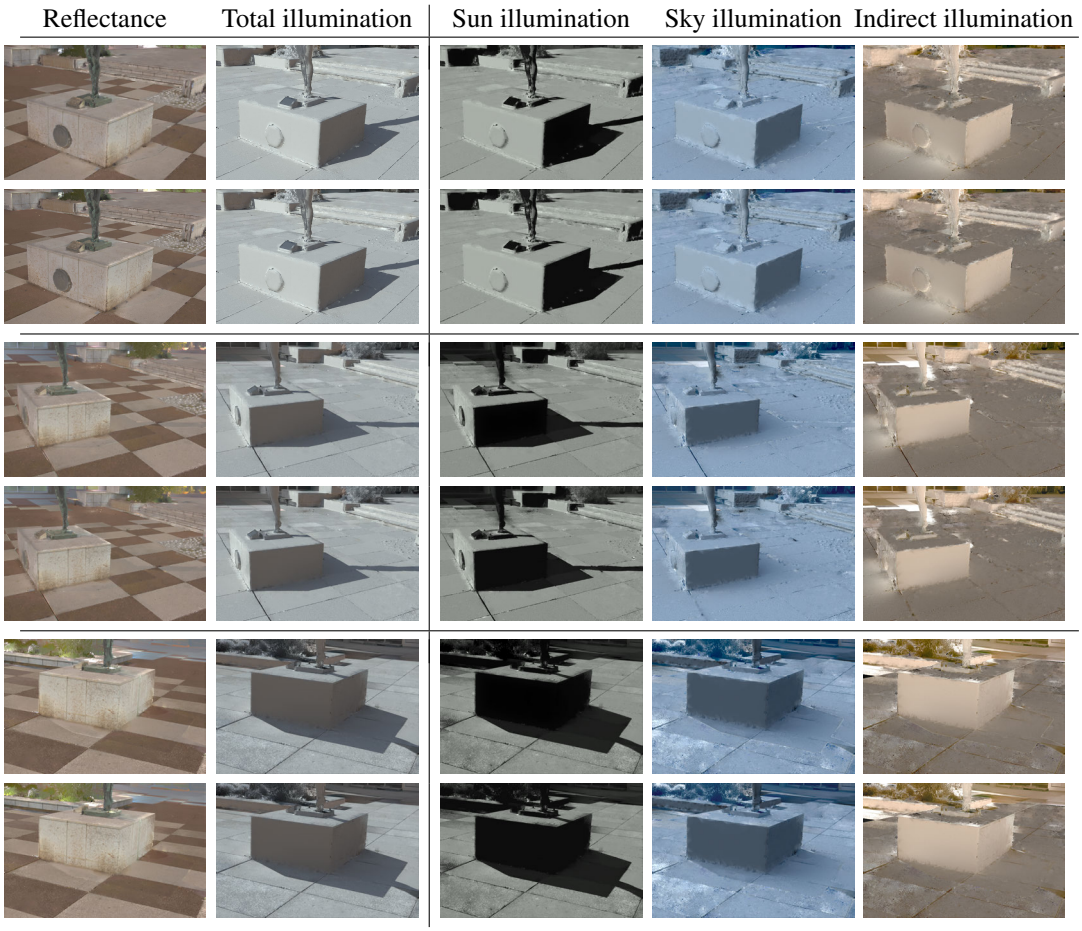


Table A.4: *Results of our decomposition on the Statue scene before (top) and after final inpainting (bottom). The linear images have been scaled for display, before applying gamma correction with* $\gamma = 1/2.2$*; the scaling values were:* 0.6 *for the input,* 1 *for the reflectance,* 0.15 *for total illumination and sun illumination,* 0.6 *for sky illumination,* 1 *for indirect illumination.*

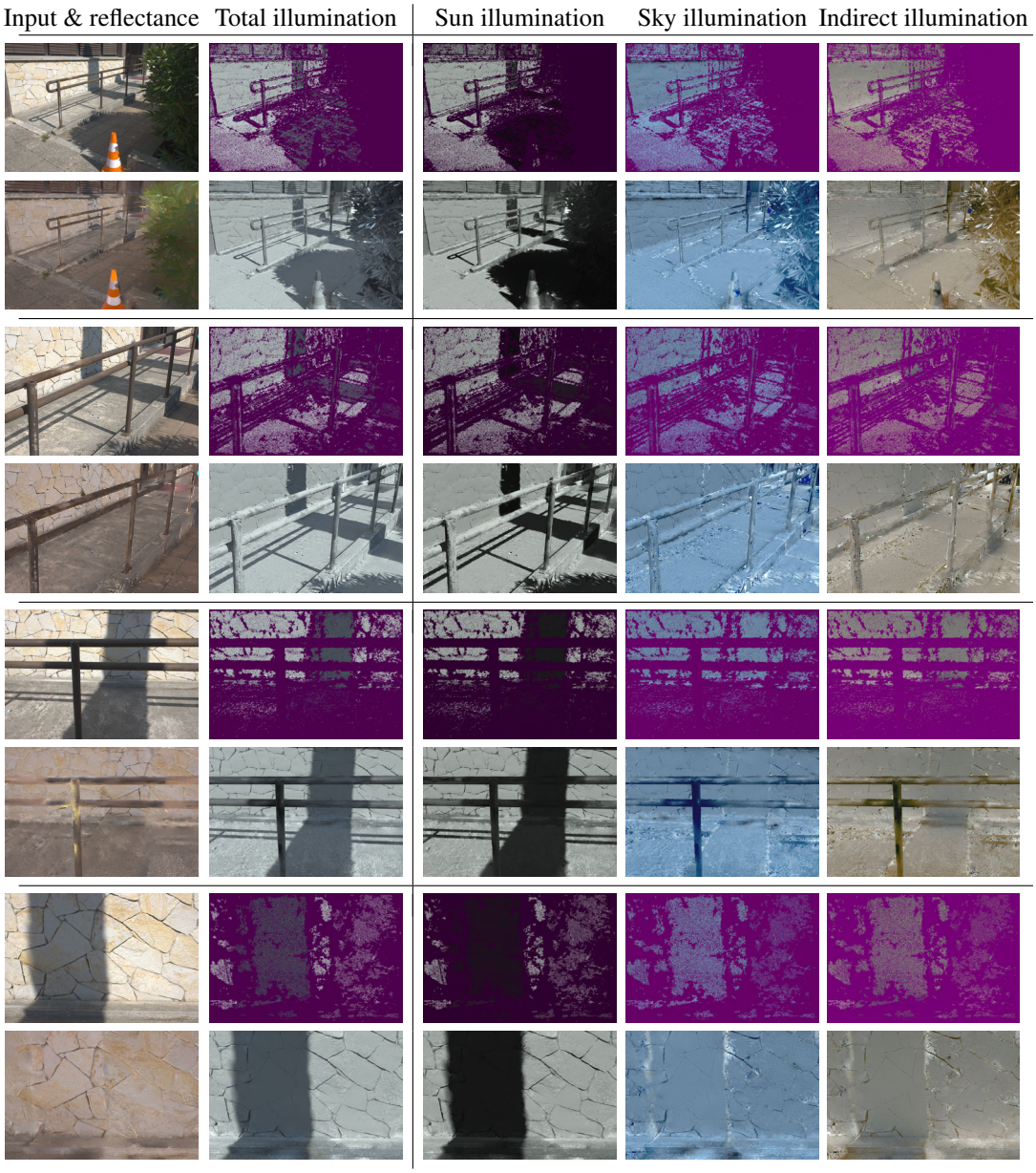


Table A.5: *Constrained pixels and results of our decomposition on the Ramp scene after final inpainting. The linear images have been scaled for display, before applying gamma correction with* $\gamma = 1/2.2$*; the scaling values were:* 0.4 *for the input,* 1 *for the reflectance,* 0.08 *for total illumination and sun illumination,* 0.45 *for sky illumination,* 0.6 *for indirect illumination.*

