VISION UNDER CHANGING SCENE APPEARANCE: DESCRIBING THE WORLD THROUGH LIGHT AND SYMMETRIES

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Daniel Cabrini Hauagge

August 2014

VISION UNDER CHANGING SCENE APPEARANCE: DESCRIBING THE WORLD THROUGH LIGHT AND SYMMETRIES

Daniel Cabrini Hauagge, Ph.D.
Cornell University 2014

Change is an inexorable aspect of the world that surrounds us. Night gives way to day as the Earth rotates around its axis, weather changes, buildings decay. All these changes alter the appearance of our surroundings. In trying to understand our world it is sometimes useful to factor out changes that are not important to the subject of our study, for instance when attempting to determine if two pictures taken decades apart depict the same building it is useful to ignore the cracks and peeling paint that are due to aging, while other times a careful examination of change might reveal interesting phenomena, like how the shading produced by sunlight on objects surrounding us can tell the time of day.

In the first part of this thesis we examine changes in light that reveal information about materials and geometry. We introduce a simple pixel-wise statistic κ that we show is linked to ambient occlusion, a measure of light accessibility. This simple realization allows us to recover the albedo for the scene, which then allows us to obtain the lighting of each input image. We start our study by focusing on a simple setup, a static scene and camera where each image is captured under varying but unknown lighting.

We then extend this foundation in two ways, both of which apply to Internet photo collections of outdoor landmarks. This presents a much more challenging source of data as cameras are radiometrically uncalibrated and not registered,

natural lighting is much more complex than what our initial model expects, and occluders obscure parts of the scene.

First, we show how physically based models of outdoor illumination developed in the computer graphics community can be used to incorporate many of the subtleties of outdoor illumination into our algorithm, such as the influence of geolocation on the sun's path in the sky, and the changes in color and intensity that occur over the course of a day. This advance allows us to correctly estimate illumination for outdoor scenes, which we show is useful in estimating the correct timestamp for images.

Second, we show how the estimated lighting can be digested into a novel image descriptor, one that captures the distribution of light in a scene in a format that is independent of geometry. This descriptor allows one to reason about many phenomena that are linked to lighting, such as weather conditions, and time of day. It also enables queries to an image database based on how light is distributed in the scene, irrespective of geometry.

In the second part of this dissertation we look at change from another angle by tackling the problem of image matching. We ask ourselves how can we match challenging image pairs of architectural structures when changes in the images are too drastic for traditional methods to work. We devise novel feature detector and descriptors based on local symmetries, a mid-level cue that we show can be more robust to drastic changes than the more traditional edge based methods.

BIOGRAPHICAL SKETCH

Daniel Cabrini Hauagge grew up in Brazil and the United States (including a few years in Ithaca). He obtained a B.Sc. in Computer Engineering in 2006 and a M.Sc. in Computer Science in 2008 from the University of Campinas (UNICAMP) in Brazil. He returned to Ithaca for graduate studies in 2009 and expects to receive a Ph.D. in Computer Science from Cornell University in August of 2014.

I dedicate this thesis to my
parents Roberto and Claudete,
and to my sister Karina.

ACKNOWLEDGEMENTS

I want to thank my advisor Noah Snavely, his support and guidance allowed me to pursue a dream of completing a Ph.D. I also had the privilege of having Kavita Bala as my co-advisor. Her energy and enthusiasm for research are an inspiration and made our weekly meeting a joy. Together they helped me grow as a researcher and professional and I will be forever grateful for their efforts.

I thank Shimon Edelman for agreeing to be part of my special committee and introducing me to the study of Cognitive Sciences.

The first part of this thesis was done in close collaboration with my fellow lab-mates Scott Wehrwein and Paul Upchurch. It was an honor and pleasure to work with them. I will always remember our long nights at the lab close to the deadlines.

Part of what makes Cornell a world class university is the result of the hard work of the administrative staff. I thank in particular Becky Stewart and Randy Hess for their diligence.

At the Graphics and Vision Group I met extraordinary researchers that helped me in my research and taught me so much. I thank Wenzel Jakob, Ivaylo Boyadzhiev, Sean Bell, Pramook Khungurn, Albert Liu, Kevin Matzen, Kyle Wilson, Changxi Zheng, and Seven An.

Cornell also gave me the opportunity to meet amazing people whose friendship shaped me and helped me through this journey. I'd like to thank Renato Paes Leme, Bruno Abrahão, Amir Sadovnik, Rachel Neugarten, Fred Gouker, Camila Flor, Paula Petrica, Carolina Carvalho, Adarsh Kowdle, Guilherme Pinto, Camilo Velez Vega, Bernardo M. Aguilera, Ilan Shomorony, Marcelo Finger, and Hussam Abu-Libdeh.

Above all, I thank my parents Roberto and Claudete. They put my well being and education above theirs and showed me what it meant to persevere. The love and wholehearted support from my parents and my sister Karina were a source of strength that made this journey possible.

TABLE OF CONTENTS

Biographical Sketch . . . iii
Dedication . . . iv
Acknowledgements . . . v
Table of Contents . . . vii
List of Tables . . . ix
List of Figures . . . x

1 Introduction 1
1.1 Road map . . . 2
1.2 Bibliographical Notes . . . 3

I Light 4

2 Related Work 5
2.1 Photometric Ambient Occlusion . . . 5
2.2 Intrinsic Image Decomposition . . . 7
2.3 Reasoning About Outdoor Illumination . . . 8
2.4 Light Descriptors . . . 9

3 Photometric Ambient Occlusion for Intrinsic Image Decomposition 11
3.1 Ambient Occlusion . . . 13
3.2 A Model for Ambient Occlusion in Image Stacks . . . 14
3.2.1 Inputs and Image Formation Model . . . 15
3.2.2 Image Formation Model . . . 17
3.3 Algorithm . . . 22
3.4 Results . . . 24
3.4.1 Image Decomposition . . . 24
3.4.2 Ambient Occlusion . . . 27
3.4.3 Albedo . . . 28
3.5 Analysis . . . 29
3.5.1 Impact of Albedo on Ambient Occlusion Estimates . . . 32
3.5.2 Convergence Rate . . . 33
3.5.3 Impact of Interreflections . . . 34
3.5.4 Color in κ . . . 35

4 Reasoning about Photo Collections using Models of Outdoor Illumination 37
4.1 Modeling Illumination for Outdoor Points . . . 39
4.2 Albedo and Sun Position in Photo Collections . . . 41
4.2.1 Estimating Albedo for Sunlit Outdoor Scenes . . . 42
4.2.2 Estimating Local Visibility Angle . . . 43
4.2.3 Estimating Time of Day . . . 47

4.3 Implementation details 48
4.4 Experiments and Applications 49
4.4.1 Albedo 50
4.4.2 Timestamps 51

5 Light Descriptors in Photo Collections 56
5.1 Light Descriptors 57
5.1.1 Concept 58
5.1.2 Representation 58
5.2 Computing Light Descriptors from Photo Collections 63
5.2.1 Estimating Camera Response 64
5.2.2 Computing Descriptors 67
5.3 Applications 70
5.3.1 Sun Position 71
5.3.2 Sun Visibility 72
5.3.3 Object Insertion 74
5.3.4 Search 76
5.4 Discussion and Limitations 79