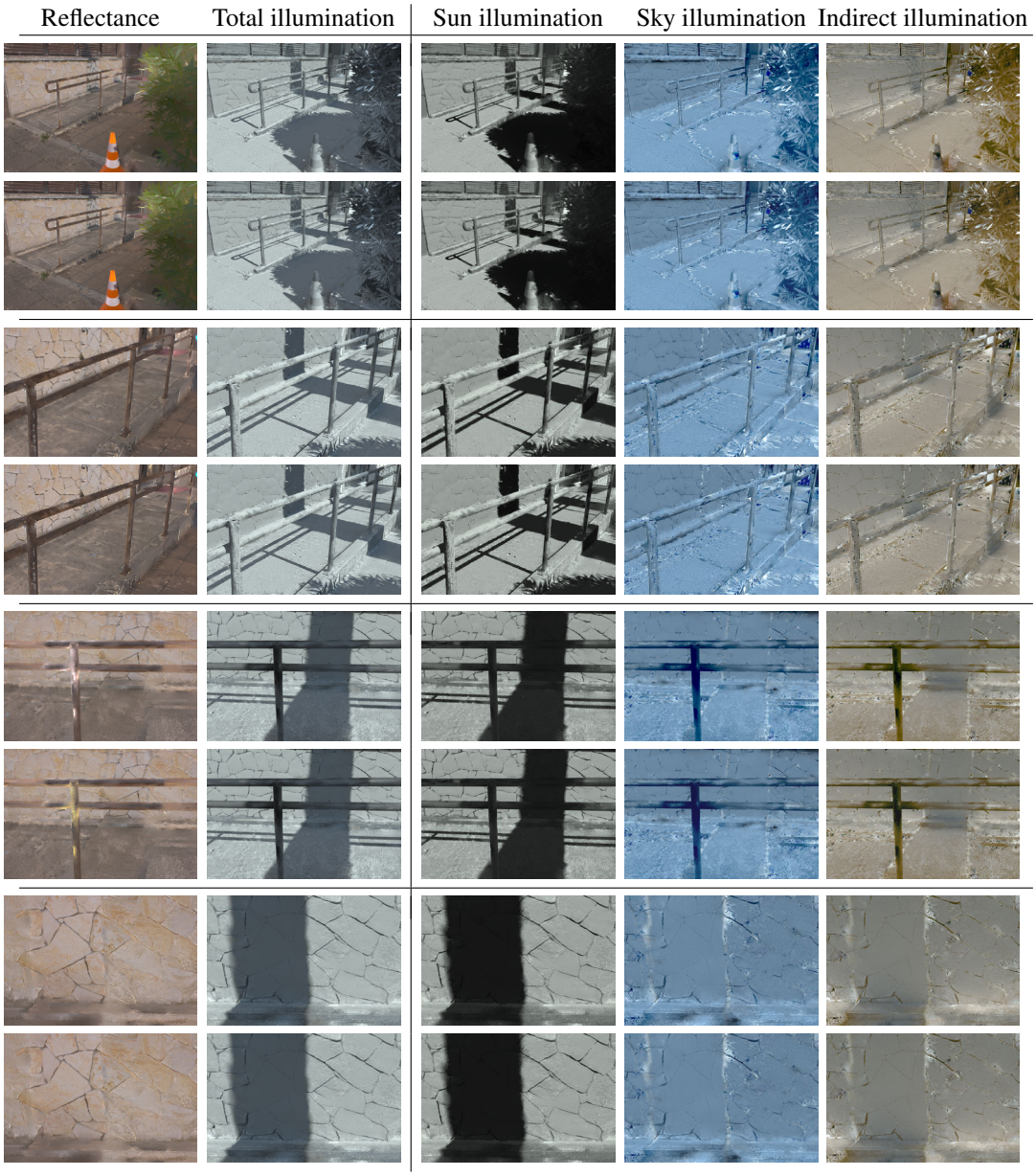


Table A.6: *Results of our decomposition on the Ramp scene before (top) and after final inpainting (bottom). The linear images have been scaled for display, before applying gamma correction with* $\gamma = 1/2.2$*; the scaling values were:* 0.4 *for the input,* 1 *for the reflectance,* 0.08 *for total illumination and sun illumination,* 0.45 *for sky illumination,* 0.6 *for indirect illumination.*

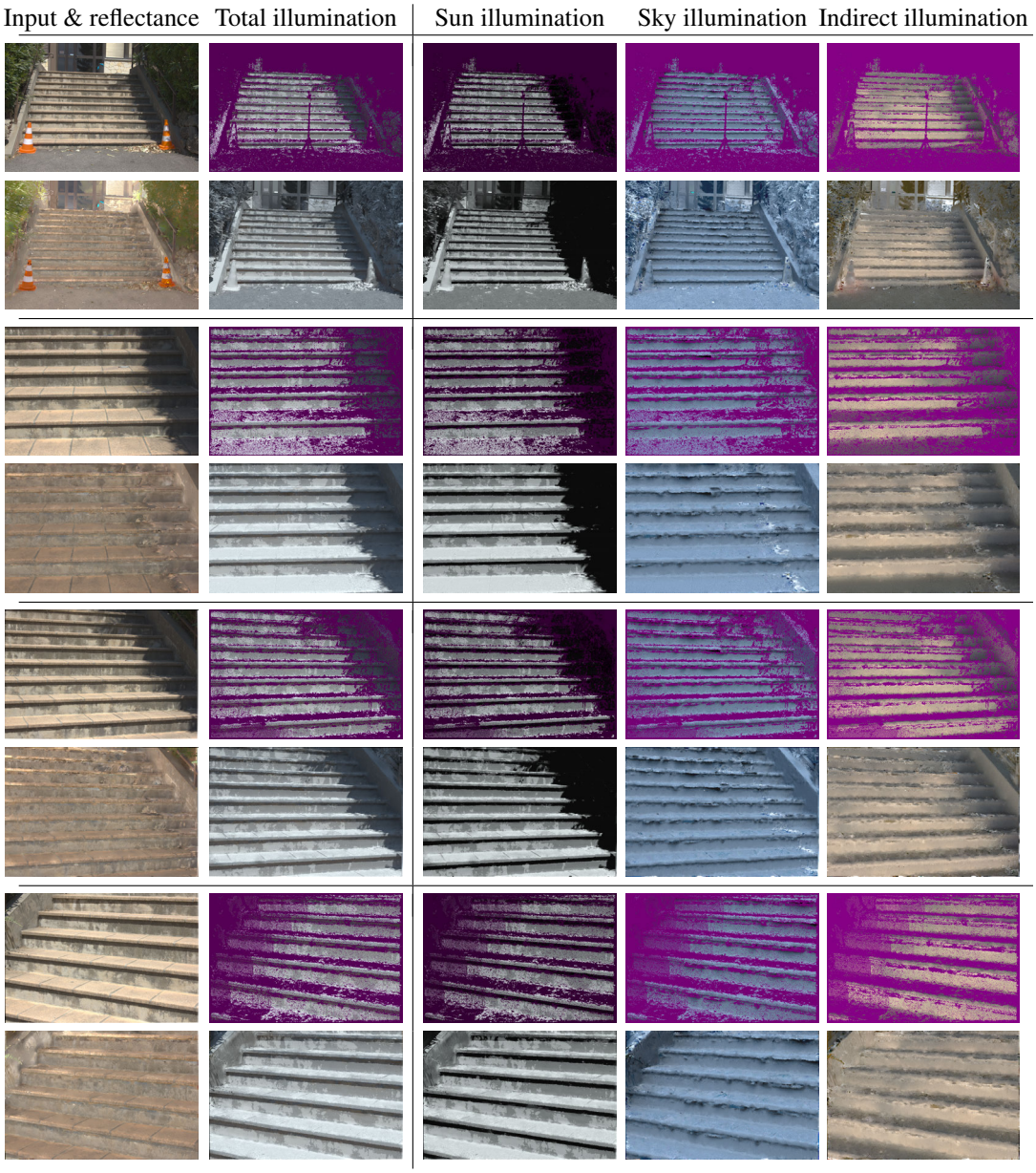


Table A.7: *Constrained pixels and results of our decomposition on the Stairs scene after final inpainting. The linear images have been scaled for display, before applying gamma correction with* $\gamma = 1/2.2$*; the scaling values were:* 0.5 *for the input,* 1.5 *for the reflectance,* 0.1 *for total illumination and sun illumination,* 0.6 *for sky illumination,* 0.8 *for indirect illumination.*

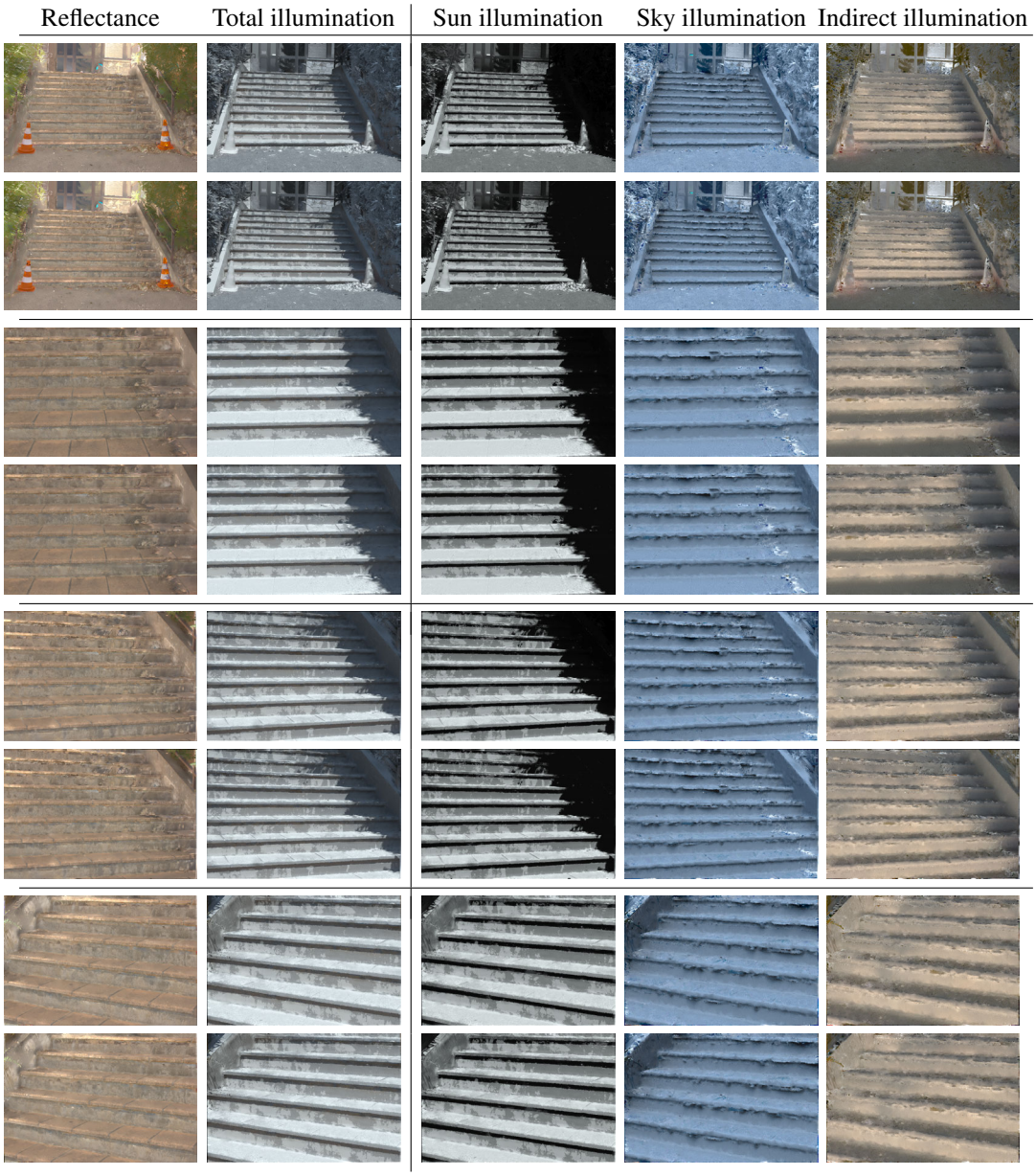


Table A.8: *Results of our decomposition on the Stairs scene before (top) and after final inpainting (bottom). The linear images have been scaled for display, before applying gamma correction with* $\gamma = 1/2.2$*; the scaling values were:* 0.5 *for the input,* 1.5 *for the reflectance,* 0.1 *for total illumination and sun illumination,* 0.6 *for sky illumination,* 0.8 *for indirect illumination.*

APPENDIX B

Appendix: Description of accompanying materials

This thesis contains accompanying materials in the form of videos, source code, and image datasets. We provide a description of these materials in this appendix. All the accompanying materials can be accessed online by visiting the author's website:

http://thesis.py-laffont.info

B.1 Accompanying materials for Chapter 3

Accompanying video. The rich intrinsic image decomposition method described in Chapter 3 can separate an input photograph into reflectance, sun illumination, sky illumination, and indirect illumination layers. This separation allows advanced manipulations in image editing software (Section 3.6.3). The accompanying video shows examples of such manipulations in Adobe Photoshop CS4.

B.2 Accompanying materials for Chapter 5

Sampling code. We provide reference Matlab code for the sampling algorithm described in Section 5.2.2. This source code selects candidate pairs in a synthetic example on a 2D point cloud, and was used to generate Figure 5.3.

Synthetic dataset. We release the synthetic dataset described in Section 5.4.1.1. We provide the rendering and ground truth images that we generated, and a quantitative evaluation of the results of previous approaches (kindly provided by the respective authors).

Intrinsic decompositions. We provide the results of our coherent intrinsic decomposition on 9 different scenes.