6 Conclusions and Future Work 81
6.1 Future Work 82

II Symmetry 84

7 Image Matching using Local Symmetry 85
7.1 Related Work 87
7.2 Local Symmetry 89
7.2.1 Scoring Local Symmetries 90
7.2.2 Gradient Histogram-Based Score 93
7.3 Scale Space 95
7.4 Local Symmetry Features 97
7.4.1 Feature Detector 97
7.4.2 Feature Descriptor 101
7.5 Experimental Results 103
7.5.1 Evaluating Detections 103
7.5.2 Evaluating Descriptors 105
7.6 Conclusion and Future Work 109

Bibliography 110

LIST OF TABLES

3.1 Local mean squared error (LMSE) for individual images of our algorithm for the 1st (only direct light) and 2nd estimates (direct and ambient term), together with results from other work when available. In the last four columns of the last row we show our average on the same subset of images as reported by [6, 80, 82, 32, 10]. On all cases our algorithm outperforms these prior methods. Note that the numbers marked with † are the geometric mean as reported in [10], other averages are arithmetic means. 31

4.1 Average and median timestamp error (in minutes) for various methods on our datasets. Rand represents chance over daylight hours, while LEN is the method of Lalonde et al. [50]. Sun position error in degrees can be calculated approximately by dividing minutes by 4. 52

5.1 Our landmark datasets taken from Internet photo collections. The number of photos and 3D points extracted are given. The last column is the average uncompressed double-precision descriptor size. 70

5.2 Quantitative results for our sun direction (measured in degrees) on the ground truth CASTLE evaluation set. We compare to the expected error given by randomly chosen times of day. 72

5.3 Cross-validation accuracy for sun visibility classification using our method and the method proposed by Lalonde et al. Our method outperforms theirs on all but CASTLE, and achieves an average accuracy of 75.7%. We also tried the approach from Shen and Tan, which uses the ratio of second to first order spherical harmonic energies. Note that the validation set for CASTLE is not the same as the set used for sun direction and is not heavily biased towards sunny images. 75

7.1 Average repeatability score for the top $n = 100, 200$ detections according to scale and score. In cases where the detector produced less than n keypoints we report the repeatability score for the full feature set. 105

7.2 Mean average precision for different combinations of detector and descriptor on our dataset. Rows represent the descriptors and columns the detectors. First four rows show performance for individual descriptors while last three rows show the performance for concatenated descriptors. Results for detectors SIFT, SYM-I, and SYM-G are over a subset of the images. 108

LIST OF FIGURES

3.1 Our method takes as input a stack of images captured with varying, unknown illumination and computes a per-pixel statistic (κ) over this stack. This statistic is then combined with a simple physical model of the local geometry at each point and illumination to obtain an estimate of the local visibility. Local visibility is then used together with the average image to obtain an estimate for per-point albedo (reflectance), which itself can be used to compute illumination for the original input images. 12

3.2 Histogram of pixel intensities for two points of TENTACLE over an image stack (only blue color channel). Notice that even though the two points have very similar albedos their histograms are quite different due to local visibility. Point A is mostly occluded with respect to the light source, so its intensity values are in general lower. 16

3.3 (a) A point x on a Lambertian surface is observed by camera c and illuminated by a distant, moving light source with intensity L_d, and a constant ambient term of intensity L_a. (b) The local geometry is approximated as a cyllindrical crevice, whose oppening is parametrized by the angle α. If the light source angle θ_d with the surface normal $\vec{n}$ is larger than α, light is blocked and does not reach point x at the bottom of the crevice. 19

3.4 $\kappa(\alpha)$ for different ratios of ambient to direct light r. Note that as $r \to \infty$ ($L_d = 0$) we have a constant curve ($\kappa(\alpha) = 1$) so information about α cannot be recovered. 21

3.5 A depiction of the full algorithm for computing the local visibility angle α and the reflectance ρ. Arrows show how information flows in our pipeline. Starting with an image stack we compute $\mathcal{E}[I]$ and $\mathcal{E}[I^2]$, which are used to compute κ. We then proceed to obtain a first estimate of the local visibility angle and reflectance, which are then refined using a non-linear optimization. 22

3.6 Results of our algorithm (2nd estimate). Each column shows results from a different dataset. The rows show 1) sample images from the original dataset, 2) our estimated AO, 3) albedo, and 4) the illumination in the sample image. 25

3.7 3D printed test objects TENTACLE and LIGHTWELL, together with a quarter dollar coin (for scale). Black tape surrounding LIGHTWELL was added to reduce subsurface scattering that resulted from light shining on the side of the object. 26

3.8 Left: the statistic κ computed for TENTACLE. Right two images: Comparison of estimated AO with ground truth (computer-generated). The background clutter is masked. 28

3.9 Comparison of LMSE error on the MIT intrinsic image dataset [32] (shorter bars are better, indicating less error). Compared algorithms are: Grayscale Retinex (GR-RET), Color Retinex (COL-RET), Weiss (W), Weiss+Retinex (W+RET), ours with only direct term (κ-D) and our second estimate containing direct and ambient terms (κ-DA). 28

3.10 Comparison of our method with W+Ret from the MIT benchmark. Results are for our first estimate of the albedo (i.e., ambient illumination is assumed to be zero) as this gave us the best results on the benchmark. We show here grayscale images as the benchmark uses grayscale versions of the decomposed images in its evaluation metric. 30

3.11 (a) Error in the estimated local visibility angle α vs. the true local visibility angle for the LIGHTWELL object printed in different colors (shown in the left). (b) Average Root Mean Square Error (RMSE) for our estimate of ambient occlusion vs. number of images used in the estimate. Each curve represents a different crevice depth and a corresponding local visibility angle α. 33

3.12 Impact of interreflections in our estimates. Top: two renderings of LIGHTWELL, only the direct component of light and then with interreflections included (notice that shadows become brighter due to reflected light). Middle plot shows a cut through the rendered LIGHTWELL ambient occlusion estimate versus ground truth. Interreflections cause our method to over estimate ambient occlusion. Bottom bar plot shows the absolute error in the estimated ambient occlusion at the center of the crevices. 36

4.1 The Statue of Liberty under a variety of natural illumination conditions 38

4.2 Influence of geolocation, date and time, orientation, and local visibility on the illumination at a point. On the left, an object located near the equator sees a band of sun paths (shaded in yellow) that is centered directly overhead. In this location, the point p_1 at the bottom of a crevice can sometimes see the sun, whereas p_2 cannot. On the right, we see a location farther from the equator, with a different sun path, where the reverse is true. 40

4.3 Pipeline for albedo estimation. Given a geographic location (a), we tabulate $L(\phi, \lambda, t, \alpha, \vec{n})$ over all parameter values (b), where we show one sphere per combination of α and time of day, with each point in the sphere representing a different normal direction $\vec{n}$. We then compute $\mathcal{E}[L]$ and κ for each α and $\vec{n}$ (c), producing the curves for $\kappa(\alpha)$ in (d) and $\mathcal{E}[L](\alpha)$ in (e) for each normal. For an observed value of κ, we look up α and then predicted average illumination $\mathcal{E}[L]$ (f), allowing us to estimate albedo. Green regions in (c) correspond to combinations of $\vec{n}$ and α for which we cannot reliably recover albedo. 44

4.4 How α_{min} varies with geographic latitude and normal. Each point on the sphere represents a surface normal direction, and its color encodes the α angle of a crevice that sees the sun 10% of the time during daylight. Normals pointing near the sun path (as determined by latitude) have lower α_{min} values and are more informative because κ is meaningful over a greater range of α. The rightmost column shows κ curves and α_{min} for the three different surface normals. 46

4.5 Results of ρ-sunsky compared to ρ-unif, with and without the mask generated using ρ-sunsky (which discards normals for which $\alpha_{min} > \alpha_0$). We also list the Local Mean Squared Error for each result. 51

4.6 Examples of timestamping results. The plot on the right shows timestamp error metrics over the full day for the image on the left. Colored triangles show the global minimum (estimated timestamp) for each method, and the red triangle indicates the ground truth timestamp. 54

4.7 Alternative visualization of timestamping results. We select a set of images taken in July with a given timestamp, reproject them to a single viewpoint using a homography, and average them. We show that Exif timestamps (left) do not produce coherent lighting when averaged due to the timestamp errors. Using our timestamps (middle), the average images match the lighting in the corresponding reference images (right), taken from the AMOS dataset where timestamps are known. Note highlighted cast shadow at 2PM and the clear change from 9AM to 2PM. . . 55

5.1 *Example lighting descriptors.* This figure shows example photographs from a range of scenes and illumination conditions, along with a visualization of the lighting descriptor we compute. Our descriptors are designed to be invariant to geometry and materials, and to capture important information about the illumination, such as directionality, color, and local effects. Note, for instance, for images with strong directional lighting from the sun, our descriptors are bright in the corresponding directions, and how the color of the surface itself (e.g., the yellow building in the upper middle image) does not appear in the descriptor. Our descriptors are simple to derive from structure-from-motion reconstructions from large Internet photo collections, and applicable in a variety of tasks. 61

5.2 The effects of spherical harmonic compression on our descriptors. On the left, we show a sample image. On the right, we show the original descriptor as computed from the image (top), the full unwrapped spherical function reconstructed after fitting using nine spherical harmonic coefficients (middle), and the same reconstruction masked according to which normals were missing data in the input (bottom). 62

5.3 *Our pipeline.* Beginning with a large photo collection of a scene, we use structure from motion (SfM) to reconstruct a 3D point cloud with surface normals. We also assign a camera response curve to each photo in the collection, and use all of this data to compute a surface albedo for each scene point. These albedos can be factored out of a new photo to produce irradiance values, which, along with surface normals, are digested into our lighting descriptor. Here, the lighting descriptor captures the strong directional component of the illumination. 65

5.4 The camera response curve for a Canon Power Shot A1000 IS for the green channel. We compare here the curve recovered from a registered set of images with varying exposure (from the Middlebury dataset) with the simple $\gamma = 2.2$ curve and the curves at different levels of our hierachical camera response data structure. 68

5.5 Inserting an object into multiple images of a scene with consistent lighting. On the left, we show an object inserted in a consistent 3D location across four images with consistent lighting within each image. On the right one more example, notice that the shading on the synthesized yellow ball matches the real-world white spherical lamp on the left side of the image. 76

5.6 Example light similarity search results. Queries were matched against database images which had at least 320 occupied histogram bins in common. Each row contains one query image (leftmost image, marked in red) and the nearest search results following on the right. The first five rows show results for queries within a dataset (RIO, HUMAYUN, PEÑA, CASTLE-day and CASTLE-night) while the remaining rows show results for cross-dataset queries (HUMAYUN ↔ CASTLE and PEÑA ↔ HEAVEN). 77

7.1 *Difficult image pairs.* Each pair of images in this figure shows a registered view of a building. Despite the geometric consistency, these images are difficult for feature matching algorithms because of dramatic changes in appearance, due to different depictions (drawing vs. photo), different time periods (modern vs. historical), and different illumination (day vs. night). While these images are dissimilar at a pixel level, each image in a pair exhibit similar symmetries, which we seek to exploit for matching. These images are selected from our dataset. 86

7.2 By taking a slice of a 2D image f, we see that the problem of detecting symmetry can be posed of that of determining if a set of 1D slices through f are symmetric. Left: an image with (approximate) bilateral symmetry about the vertical axis. All horizontal slices (green) are close to even functions (as shown in the function profile at bottom). Right: (approximate) $2n$-fold rotational symmetry. Most slices through the center of the image are even functions. 91

7.3 Response of the Laplacian filter as a function of scale for the central pixel of three synthetic images (top row), using SYM-IR. Two weight masks w_σ are shown: ring (middle row) and Gaussian (bottom row). The horizontal axis corresponds to kernel size and the vertical axis to fraction of maximum response of the LoG filter. The ring mask gives a much more localized response. 97

7.4 Two distinct levels of the scale space pyramid for SYM-G on the image in Figure 7.5. The top row shows the symmetry score computed at a fine scale, the bottom row a coarse scale. Columns correspond to horizontal, vertical, and the product of the two symmetries. 98

7.5 Detected features for SIFT, SYM-I and SYM-G for an example image. Each circle shows a feature with scale. To ease vizualization the non-maxima overlap threshold was set to a stricter 0.1 for SYM-I. Note how the symmetry-based detectors more reliably fire on features such as the ring of circles around the clock's circumference. Our symmetry features also tend to fire in larger regions on average than SIFT. 99

7.6 Detected features for SYM-G for images of the Notre Dame Cathedral. Note how some symmetric features are repeatably detected across each image, including the central rose window. . 100

7.7 Visualization of the LPG descriptor. Left: input image with marked feature and descriptor support. Right: the three symmetry score maps. 102

7.8 Example results for the detector repeatability experiment for two pairs of images, for each of three detectors, SIFT, SYM-I, and SYM-G. The first plot shows the repeatability score as a function of subsets of features with largest scale, and the second plot shows the repeatability score according to subsets of features with highest detector score. In these plots, higher (closer to 1) represents better repeatability. 106

7.9 Precision-recall curves for selected image pairs from the dataset. Each column corresponds to a different detector and each row to a different pairs of images. 108

CHAPTER 1

INTRODUCTION

Change is an inexorable aspect of the world that surrounds us. Night gives way to day as the Earth rotates around its axis, weather changes, buildings decay. All these changes alter the appearance of our surroundings. In trying to understand our world it is sometimes useful to factor out changes that are not important to the subject of our study. For instance, when attempting to determine if two pictures taken decades apart depict the same building it is useful to ignore the cracks and peeling paint that are due to aging. While other times a careful examination of change might reveal interesting phenomena, like how the shading produced due to sunlight on objects surrounding us can tell time of day.

In this thesis we observe these changes through the large trove of photos gathered online by people all over the globe. In the past 15 years we have witnessed an explosion in the number of images available online thanks to affordable and ubiquitous digital cameras and online photo-sharing portals. What can we learn about a scene from these photos? The photos are uncalibrated, taken in different and unknown lighting conditions, with occluders that obscure features making detection difficult, etc. Yet despite all of this complexity and variation there is a lot of information to be learned from them. For instance, advances in structure from motion and image matching techniques have created systems capable of large scale 3D reconstructions of many large outdoor spaces, recovering accurate camera position and sparse geometry [84, 26].

1.1 Road map

This dissertation talks about two views on change. In the first part we focus on materials and lighting in a scene and what can be learned from them.