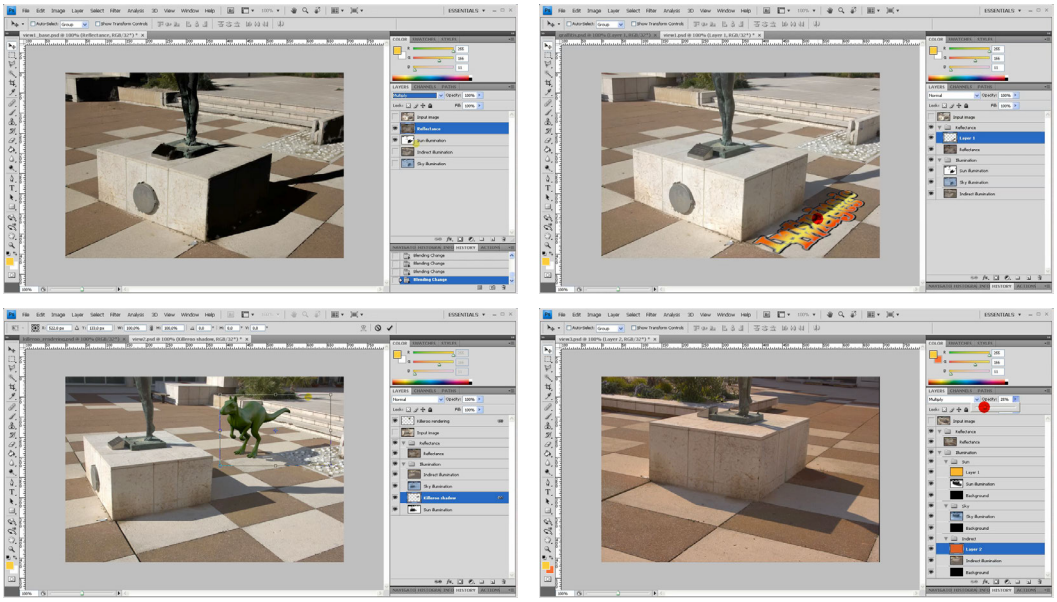


Figure B.1: *Captures of the accompanying video for Chapter 3.*

Figure B.2: *Capture of a page containing some of the results obtained with the coherent intrinsic image decomposition of Chapter 5.*

Accompanying video. The coherent intrinsic image decomposition described in Chapter 5 enables the transfer of lighting across images of a photo collection (Section 5.5.3). This video shows examples of image-based view transitions with harmonized photographs on two scenes: Florence and Manarola. We also transfer all illumination conditions to a single viewpoint, creating artificial *timelapses*.

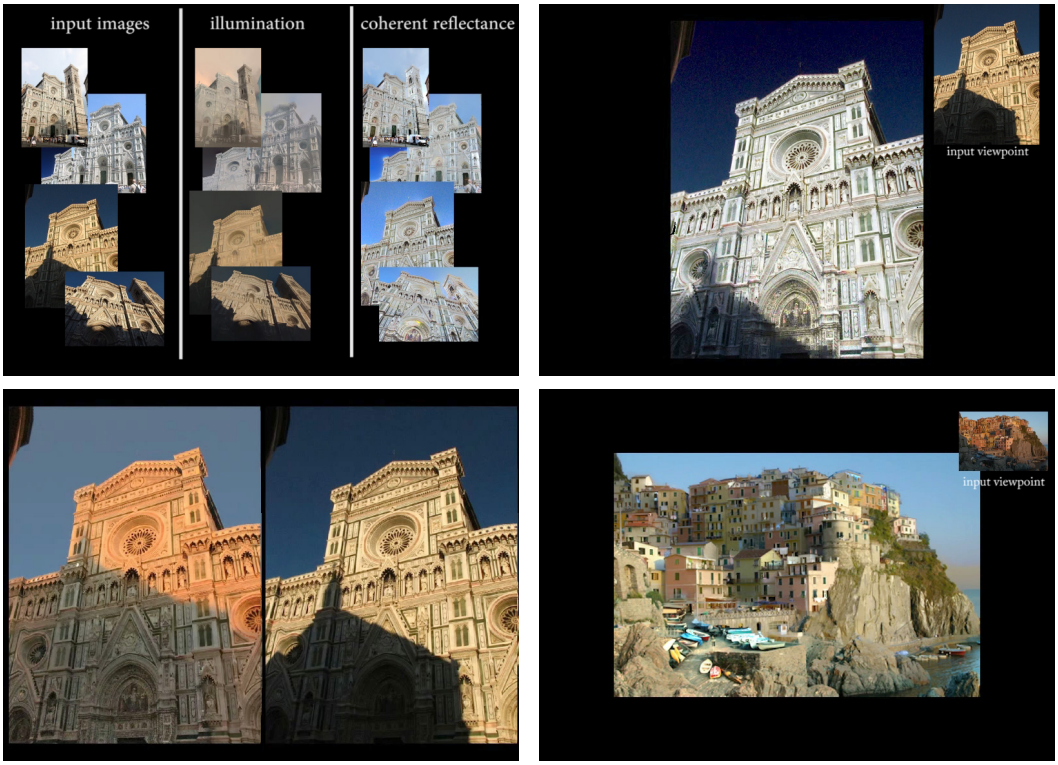


Figure B.3: *Captures of the accompanying video for Chapter 5.*

Bibliography

Personal publications

[Bazin 2010] Jean-Charles Bazin, **Pierre-Yves Laffont**, In So Kweon, Cédric Demonceaux and Pascal Vasseur. *An Original Approach For Automatic Plane Extraction By Omnidirectional Vision*. In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010), Taipei, Taiwan, October 18-22 2010. (Not cited.)

[Bosch 2011] Carles Bosch, **Pierre-Yves Laffont**, Holly Rushmeier, Julie Dorsey and George Drettakis. *Image-guided weathering: A new approach applied to flow phenomena*. ACM Transactions on Graphics, vol. 30, no. 20, 2011. **Presented at SIGGRAPH 2011**, Vancouver. (Not cited.)

[Laffont 2010] **Pierre-Yves Laffont**, Jong Yun Jun, Christian Wolf, Yu-Wing Tai, Khalid Idrissi, George Drettakis and Sung-eui Yoon. *Interactive content-aware zooming*. In Proc. Graphics Interface (GI 2010), pages 79–87, Ottawa, Ontario, Canada, 2010. (Not cited.)

[Laffont 2011] **Pierre-Yves Laffont**, Adrien Bousseau and George Drettakis. *Images intrinsèques de scènes en extérieur à partir de multiples vues*. REFIG (Revue Electronique Francophone d'Informatique Graphique), vol. 5, no. 2, pages 57–66, 2011. In French. Presented at *Journées de l'AFIG 2011*, **best paper award**. (Cited on page 23.)

[Laffont 2012a] **Pierre-Yves Laffont**, Adrien Bousseau and George Drettakis. *Rich Intrinsic Image Decomposition of Outdoor Scenes from Multiple Views*. IEEE Transactions on Visualization and Computer Graphics, vol. in press, 2012. **Presented at SIGGRAPH 2012**, Los Angeles (Poster and Talk sessions). (Cited on page 23.)

[Laffont 2012b] **Pierre-Yves Laffont**, Adrien Bousseau, Sylvain Paris, Frédo Durand and George Drettakis. *Coherent intrinsic images from photo collections*. ACM Transactions on Graphics (proc. of SIGGRAPH Asia), vol. 31, no. 6, 2012. **Presented at SIGGRAPH Asia 2012**, Singapore. (Cited on page 71.)