In Chapter 3 we introduce a simple method for estimating ambient occlusion, a measure of how much light can access a point on the surface of an object. Our method takes as input a large set of images with varying, but unknown, illumination. It then computes a simple per-pixel statistic that is coupled with a physical model of the illumination at a point, enabling us to estimate ambient occlusion at each point in the scene. Given ambient occlusion we proceed to estimate an albedo per scene point which allows us to estimate the illumination at each of the input images. Our method is simple, highly scalable, and obtains state-of-the art performance on the MIT Intrinsic Images benchmark.

We refine our illumination model in Chapter 4 to incorporate aspects of illumination that are particular to outdoor scenes. Our new method leverages a physically based model of outdoor illumination [36] in order to account for the sun path and the changes in intensity and color that occur over the course of a day. Whereas in Chapter 3 we use registered images of a static scene, we now apply the method to unstructured photo collections and show that our method can be successfully used to estimate the sun position in each image.

In Chapter 5 we introduce a novel image descriptor, one that captures the distribution of light in a scene. The method leverages large 3D models obtained from structure from motion and our algorithm for estimating ambient occlusion from Chapter 3, which together are used to obtain an estimate of the lighting in the scene for a sparse set of points. These estimates are then processed to produce a descriptor of the lighting in the scene in a format that is independent of the scene's geometry. This descriptor allows one to reason about many phenomena that are linked to lighting, such as weather conditions and time of day. It also enables queries to an image database based on how light is distributed in the scene, irrespective of geometry.

In the second part of this dissertation we look at change from another angle. In Chapter 7 we tackle the problem of image matching and ask ourselves how can we match challenging image pairs of architectural structures when changes in the images are too drastic for traditional methods to work. We devise novel feature detector and descriptors based on local symmetries, a mid-level cue that we show can be more robust to drastic changes than the more traditional edge based methods.

1.2 Bibliographical Notes

The work presented in Chapter 3 is an extended version of [34]. The method in Chapter 4 was presented as a poster at [35] and published in [35]. Chapter 7 is an extended version of [15].

Part I

Light

CHAPTER 2

RELATED WORK

The work presented in the first part of this thesis is related to many different problems studied in computer vision. Starting with estimation and use of Ambient Occlusion, which has received relatively little attention from the community, there is also a strong connection to the intrinsic image decomposition literature, since the methods in Chapters 3 and 4 estimate the light source visibility in order to determine the albedo in the scene. The work in Chapter 4 is related to other photometric methods that deal with outdoor illumination conditions. Finally, the work in Chapter 5 introduces a novel type of descriptor and therefore is related to general light representations and other families of descriptors.

2.1 Photometric Ambient Occlusion

Ambient occlusion has received relatively little attention in computer vision. Some examples of its use include early work in shape-from-shading [55], where it was used in models of images under diffuse illumination, as well as more recent work that considers AO in various applications. In the context of high-quality face capture, Beeler et al. [13] and Aldrian and Smith [5] model AO by assuming a uniform, constant, light source, and require an initial estimate of the geometry.

In the area of multi-view stereo, Wu et al. assume that a scene consists of a single albedo, and so the scene brightness under uniform area lighting is itself a good approximation to AO (e.g., darker regions are more occluded) [92].

For the problem of intrinsic image decomposition from large photo collections, Laffont et al. require accurate estimates of the albedo for a sparse set of 3D scene points [49]. To account for points that are darker due to AO, they compute AO explicitly by generating and analyzing a 3D scene reconstruction.

In contrast to these methods, the method we present in Chapter 3 does not explicitly model geometry, and instead reason about AO purely from observed pixel values. This yields a very simple approach that could be used as a preprocess to account for light visibility in other vision algorithms.

Our work is also related to methods that analyze pixel intensity variation in images under varying illumination. Weiss proposed a method for intrinsic images from image sequences [89], derived from a model of edge intensities. In that work, a final step involves integrating a gradient field to compute a reflectance image. In our experience, and in agreement with other reports [30], this integration performs poorly in the presence of soft and persistent shadows (exactly the kind caused by AO), and we find that it can also propagate noise across the image. In contrast, our method explicitly models one cause of soft shadows (namely AO), and does not require a final integration step, which we find makes the algorithm more robust. For outdoor scenes illuminated by the sun, Sunkavalli et al. recover albedo and normals by directly tracking the intensity of pixel values over time [86]. While they use heuristics to determine whether a pixel is in shadow, our method makes no such hard decisions, instead reasoning about statistics over the entire image sequence. In more recent work, Barron and Malik optimize for reflectance, shape, and illumination from single images under strong priors on illumination and color of natural scenes [7]. In contrast, our method operates at a per-pixel level and does not make assumptions about the texture in the scene.

Photometric stereo techniques [91, 11] are similar to our method in their setup and the fact that they estimate albedo, but differ in that they recover different information about shape (surface normals), compared to our work. Our approach is especially related to uncalibrated photometric stereo, in which the light source directions are unknown. A key challenge in photometric stereo is dealing with shadows, either by detecting them in some manner [17, 87] (a nontrivial problem with surfaces of varying albedo or complex self-occlusions), or treating them as a source of noise [94].

Sunkavalli et al. reason about lighting visibility of surface points, by clustering them into "visibility subspaces" that see a common set of lights [87]. However, they use an implicit model of lighting visibility that grows in complexity as the number of lighting conditions increases. In contrast, our method relies on a simple per-pixel measure of ambient occlusion that becomes more robust as more images are added. In addition, our model incorporates ambient illumination as well as directional lighting.

Finally, our work is also related to methods that recover shape from ambient occlusion [55, 70], and our algorithm could potentially be used to generate inputs to such methods.

2.2 Intrinsic Image Decomposition

Intrinsic image techniques have also been used to estimate albedo and illumination maps from single or multiple images [89, 82]. Laffont et al. also work with multiple images from varying viewpoints from one [49] or many [48] points in time. However, their approach either requires extra input (e.g., a light probe [49]), or additional smoothness priors [48]. The statistical approaches we present in Chapters 3 and 4 yields a per-pixel estimate that avoid smoothness priors and requires only measurements from the images themselves.

2.3 Reasoning About Outdoor Illumination

Estimating illumination from outdoor imagery. Several vision methods have been proposed to recover illumination from single images [64]. Lalonde et al. [50] use an analytic sun-sky model as a cue for determining sun direction, by using the model to predict the appearance of the visible sky in a single outdoor photograph. The method in Chapter 4 uses a sun-sky model to estimate and predict the appearance of *objects* using statistics across many images. Both methods can be used to timestamp images, and we compare to [50] in Section 4.4.

Haber et al. [33] pose the problem of estimating reflectance and illumination of a scene from Internet photos as an explicit inverse rendering problem, which results in a complex optimization procedure. Their work assumes arbitrary (smooth) illumination; in contrast, we leverage strong models of outdoor illumination to derive a much simpler statistical approach.

Outdoor photometric stereo. Several techniques estimate scene geometry and appearance from outdoor illumination over time, particularly from webcam data. Sunkavalli et al. decompose webcam video into components modeling albedo, geometry, and shading using a factoring approach [86, 87]. Ackermann et al. [3] and Abrams et al. [2] estimate scene albedos and normals using photometric stereo, using the sun as the varying light source. These methods rely on images captured from a single, static, georegistered camera, with known timestamps (and hence sun position). In contrast, we work with images taken from many viewpoints and cameras with largely incorrect timestamps. Finally, Yu et al. [96] solve photometric stereo problems for images using environment light measured using light probes. Again, their data is more structured in that they directly measure illumination.

2.4 Light Descriptors

Representations of illumination play a key role in both computer graphics and vision. The work in Chapter 5 is particularly inspired by prior work on extracting light descriptions from imagery. Notably, the Webcam Clip Art work of Lalonde et al. use calibrated webcam data to derive analytic sun positions from photo collections [52], which they use to generate full environment maps via graphics models of the sun and sky [71]. Earlier, Lalonde et al. compute coarse illumination descriptions by computing simple color histograms for sky, ground, and vertical surface regions of images [51], and use these to match rough illumination statistics between images (to enable "Photo Clip Art"). Our work goes further than Webcam Clip Art in that we address large consumer photo collections and not carefully calibrated webcam data. We also explore richer illumination descriptors than in Photo Clip Art, and explicitly separate albedo from illumination.

Many other representations of illumination are used in graphics. We explore descriptors related to **irradiance maps**, but also augment these descriptors with additional information that capture more than simple distant illumination. **Spherical harmonic** coefficients are also commonly used to represent illumination, and it has been shown that diffuse illumination lies in a low dimensional subspace in this representation [72, 12, 39, 42]. These representations also allow for recovery of illumination from imagery [73, 11] (generalizing from shape-from-shading and photometric stereo), but these methods often require limiting assumptions or more careful calibration. In graphics and vision, **intrinsic images** represent images as a product of reflectance and illumination layers [31], but the resulting light description is a pixel-level, rather than scene-level description of light that can be compared across images. Finally, one can directly estimate explicit light sources (e.g., for indoor scenes), as in Chen et al. [18] or Karsch et al. [40], but these methods require sufficient training data with geometry or labeled illuminants. Xing et al. integrate many of these ideas together, but rely on user input to derive geometry and lighting information [95].

We also present a simple way to extract lighting descriptors from image collections reconstructed using structure from motion. Our method relies on per-pixel statistics, and is more scalable and easier to implement than inverse rendering methods based on global optimization [33, 24, 78]. Similarly, we do not rely on complex priors, such as those of Barron and Malik [8], nor additional information (e.g., Kinect data [9] or user input [41]). While we opt for simplicity, we could also use these methods (and other intrinsic images methods such as [49]) to separate shading from reflectance in our imagery for generating descriptors. Our overarching goal is to show the practical utility of simple descriptors derived from photo collections.

CHAPTER 3

PHOTOMETRIC AMBIENT OCCLUSION FOR INTRINSIC IMAGE DECOMPOSITION

Many vision methods estimate the physical properties of a scene from images taken under varying illumination. Some notable examples include recovering surface normals using photometric stereo [11, 87, 3], recovering diffuse reflectance and illumination as intrinsic images [89, 53], and computing low-dimensional models of appearance of objects and scenes [88, 28]. However, these methods typically disregard the effect of the *local visibility* of illumination in determining shading. Further, many of these methods require calibrated setups (e.g., known lighting directions), special priors (e.g., smoothness of surface reflectance), or limiting assumptions (e.g., no cast shadows).

In this chapter we revisit such estimation problems by posing the following question: what can we tell about a scene point simply by observing its appearance under many different, unknown illumination conditions? The appearance of a point over such an image stack depends on many factors, such as the point's albedo and the distribution of illuminations. However, a key observation is that the local visibility of a point—i.e., its accessibility to light from different directions, often modeled as Ambient Occlusion (AO) in computer graphics—is also an important property in determining its appearance in images. We show that we can estimate ambient occlusion directly from image observations, by introducing a simple pixel-wise, aggregate statistic (κ in Figure 3.1), and relating this statistic to ambient occlusion. To do so, we consider a physical model of a scene point with a cone of visibility to the hemisphere, lit by a moving point light and constant ambient light over the image stack. We then combine this model with

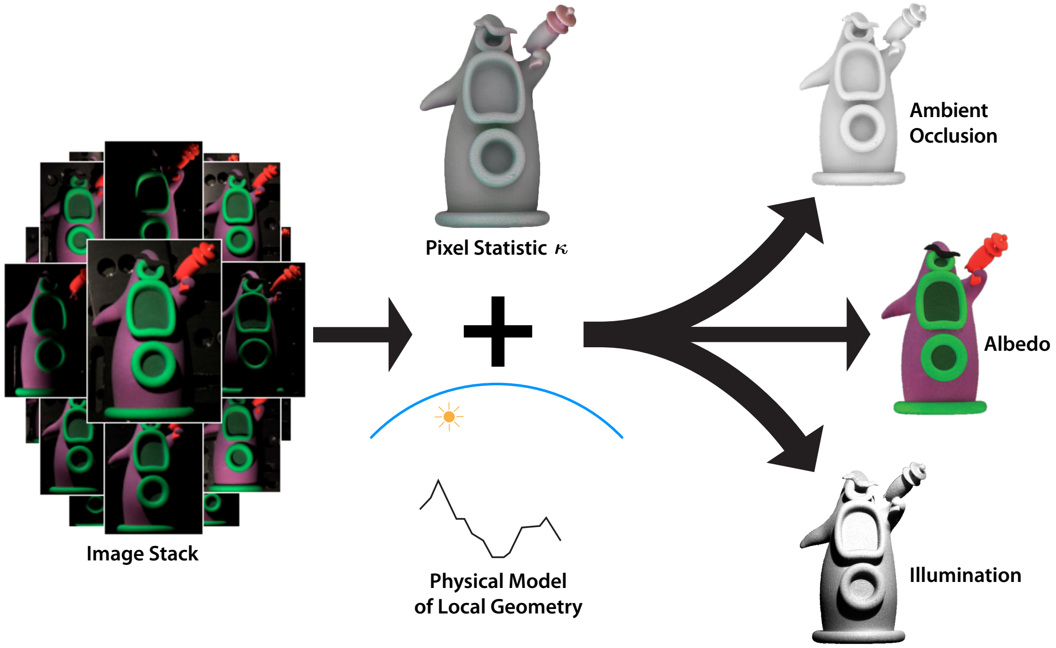


Figure 3.1: Our method takes as input a stack of images captured with varying, unknown illumination and computes a per-pixel statistic (κ) over this stack. This statistic is then combined with a simple physical model of the local geometry at each point and illumination to obtain an estimate of the local visibility. Local visibility is then used together with the average image to obtain an estimate for per-point albedo (reflectance), which itself can be used to compute illumination for the original input images.

our statistic to infer ambient occlusion for each scene point. This kind of lighting visibility is often treated as a nuisance in computer vision methods, and in many cases is simply ignored. In contrast, we explicitly model such visibility for each scene point, and use it to aid in estimating other physical parameters, such as surface albedo. The result is a *photometric* approach to estimating ambient occlusion and albedo.

Our method has several key properties: we do not require knowledge of light positions, explicit scene geometry, or surface normals. The setup for acquisition is simple, requiring a point light source and a camera. However, we do assume that light source positions vary uniformly over the full hemisphere, although in practice we achieve good results even when this assumption does not hold. Note that we use the term image "stack" to refer to a set of images of the same scene lit under varying illumination, but captured from the same viewpoint. No frame-by-frame coherence or ordering is implied.

We demonstrate our method in experiments on several scenes. These include artificially generated images from a physically based renderer, as well as real objects captured in a laboratory environment. Our experiments on real objects include a validation on 3D printed objects with known geometry, including the TENTACLE dataset in Figure 3.1. In addition, we show that our method—despite its simplicity and its per-pixel analysis of a scene, without additional smoothness priors—outperforms current approaches on the MIT intrinsic images benchmark [30]. This demonstrates the utility of reasoning about AO when measuring properties of scenes from images.