[Vangorp 2011] Peter Vangorp, Gaurav Chaurasia, **Pierre-Yves Laffont**, Roland Fleming and George Drettakis. *Perception of Visual Artifacts in Image-Based Rendering of Façades*. Computer Graphics Forum (proc. of Eurographics Symposium on Rendering 2011), vol. 30, no. 4, pages 1241–1250, July 2011. (Not cited.)

References

[Arbel 2011] Eli Arbel and Hagit Hel-Or. *Shadow Removal Using Intensity Surfaces and Texture Anchor Points*. IEEE Trans. PAMI, vol. 33, no. 6, pages 1202–1216, 2011. (Cited on page 15.)

[Barron 2012] Jonathan T Barron and Jitendra Malik. *Shape, Albedo, and Illumination from a Single Image of an Unknown Object*. In CVPR, 2012. (Cited on page 14.)

[Barrow 1978] Harry G Barrow and J Martin Tenenbaum. *Recovering intrinsic scene characteristics from images*. Computer Vision Systems, vol. 3, pages 3–26, 1978. (Cited on page 4.)

[Beigpour 2011] Shida Beigpour and Joost van de Weijer. *Object recoloring based on intrinsic image estimation*. In ICCV, pages 327–334, 2011. (Cited on page 20.)

[Bell 2001] Matt Bell and William T. Freeman. *Learning local evidence for shading and reflectance*. In ICCV, pages 670–677, 2001. (Cited on page 13.)

[Boivin 2001] Samuel Boivin and Andre Gagalowicz. *Image-based rendering of diffuse, specular and glossy surfaces from a single image*. In SIGGRAPH, pages 107–116, 2001. (Cited on page 10.)

[Bousseau 2009] Adrien Bousseau, Sylvain Paris and Frédo Durand. *User-assisted intrinsic images*. ACM Transactions on Graphics (proc. of SIGGRAPH Asia), vol. 28, no. 5, 2009. (Cited on pages 15, 16, 19, 20, 23, 40, 41, 45, 46, 71, 74, 81, 87, 89, 92, 95, 96, 97, 100 and 101.)

[Brooks 1985] M.J. Brooks and B.K.P. Horn. *Shape and source from shading*. In Proc. Int. Joint Conf. Artificial Intell., pages 932–936, 1985. (Cited on page 52.)

[Canny 1986] John Canny. *A Computational Approach to Edge Detection*. IEEE Trans. PAMI, vol. 8, no. 6, pages 679–698, 1986. (Cited on page 56.)

[Carroll 2011] Robert Carroll, Ravi Ramamoorthi and Maneesh Agrawala. *Illumination Decomposition for Material Recoloring with Consistent Interreflections*. ACM Transactions on Graphics (proc. of SIGGRAPH), vol. 30, no. 4, 2011. (Cited on pages 20 and 103.)

[Chaurasia 2011] Gaurav Chaurasia, Olga Sorkine and George Drettakis. *Silhouette-Aware Warping for Image-Based Rendering*. Computer Graphics Forum (proc. of Eurographics Symposium on Rendering), vol. 30, no. 4, 2011. (Cited on pages 70 and 108.)

[Cignoni 2008] Paolo Cignoni, Massimiliano Corsini and Guido Ranzuglia. *MeshLab: an Open-Source 3D Mesh Processing System*. ERCIM News, no. 73, pages 45–46, 2008. (Cited on page 28.)

[Comaniciu 2000] D. Comaniciu, V. Ramesh and P. Meer. *Real-time tracking of non-rigid objects using mean shift*. In CVPR, pages 142 –149, 2000. (Cited on page 37.)

[Comaniciu 2002] D. Comaniciu and P. Meer. *Mean shift: a robust approach toward feature space analysis*. IEEE Trans. PAMI, vol. 24, no. 5, pages 603–619, 2002. (Cited on pages 23, 34, 37 and 38.)

[Debevec 1997] Paul Debevec and Jitendra Malik. *Recovering high dynamic range radiance maps from photographs*. In SIGGRAPH, pages 369–378, 1997. (Cited on pages 10, 27 and 29.)

[Debevec 1998] Paul Debevec. *Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography*. In SIGGRAPH, pages 189–198, 1998. (Cited on pages 10 and 52.)

[Debevec 2004] Paul Debevec, Chris Tchou, Andrew Gardner, Tim Hawkins, Charis Poullis, Jessi Stumpfel, Andrew Jones, Nathaniel Yun, Per Einarsson, Therese Lundgren, Marcos Fajardo and Philippe Martinez. *Estimating surface reflectance properties of a complex scene under captured natural illumination*. Technical report, USC Institute for Creative Technologies, 2004. (Cited on pages 10 and 27.)

[Digne 2011] Julie Digne, Jean-Michel Morel, Charyar-Mehdi Souzani and Claire Lartigue. *Scale Space Meshing of Raw Data Point Sets*. Computer Graphics Forum, vol. 6, no. 4, pages 1630–1642, 2011. (Cited on page 28.)

[Dong 2011] Yue Dong, Xin Tong, Fabio Pellacini and Baining Guo. *AppGen: interactive material modeling from a single image*. ACM Transactions on Graphics (proc. of SIGGRAPH Asia), vol. 30, no. 6, 2011. (Cited on page 16.)

[Eisemann 2008] Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson de Aguiar, Naveed Ahmed, Christian Theobalt and Anita Sellent. *Floating Textures*. Computer Graphics Forum (proc. of Eurographics), vol. 27, no. 2, pages 409–418, 2008. (Cited on page 108.)

[Fang 2004] Hui Fang and John C. Hart. *Textureshop: texture synthesis as a photograph editing tool*. ACM Transactions on Graphics (proc. of SIGGRAPH), vol. 23, no. 3, pages 354–359, 2004. (Cited on page 20.)

[Finlayson 2002] Graham D. Finlayson, Steven D. Hordley and Mark S. Drew. *Removing Shadows from Images*. In ECCV, 2002. (Cited on page 15.)

[Finlayson 2004] Graham D. Finlayson, Mark S. Drew and Cheng Lu. *Intrinsic Images by Entropy Minimization*. In ECCV, pages 582–595, 2004. (Cited on pages 15 and 43.)

[Fukunaga 1975] K Fukunaga and L Hostetler. *The estimation of the gradient of a density function, with applications in pattern recognition*. IEEE Transactions on Information Theory, vol. 21, no. 1, pages 32–40, 1975. (Cited on page 34.)

[Funt 1992] Brian V. Funt, Mark S. Drew and Michael Brockington. *Recovering Shading from Color Images*. In ECCV, pages 124–132, 1992. (Cited on page 13.)

[Furukawa 2009a] Y. Furukawa, B. Curless, S. M. Seitz and R. Szeliski. *Manhattan-world stereo*. In CVPR, pages 1422–1429, 2009. (Cited on page 28.)

[Furukawa 2009b] Yasutaka Furukawa and Jean Ponce. *Accurate, Dense, and Robust Multi-View Stereopsis*. IEEE Trans. PAMI, vol. 32, no. 8, pages 1362–1376, 2009. (Cited on pages 21, 28, 72 and 83.)

[Gallup 2010] D. Gallup, J.-M. Frahm and M. Pollefeys. *Piecewise planar and non-planar stereo for urban scene reconstruction*. In CVPR, pages 1418–1425, 2010. (Cited on page 28.)