3.1 Ambient Occlusion

Ambient occlusion (AO) [54] is a measure of light accessibility commonly used in computer graphics to properly account for ambient illumination. Formally, for a single scene point x, AO is the integral over the hemisphere

$$AO(x) = \frac{1}{\pi} \int_{\Omega} V(x, \vec{\omega}) \langle \vec{n}, \vec{\omega} \rangle d\omega \tag{3.1}$$

of the local visibility function $V(x, \vec{\omega})$ (i.e. $V(x, \vec{\omega}) = 1$ if there are no occluders between point x and the environment in direction $\vec{\omega}$, $V(x, \vec{\omega}) = 0$ otherwise) weighted by the dot product $\langle \vec{n}, \vec{\omega} \rangle$ between direction $\vec{\omega}$ and the point normal $\vec{n}$. For an example, see upper right of Figure 3.1. At points where most of hemisphere is occluded, e.g., in a deep crevice, V is mostly 0 and so AO is close to

0, while for points whose visibility of the hemisphere is unoccluded, AO is 1. If the albedo at x is ρ, the measured radiance due to constant, ambient illumination with intensity L_a can be expressed as:

$$I_a = \rho \pi l_a AO \tag{3.2}$$

Note that this only considers the first bounce of light (direct illumination), and as such does not account for interreflections.

Two properties of ambient occlusion that are useful in computer vision are: (1) it is independent of surface albedo, and so variation and discontinuities are due only to scene geometry, and (2) it explains in a simple way why regions with same albedo can have different intensities even when lit with uniform illumination [55].

In computer graphics, the main focus is on computing ambient occlusion in 3D scenes to render images [68, 44, 67]. In contrast, we are interested in *estimating* ambient occlusion from a set of images illuminated by a varying, unknown light source.

3.2 A Model for Ambient Occlusion in Image Stacks

We now describe how to obtain a simple approximation to ambient occlusion (AO) by observing pixel intensities in multiple images under varying directional lighting. We first introduce a physically-based image formation model for our measure of AO, then use this model to derive AO and albedo from image sequences.

3.2.1 Inputs and Image Formation Model

Our method takes as input a set of images, $I_1, I_2, \ldots, I_n$, captured from a fixed camera observing a static, Lambertian scene. The scene is lit by an unknown, directional light source that changes from image to image, together with a uniform ambient light source; both have constant intensity over time. We further assume that the distribution of directional light sources across the image stack is uniform over the hemisphere. The images are radiometrically calibrated and so the image intensity $I(x)$ at each pixel x is proportional to the radiance at a given scene point under a particular illumination. Because the camera is static, the same pixel x records radiance for the same scene point in each image. In the following derivation the images are treated as monochromatic without loss of generality.

A key idea in our work is that for a given pixel x, the measured radiances over all images are drawn from an underlying distribution that we refer to as its *pixel intensity distribution* (PID). This distribution of pixel intensities at a point is related to the distribution of illuminations over the image stack, as well as to the albedo of that point and to the surrounding geometry (which can occlude the light source from the point of view of that point). Figure 3.2 shows an example of observed PIDs in an image stack for two points. For example, a point in a deep concavity will very often appear dark, because light rarely reaches it (only when the light is shining straight down into the hole). Such a point will have a PID with mostly low intensity values. (For example, consider point A in Figure 3.2.) The intuition then, is that the samples we record give us information about a pixel's PID, which in turn reveals information about surface albedo and ambient occlusion. As we capture images lit under more and more possible

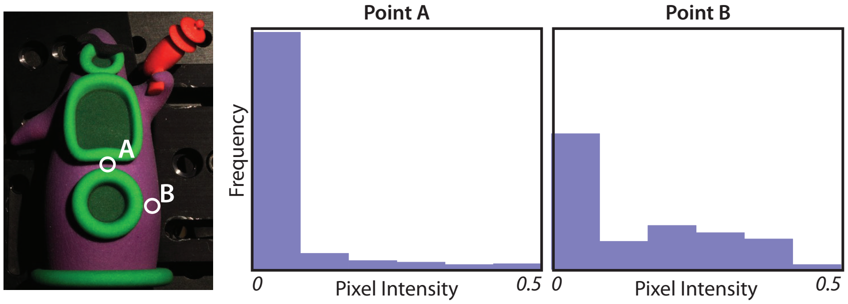


Figure 3.2: Histogram of pixel intensities for two points of TENTACLE over an image stack (only blue color channel). Notice that even though the two points have very similar albedos their histograms are quite different due to local visibility. Point A is mostly occluded with respect to the light source, so its intensity values are in general lower.

directions, we begin to capture the actual underlying PID of a pixel.

As a useful summary of a PID, we introduce a statistic for a single pixel x over time, which we denote κ:

$$\kappa(x) = \frac{\mathcal{E}[I(x)]^2}{\mathcal{E}[I(x)^2]} \tag{3.3}$$

where $\mathcal{E}[\cdot]$ is the expectation operator over the set of images. That is, κ is the square of the expected (average) intensity value for that pixel, divided by the expected squared pixel intensity; this quantity is related to the *coefficient of variation*, a normalized measure of variance used in statistics.[1] Figure 3.1 (top center) shows κ for an example image stack. In what follows, we show that this simple ratio of statistics over recorded intensities yields an approximation to ambient occlusion; to understand this relationship between κ and ambient occlusion, we first describe our image formation model, then relate this to a physical model of local scene geometry.

For a Lambertian scene, an image formation model commonly used in the

[1] The coefficient of variation, c_v, is defined as $\frac{\sigma}{\mu}$, so the statistic $\kappa = \frac{1}{1+c_v^2}$.

intrinsic images literature is:

$$I(x) = \rho(x)L(x) \tag{3.4}$$

where $I(x) \in \mathbb{R}^{+}$ is the observed radiance at point x in the image, $\rho(x) \in [0, 1)$ is the diffuse albedo, and $L(x) \in \mathbb{R}^{+}$ is a factor that depends on both light and geometry.

Over our sequence of images I, $\rho(x)$ is constant and greater than zero, while $L(x)$ varies due to changes in illumination. Under these assumptions, we can substitute Eq. (3.4) into the definition of our κ statistic in Eq. (3.3) to obtain:

$$\kappa = \frac{\mathcal{E}[\rho L]^2}{\mathcal{E}[\rho^2 L^2]} = \frac{\cancel{\rho^2}\mathcal{E}[L]^2}{\cancel{\rho^2}\mathcal{E}[L^2]} \tag{3.5}$$

(for simplicity, we do not explicitly write the dependence on x, but as before κ is a statistic defined per pixel across the image stack). Thus, κ depends only on the lighting factors L, and not on albedo.

What range of values can κ take on? Because κ is the quotient of non-negative numbers, it follows that $\kappa \geq 0$. By observing that $\mathrm{Var}(I) = \mathcal{E}[L^2] - \mathcal{E}[L]^2 \geq 0$ we can also show that $\kappa \leq 1$. For points that *never* receive light $\mathcal{E}[L] = 0$ and $\kappa = 0$ (one can arrive at this via a limit analysis). For points whose illumination term never changes we have that $\mathrm{Var}[I] = \mathcal{E}[L^2] - \mathcal{E}[L]^2 = 0$, which implies $\mathcal{E}[L^2] = \mathcal{E}[L]^2$ and therefore $\kappa = 1$. This behavior suggests that κ might be useful as a measure of ambient occlusion at a point.

3.2.2 Image Formation Model

So far we have shown that κ is independent of albedo and is bounded. But what exactly does κ tell us about a scene point? As a statistic, κ relates to the geometry and visibility at a point; to show this, we introduce a simplified geometry and lighting model to connect κ to a physical measure of local visibility.

Our model assumes that the visibility at a point can be approximated by a cone of angle α. This idea, along with our illumination model, is illustrated in Figure 3.3, where a point x on a Lambertian surface, is observed by camera c while illuminated by two light sources: a directional light with intensity L_d, and a background ambient illumination with constant intensity L_a. One can think of these two components as roughly similar to a "sun" (directional) and a "sky," (ambient) light source, respectively. Surface geometry around the point blocks all light outside the cone with angle α from reaching x. We refer to this angle $\alpha(x)$ as the *local visibility angle* for point x. Further, across our input images, we assume that the directional light uniformly samples the full hemisphere, so each measure of the radiance of x captured by the camera corresponds to a different (unknown) position for the light L_d. Given these assumptions, and sufficient samples of images under different illumination conditions, $\kappa(x)$ only depends on the local visibility angle $\alpha(x)$, and we can derive the relationship between κ and α as follows:

To begin, each image I is the sum of the contributions from both light sources:

$$I = I_d + I_a \tag{3.6}$$

The directional component I_d varies from image to image and depends on the angle $\theta_d(t)$ between the light source direction $\vec{\omega}_d(t)$ and the surface normal $\vec{n}$ at a point, and whether the light is blocked by other geometry. This component is

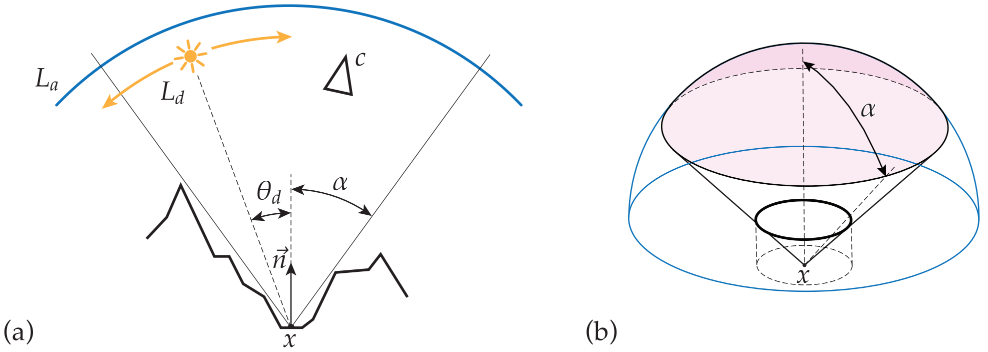


Figure 3.3: (a) A point x on a Lambertian surface is observed by camera c and illuminated by a distant, moving light source with intensity L_d, and a constant ambient term of intensity L_a. (b) The local geometry is approximated as a cyllindrical crevice, whose oppening is parametrized by the angle α. If the light source angle θ_d with the surface normal $\vec{n}$ is larger than α, light is blocked and does not reach point x at the bottom of the crevice.

given by:

$$\begin{aligned} I_d(t) &= \rho L_d V_\alpha(\vec{n}, \vec{\omega}_d(t)) \langle \vec{n}, \vec{\omega}_d(t) \rangle && (3.7) \\ &= \rho L_d V_\alpha(\theta(t)) \cos \theta_d(t) && (3.8) \end{aligned}$$

where V_α is the light visibility term: $V_\alpha(\theta) = 1$ if $\theta \leq \alpha$ (i.e., the light enters the visibility cone), ands $V_\alpha(\theta) = 0$ otherwise. The ambient component I_a is constant across the image stack for a given point, and is proportional to the projected solid angle of the local visibility angle α. In particular, from Eqs. (3.1) and (3.2) we can integrate the ambient illumination over the visible portion of the hemisphere to derive a closed form relationship between I_a and α at a given point:

$$I_a = \rho \int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\alpha} L_a \cos(\theta) \sin(\theta) d\theta d\varphi = \rho L_a \pi \sin^2 \alpha \qquad (3.9)$$

Given this model for I_d and I_a, to relate κ to our physical parameter α, we compute the expectations in Eq. (3.5) over images under varying light source positions:

$$\mathcal{E}[I] = \mathcal{E}[I_d] + \mathcal{E}[I_a] = \mathcal{E}[I_d] + I_a$$

$$\mathcal{E}[I^2] = \mathcal{E}[(I_d + I_a)^2] = \mathcal{E}[I_d^2] + 2I_a\mathcal{E}[I_d] + I_a^2$$

where we use the linearity of expectation, $\mathcal{E}[\cdot]$, and the assumption that I_a does not change over the image stack.

Finally, we can compute $\mathcal{E}[I_d]$ and $\mathcal{E}[I_d^2]$ in closed form by integrating over the visible cone of angles at the point, assuming the point light is uniformly distributed over the hemisphere for the image stack:

$$\mathcal{E}[I_d] = \frac{1}{2\pi}\int_{\varphi=0}^{2\pi}\int_{\theta=0}^{\alpha} I_d \sin\theta d\theta d\varphi = \frac{1}{2}\rho L_d \sin^2(\alpha) \tag{3.10}$$

$$\mathcal{E}[I_d^2] = \frac{1}{2\pi}\int_{\varphi=0}^{2\pi}\int_{\theta=0}^{\alpha} I_d^2 \sin\theta d\theta d\varphi = -\frac{1}{3}\rho^2 L_d^2\left(\cos^3(\alpha) - 1\right) \tag{3.11}$$

Given these equations, we can derive κ in terms of α as:

$$\begin{aligned}\kappa(\alpha) &= \frac{\mathcal{E}^2[I]}{\mathcal{E}[I^2]} = \frac{(\mathcal{E}[I_d] + I_a)^2}{\mathcal{E}[I_d^2] + 2I_a\mathcal{E}[I_d] + I_a^2} \\ &= \frac{3}{4}\frac{(2\pi L_a + L_d)^2 \sin^4(\alpha)}{3\pi L_a(\pi L_a + L_d)\sin^4(\alpha) - L_d^2\cos^3(\alpha) + L_d^2}\end{aligned} \tag{3.12}$$

which can be further simplified by noting that κ actually depends on the ratio of light source intensities $L_a/L_d = r$ and not their absolute values. After substituting $L_a = rL_d$ into Eq. (3.12) and simplifying we arrive at:

$$\kappa(\alpha) = \frac{3}{4}\frac{(2\pi r + 1)^2 \sin^4(\alpha)}{1 + 3\pi r(\pi r + 1)\sin^4(\alpha) - \cos^3(\alpha)} \tag{3.13}$$

To get a better intuition for κ, we consider what happens to κ under two special cases, $L_d = 0$ and $L_a = 0$, which correspond to $r \to \infty$ and $r = 0$ respectively:

$$\kappa|_{L_d=0} = 1 \qquad \kappa|_{L_a=0} = \frac{3}{4}\frac{\sin^4(\alpha)}{1 - \cos^3(\alpha)}$$

In other words, if there is no directional illumination component (i.e., $L_d = 0$) then $\kappa(\alpha)$ is always 1, and α cannot be recovered from pixel measurements alone.