[Garces 2012] Elena Garces, Adolfo Munoz, Jorge Lopez-Moreno and Diego Gutierrez. *Intrinsic Images by Clustering*. Computer Graphics Forum (Proc. EGSR), vol. 31, no. 4, 2012. (Cited on pages 14, 87, 89 and 95.)

[Garg 2009] Rahul Garg, Hao Du, Steven M. Seitz and Noah Snavely. *The dimensionality of scene appearance*. In ICCV, pages 1917–1924, 2009. (Cited on page 109.)

[Gehler 2011] Peter Gehler, Carsten Rother, Martin Kiefel, Lumin Zhang and Bernhard Schölkopf. *Recovering Intrinsic Images with a Global Sparsity Prior on Reflectance*. In Advances in Neural Information Porcessing Systems (NIPS), pages 765–773, 2011. (Cited on pages 14 and 95.)

[Grosse 2009] Roger Grosse, Micah K. Johnson, Edward H. Adelson and William T. Freeman. *Ground-truth dataset and baseline evaluations for intrinsic image algorithms*. In ICCV, 2009. (Cited on pages 19 and 86.)

[Guo 2011] Ruiqi Guo, Qieyun Dai and Derek Hoiem. *Single-image shadow detection and removal using paired regions*. In CVPR, pages 2033–2040, 2011. (Cited on pages 15, 55 and 69.)

[Haber 2009] T. Haber, C. Fuchs, P. Bekaer, H.-P. Seidel, M. Goesele and H.P.A. Lensch. *Relighting objects from image collections*. In CVPR, pages 627–634, 2009. (Cited on pages 11 and 53.)

[Hanrahan 1993] Pat Hanrahan. *Rendering concepts*. In Michael F. Cohen and John Wallace, editeurs, Radiosity and Realistic Image Synthesis, chapitre 2, pages 13–40. Academic Press Professional, Inc., 1993. (Cited on pages 9 and 12.)

[Hays 2007] James Hays and Alexei A Efros. *Scene Completion Using Millions of Photographs*. ACM Transactions on Graphics (proc. of SIGGRAPH), vol. 26, no. 3, 2007. (Cited on page 71.)

[Hoiem 2005] Derek Hoiem, Alexei A. Efros and Martial Hebert. *Automatic photo pop-up*. ACM Transactions on Graphics (proc. of SIGGRAPH), vol. 24, no. 3, pages 577–584, 2005. (Cited on pages 60 and 104.)

[Hong 1998] Lin Hong, Yifei Wan and Anil Jain. *Fingerprint Image Enhancement: Algorithm and Performance Evaluation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, pages 777–789, 1998. (Cited on page 57.)

[Hoppe 1992] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald and Werner Stuetzle. *Surface reconstruction from unorganized points*. Computer Graphics (proc. of SIGGRAPH), vol. 26, no. 2, pages 71–78, 1992. (Cited on pages 22, 28 and 83.)

[Horn 1974] Berthold K.P. Horn. *Determining lightness from an image*. Computer Graphics and Image Processing, vol. 3, no. 4, pages 277–299, 1974. (Cited on page 13.)

[Horn 1986] Berthold K. Horn. Robot vision. McGraw-Hill Higher Education, 1st édition, 1986. (Cited on pages 9 and 52.)

[Hsieh 2009] Sung-Hsien Hsieh, Chih-Wei Fang, Te-Hsun Wang, Chien-Hung Chu and Jenn-Jier James Lien. *Weighted map for reflectance and shading separation using a single image*. In ACCV, pages 85–95, 2009. (Cited on page 13.)

[Hsu 2008] Eugene Hsu, Tom Mertens, Sylvain Paris, Shai Avidan and Frédo Durand. *Light mixture estimation for spatially varying white balance*. ACM Transactions on Graphics (proc. of SIGGRAPH), vol. 27, no. 3, page 70, 2008. (Cited on pages 34 and 42.)

[Jiang 2010] Xiaoyue Jiang, Andrew Schofield and Jeremy Wyatt. *Correlation-Based Intrinsic Image Extraction from a Single Image*. In ECCV, pages 58–71, 2010. (Cited on page 14.)

[Karsch 2011] Kevin Karsch, Varsha Hedau, David Forsyth and Derek Hoiem. *Rendering synthetic objects into legacy photographs*. ACM Transactions on Graphics (proc. of SIGGRAPH Asia), vol. 30, no. 6, 2011. (Cited on pages 16, 17 and 20.)

[Kazhdan 2006] Michael Kazhdan, Matthew Bolitho and Hugues Hoppe. *Poisson surface reconstruction*. In Eurographics Symposium on Geometry Processing, pages 61–70, 2006. (Cited on pages 22 and 28.)

[Kimmel 2003] Ron Kimmel, Michael Elad, Doron Shaked, Renato Keshet and Irwin Sobel. *A Variational Framework for Retinex*. Int. J. Comput. Vision, vol. 52, no. 1, pages 7–23, 2003. (Cited on page 13.)

[Koenderink 2003] J. J. Koenderink and S. C. Pont. *Irradiation direction from texture*. Journal of the Optical Society of America, vol. 20, no. 10, pages 1875–1882, 2003. (Cited on page 52.)

[Lalonde 2009] Jean-François Lalonde, Alexei A. Efros and Srinivasa G. Narasimhan. *Webcam Clip Art: Appearance and Illuminant Transfer from Time-lapse Sequences*. ACM Transactions on Graphics (proc. of SIGGRAPH Asia), vol. 28, no. 5, 2009. (Cited on page 52.)

[Lalonde 2011] Jean-François Lalonde, Alexei A. Efros and Srinivasa G. Narasimhan. *Estimating the natural illumination conditions from a single outdoor image*. International Journal of Computer Vision, 2011. (Cited on page 52.)

[Land 1971] Edwin H. Land and John J. McCann. *Lightness and Retinex theory*. Journal of the optical society of America, vol. 61, no. 1, 1971. (Cited on page 13.)

[Lensch 2003] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich and Hans-Peter Seidel. *Image-based reconstruction of spatial appearance and geometric detail*. ACM Trans. Graph., vol. 22, no. 2, pages 234–257, 2003. (Cited on page 10.)

[Levin 2004] Anat Levin, Dani Lischinski and Yair Weiss. *Colorization using optimization*. ACM Transactions on Graphics (proc. of SIGGRAPH), vol. 23, no. 3, pages 689 – 694, 2004. (Cited on page 23.)

[Levin 2008] Anat Levin, Dani Lischinski and Yair Weiss. *A Closed-Form Solution to Natural Image Matting*. IEEE Trans. PAMI, 2008. (Cited on pages 15, 42, 71, 74, 81, 100 and 101.)

[Liu 2008] Xiaopei Liu, Liang Wan, Yingge Qu, Tien-Tsin Wong, Stephen Lin, Chi-Sing Leung and Pheng-Ann Heng. *Intrinsic Colorization*. ACM Transactions on Graphics (proc. of SIGGRAPH Asia), vol. 27, pages 152:1–152:9, 2008. (Cited on pages 18, 20 and 86.)

[Lopez-Moreno 2010] Jorge Lopez-Moreno, Sunil Hadap, Erik Reinhard and Diego Gutierrez. *Compositing images through light source detection*. Computers and Graphics, vol. 34, no. 6, pages 698–707, 2010. (Cited on pages 52 and 55.)

[Loscos 1999] Céline Loscos, Marie-Claude Frasson, George Drettakis, Bruce Walter, Xavier Granier and Pierre Poulin. *Interactive Virtual Relighting and Remodeling of Real Scenes*. In Rendering Techniques (Proceedings of the Eurographics Workshop on Rendering), volume 10, pages 235–246, 1999. (Cited on page 10.)

[Lowe 2004] David G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. Int. J. Comput. Vision, vol. 60, no. 2, pages 91–110, 2004. (Cited on page 21.)

[Luo 2012] Wei Luo, Zheng Lu, Ying-Qing Xu, Xiaogang Wang, Moshe Ben-Ezra and Michael S. Brown. *Synthesizing Oil Painting Surface Geometry from a Single Photograph*. In CVPR, 2012. (Cited on pages 20 and 108.)

[Marschner 1998] Stephen Robert Marschner. *Inverse rendering for computer graphics*. PhD thesis, Cornell University, 1998. (Cited on page 10.)

[Matsushita 2004a] Yasuyuki Matsushita, Stephen Lin, Sing Kang and Heung-Yeung Shum. *Estimating Intrinsic Images from Image Sequences with Biased Illumination*. In ECCV, pages 274–286, 2004. (Cited on pages 17, 71 and 74.)

[Matsushita 2004b] Yasuyuki Matsushita, Ko Nishino, Katsushi Ikeuchi and Masao Sakauchi. *Illumination Normalization with Time-Dependent Intrinsic Images for Video Surveillance*. IEEE Trans. PAMI, vol. 26, pages 1336–1347, 2004. (Cited on page 17.)

[Matusik 2004] Wojciech Matusik, Matthew Loper and Hanspeter Pfister. *Progressively-refined reflectance functions from natural illumination*. In Eurographics Symposium on Rendering, pages 299–308, 2004. (Cited on page 17.)

[Melendez 2011] F. Melendez, M. Glencross, G. J. Ward and R. J. Hubbold. *Relightable Buildings from Images*. In Eurographics: Special Area on Cultural Heritage, pages 33–40, 2011. (Cited on page 20.)

[Mohan 2007] Ankit Mohan, Jack Tumblin and Prasun Choudhury. *Editing Soft Shadows in a Digital Photograph*. IEEE Computer Graphics and Applications, vol. 27, no. 2, pages 23–31, 2007. (Cited on page 15.)

[Nillius 2001] Peter Nillius and Jan-Olof Eklundh. *Automatic Estimation of the Projected Light Source Direction*. In CVPR, pages 1076–1083, 2001. (Cited on page 52.)

[Okabe 2006] Makoto Okabe, Gang Zeng, Yasuyuki Matsushita, Takeo Igarashi, Long Quan and Heung-Yeung Shum. *Single-view relighting with normal map painting*. In Pacific Graphics, pages 27–34, 2006. (Cited on pages 16 and 43.)

[Omer 2004] Ido Omer and Michael Werman. *Color Lines: Image Specific Color Representation*. In CVPR, pages 946–953, 2004. (Cited on page 34.)

[Panagopoulos 2009] Alexandros Panagopoulos, Dimitris Samaras and Nikos Paragios. *Robust shadow and illumination estimation using a mixture model*. In CVPR, pages 651–658, 2009. (Cited on page 53.)

[Patterson 2012] Genevieve Patterson and James Hays. *SUN Attribute Database: Discovering, Annotating, and Recognizing Scene Attributes*. In CVPR, 2012. (Cited on page 109.)

[Pentland 1982] A.P. Pentland. *Finding the illuminant direction*. Journal of the Optical Society of America, vol. 72, no. 4, pages 448–455, 1982. (Cited on page 52.)

[Perez 1993] R. Perez, R. Seals and J. Michalsky. *All-weather model for sky luminance distribution – Preliminary configuration and validation*. Solar Energy, vol. 50, no. 3, pages 235–245, 1993. (Cited on page 61.)

[Perona 1990] Pietro Perona and Jitendra Malik. *Scale-space and edge detection using anisotropic diffusion*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, pages 629–639, 1990. (Cited on page 63.)

[Pharr 2010] Matt Pharr and Greg Humphreys. Physically based rendering: From theory to implementation, second edition. Morgan Kaufmann Publishers Inc., 2010. (Cited on pages 33 and 85.)

[Preetham 1999] A. J. Preetham, Peter Shirley and Brian Smits. *A practical analytic model for daylight*. In SIGGRAPH, pages 91–100, 1999. (Cited on pages 61 and 85.)

[Roberts 2009] David A Roberts. *PixelStruct, an opensource tool for visualizing 3D scenes reconstructed from photographs.*, 2009. (Cited on page 104.)

[Sanin 2012] Andres Sanin, Conrad Sanderson and Brian C. Lovell. *Shadow detection: A survey and comparative evaluation of recent methods*. Pattern Recognition, vol. 45, no. 4, pages 1684–1695, 2012. (Cited on page 15.)

[Sato 1997] Yoichi Sato, Mark D. Wheeler and Katsushi Ikeuchi. *Object shape and reflectance modeling from observation*. In SIGGRAPH, pages 379–387, 1997. (Cited on page 10.)

[Seitz 2006] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein and Richard Szeliski. *A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms*. In CVPR, pages 519–528, 2006. (Cited on page 21.)

[Shen 2008] Li Shen, Ping Tan and Stephen Lin. *Intrinsic Image Decomposition with Non-Local Texture Cues*. In CVPR, 2008. (Cited on pages 14, 45, 46, 92 and 95.)

[Shen 2011a] Jianbing Shen, Xiaoshan Yang, Yunde Jia and Xuelong Li. *Intrinsic Images Using Optimization*. In CVPR, 2011. (Cited on pages 15, 45, 46, 87, 89 and 95.)

[Shen 2011b] Li Shen and Chuohao Yeo. *Intrinsic images decomposition using a local and global sparse representation of reflectance*. In CVPR, pages 697–704, 2011. (Cited on pages 14, 87, 89 and 95.)

[Shor 2008] Yael Shor and Dani Lischinski. *The Shadow Meets the Mask: Pyramid-Based Shadow Removal*. Computer Graphics Forum, vol. 27, no. 2, pages 577–586, 2008. (Cited on page 15.)

[Sinha 1993] Pawan Sinha and Edward Adelson. *Recovering reflectance and illumination in a world of painted polyhedra*. In ICCV, pages 156–163, 1993. (Cited on page 13.)

[Sinha 2009] Sudipta Sinha, Drew Steedly and Richard Szeliski. *Piecewise planar stereo for image-based rendering*. In ICCV, pages 1881–1888, 2009. (Cited on page 28.)

[Snavely 2006] Noah Snavely, Steven M. Seitz and Richard Szeliski. *Photo tourism: exploring photo collections in 3D*. ACM Transactions on Graphics (proc. of SIGGRAPH), vol. 25, no. 3, pages 835–846, 2006. (Cited on pages 21, 28, 71, 104, 108 and 109.)

[Snavely 2008] Noah Snavely, Rahul Garg, Steven M. Seitz and Richard Szeliski. *Finding Paths through the World's Photos*. ACM Transactions on Graphics (proc. of SIGGRAPH), vol. 27, no. 3, pages 11–21, 2008. (Cited on pages 104 and 109.)

[Snavely 2010] N. Snavely, I. Simon, M. Goesele, R. Szeliski and S. M. Seitz. *Scene Reconstruction and Visualization From Community Photo Collections*. Proceedings of the IEEE, vol. 98, no. 8, pages 1370–1390, 2010. (Cited on page 21.)

[Sunkavalli 2007] Kalyan Sunkavalli, Wojciech Matusik, Hanspeter Pfister and Szymon Rusinkiewicz. *Factored time-lapse video*. ACM Transactions on Graphics (proc. of SIGGRAPH), vol. 26, no. 3, 2007. (Cited on page 17.)

[Sunkavalli 2008] Kalyan Sunkavalli, Fabiano Romeiro, Wojciech Matusik, Todd Zickler and Hanspeter Pfister. *What do color changes reveal about an outdoor scene?* In CVPR, 2008. (Cited on page 17.)

[Tappen 2005] Marshall F. Tappen, William T. Freeman and Edward H. Adelson. *Recovering Intrinsic Images from a Single Image*. IEEE Trans. PAMI, vol. 27, no. 9, 2005. (Cited on pages 13, 19, 92 and 95.)

[Tappen 2006] Marshall F. Tappen, Edward H. Adelson and William T. Freeman. *Estimating Intrinsic Component Images using Non-Linear Regression*. In CVPR, pages 1992–1999, 2006. (Cited on pages 13 and 19.)

[Troccoli 2008] Alejandro Troccoli and Peter Allen. *Building Illumination Coherent 3D Models of Large-Scale Outdoor Scenes*. Int. J. Comput. Vision, vol. 78, no. 2-3, pages 261–280, 2008. (Cited on page 10.)

[Tuite 2011] Kathleen Tuite, Noah Snavely, Dun-yu Hsiao, Nadine Tabing and Zoran Popovic. *PhotoCity: training experts at large-scale image acquisition through a competitive game*. In SIGCHI, pages 1383–1392, 2011. (Cited on page 71.)

[Varma 2004] M. Varma and A. Zisserman. *Estimating Illumination Direction from Textured Images*. In CVPR, volume 1, pages 179–186, June 2004. (Cited on page 52.)

[Weiss 2001] Yair Weiss. *Deriving intrinsic images from image sequences*. In ICCV, pages 68–75, 2001. (Cited on pages 17, 18, 19, 71, 74, 86, 87, 89 and 92.)

[Wu 2007a] Changchang Wu. *SiftGPU: A GPU Implementation of Scale Invariant Feature Transform (SIFT)*. `http://cs.unc.edu/~ccwu/siftgpu`, 2007. (Cited on page 21.)

[Wu 2007b] Tai-Pang Wu, Chi-Keung Tang, Michael S. Brown and Heung-Yeung Shum. *Natural Shadow Matting*. ACM Transactions on Graphics, vol. 26, no. 2, page 8, 2007. (Cited on pages 15 and 43.)

[Wu 2011] Changchang Wu, Sameer Agarwal, Brian Curless and Steven M. Seitz. *Multicore bundle adjustment*. In CVPR, pages 3057–3064, 2011. (Cited on pages 21 and 83.)

[Yan 2010] Xing Yan, Jianbing Shen, Ying He and Xiaoyang Mao. *Re-texturing by Intrinsic Video*. In Digital Image Computing: Techniques and Applications, pages 486–491, 2010. (Cited on page 20.)

[Yu 1998] Yizhou Yu and Jitendra Malik. *Recovering photometric properties of architectural scenes from photographs*. In SIGGRAPH, pages 207–217, 1998. (Cited on pages 10, 27, 33 and 52.)

[Yu 1999] Yizhou Yu, Paul Debevec, Jitendra Malik and Tim Hawkins. *Inverse global illumination: recovering reflectance models of real scenes from photographs*. In SIGGRAPH, pages 215–224, 1999. (Cited on page 10.)

[Yu 2006] Tianli Yu, Hongcheng Wang and Narendra Ahuja. *Sparse lumigraph relighting by illumination and reflectance estimation from multi-view images*. In Proc. Eurographics Symposium on Rendering (EGSR), 2006. (Cited on page 10.)

[Zhao 2012] Qi Zhao, Ping Tan, Qiang Dai, Li Shen, Enhua Wu and Stephen Lin. *A Closed-Form Solution to Retinex with Nonlocal Texture Constraints*. IEEE Trans. PAMI, vol. 34, pages 1437–1444, 2012. (Cited on pages 14, 87, 89, 92, 95, 96 and 97.)

Décomposition en images intrinsèques à partir de plusieurs photographies

Résumé :

La modification d'éclairage et de matériaux dans une image est un objectif de longue date en traitement d'image, vision par ordinateur et infographie.

Cette thèse a pour objectif de calculer une décomposition en images intrinsèques, qui sépare une photographie en composantes indépendantes : la réflectance, qui correspond à la couleur des matériaux, et l'illumination, qui représente la contribution de l'éclairage à chaque pixel. Nous cherchons à résoudre ce problème difficile à l'aide de plusieurs photographies de la scène. L'intuition clé des approches que nous proposons est de contraindre la décomposition en combinant des algorithmes guidés par l'image, et une reconstruction 3D éparse et incomplète générée par les algorithmes de stéréo multi-vue.

Une première approche permet de décomposer des images de scènes extérieures, à partir de plusieurs photographies avec un éclairage fixe. Cette méthode permet non seulement de séparer la réflectance de l'illumination, mais décompose également cette dernière en composantes dues au soleil, au ciel, et à l'éclairage indirect. Une méthodologie permettant de simplifier le processus de capture et de calibration, est ensuite proposée. La troisième partie de cette thèse est consacrée aux collections d'images: nous exploitons les variations d'éclairage afin de traiter des scènes complexes sans intervention de l'utilisateur.

Les méthodes décrites dans cette thèse rendent possible plusieurs manipulations d'images, telles que l'édition de matériaux tout en préservant un éclairage cohérent, l'insertion d'objets virtuels, ou le transfert d'éclairage entre photographies d'une même scène.

Mots-clés : édition d'image, rendu basé image, reconstruction multi-vues

Intrinsic image decomposition from multiple photographs

Abstract:

Editing materials and lighting is a common image manipulation task that requires significant expertise to achieve plausible results. Each pixel aggregates the effect of both material and lighting, therefore standard color manipulations are likely to affect both components.

Intrinsic image decomposition separates a photograph into independent layers: reflectance, which represents the color of the materials, and illumination, which encodes the effect of lighting at each pixel.

In this thesis, we tackle this ill-posed problem by leveraging additional information provided by multiple photographs of the scene. We combine image-guided algorithms with sparse 3D information reconstructed from multi-view stereo, in order to constrain the decomposition.

We first present an approach to decompose images of outdoor scenes, using photographs captured at a single time of day. This method not only separates reflectance from illumination, but also decomposes the illumination into sun, sky, and indirect layers. We then develop a methodology to extract lighting information about a scene solely from a few images, thus simplifying the capture and calibration steps of our intrinsic decomposition. In a third part, we focus on image collections gathered from photo-sharing websites or captured with a moving light source. We exploit the variations of lighting to process complex scenes without user assistance, nor precise and complete geometry.

The methods described in this thesis enable advanced image manipulations such as lighting-aware editing, insertion of virtual objects, and image-based illumination transfer between photographs of a collection.

Keywords: image editing, image-based rendering, image-guided propagation, multi-view stereo, light estimation