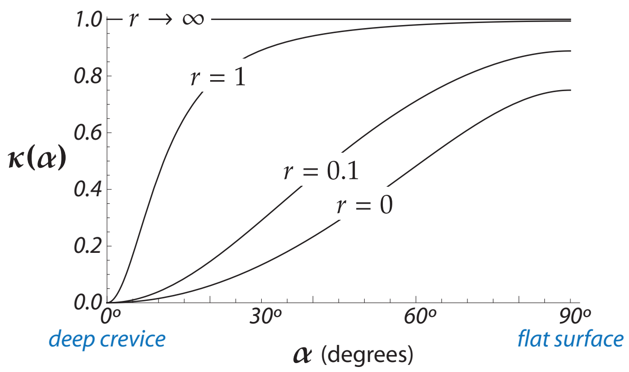


Figure 3.4: $\kappa(\alpha)$ for different ratios of ambient to direct light r. Note that as $r \to \infty$ ($L_d = 0$) we have a constant curve ($\kappa(\alpha) = 1$) so information about α cannot be recovered.

This case corresponds to all images in our stack being identical, with the scene lit only by an ambient term, so there is no variation in intensity for each point. In this case there is no way of directly disambiguating between shading and reflectance.

If there is no ambient component (i.e., $L_a = 0$) then κ increases monotonically in the valid range for α and is independent of L_d (as long as $L_d > 0$). In Figure 3.4 we show $\kappa(\alpha)$ for a few different values of r.

One interesting property of the curves in Figure 3.4 is that they have different κ values for $\alpha = 90^\circ$, ranging from 0.75 to 1 as r goes from 0 to ∞. This means that if we know that a given point in our scene is not occluded by any other geometry (i.e., $\alpha = 90^\circ$), then we can recover r directly from the value of κ for that point from Eq. (3.13):

$$r(\kappa)|_{\alpha=90^\circ} = \frac{\sqrt{3}\sqrt{\kappa-\kappa^2}+3\kappa-3}{6\pi(1-\kappa)} \tag{3.14}$$

In summary, we have derived a relation between the statistic κ, and the ambient occlusion at a point, using a physical model of a crevice (with a cone of

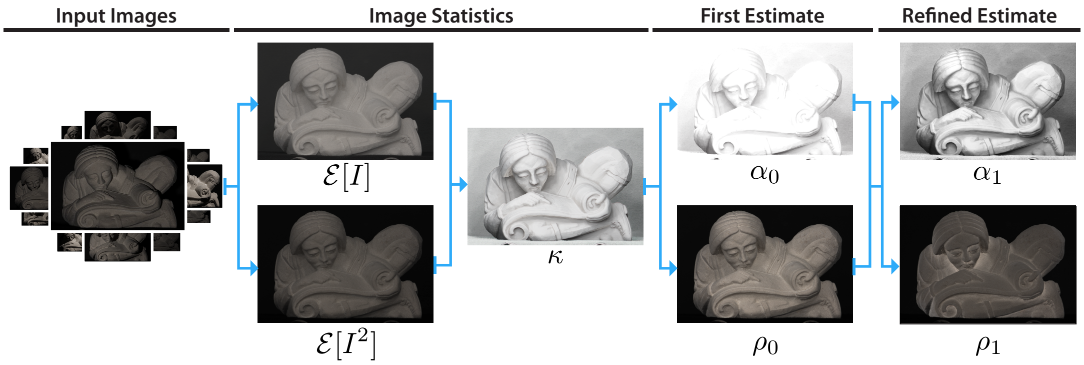


Figure 3.5: A depiction of the full algorithm for computing the local visibility angle α and the reflectance ρ. Arrows show how information flows in our pipeline. Starting with an image stack we compute $\mathcal{E}[I]$ and $\mathcal{E}[I^2]$, which are used to compute κ. We then proceed to obtain a first estimate of the local visibility angle and reflectance, which are then refined using a non-linear optimization.

visibility characterized by α) lit by a varying directional light, and a constant ambient light over a stack of images. No assumptions of smoothness or geometric reconstruction are required to derive this parameter. As we show later, this physical model, though simple and an approximation of real scenarios, works surprisingly well in characterizing the visibility at points in a scene.

3.3 Algorithm

In this section we use our model to compute a per-pixel local visibility angle $\alpha(x)$ and albedo $\rho(x)$ given a stack of images of the same scene under varying illumination. While our derivation has assumed grayscale images, our algorithm makes use of additional constraints from the three different color channels; while we solve for a color albedo and a per-color-channel value for r, α is constant for a given point, and the r variables are assumed constant over the image stack and across pixels. Our full algorithm is described below, and illustrated in Figure 3.5.

We first compute κ using Eq. (3.3) by assuming $r_0 = 0$ (i.e., ambient lighting is negligible) to derive an initial α_0 using Eq. (3.13). We then refine $\alpha(x)$ (one value per pixel) and r (one value per color channel, but constant across pixels) by minimizing the objective function:

$$\alpha_1, r_1 \leftarrow \min_{\alpha,r} \sum ||\kappa_{obs} - \kappa(\alpha_0, r_0)||^2 \tag{3.15}$$

where the subscript *obs* stands for "observed". In other words, we compute α and r so as to best explain the observed statistic κ. In total we have $n_c \times n_p$ equations, where n_c is the number of color channels and n_p the number of pixels, and $n_p + n_c$ variables, one α per pixel and n_c variables corresponding to the direct to ambient illumination ratios r. Eq. (3.15) defines a non-linear least squares problem, which we minimize using the trust-region-reflective mode of MATLAB's `lsqnonlin` function.

Given our final estimates α_1 and r_1, we compute estimates for the albedo $\rho(x)$ at each point from Eqs. (3.10) and (3.9). We express albedo as a function of the expected pixel value, the ratio r, the local visibility angle α, and the intensity L_d of the direct component:

$$\rho = \frac{2\mathcal{E}[I]}{L_d \sin^2(\alpha)\,(1 + 2r\pi)} \tag{3.16}$$

Note that there is an inherent ambiguity between light source intensity L_d and the scene albedo, so we can only estimate albedo up to a scale factor. Therefore, we assume that $l_d = 1$ to obtain ρ_1, our final estimate of the albedo.

3.4 Results

We begin by demonstrating results for our algorithm on various datasets (Section 3.4.1) and exploring the different outputs the algorithm produces. In Section 3.4.2 we use an object with known geometry to measure the error in our estimate of ambient occlusion. In Section 3.4.3 we evaluate our estimate of albedo by comparing our algorithm with others using the MIT Intrinsic Images benchmark [32]. Finally, Section 3.5 provides a detailed analysis of various aspects of our algorithm on a specially manufactured test object with crevices of varying (and known) depth; this includes an analysis of convergence rate as the number of images grows, and the impact of error factors such as interreflections.

3.4.1 Image Decomposition

Figure 3.6 shows image decomposition results on several datasets, including image stacks used in prior work. For each dataset we show ambient occlusion, reflectance ρ, and the illumination.

Datasets. The first dataset, TENTACLE, contains 350 images of a 3D printed object with known geometry. The light source position in TENTACLE was precisely controlled by a mechanical gantry allowing us to sample uniformly random positions over the full hemisphere. The known geometry lets us compare against ground truth ambient occlusion.

The other datasets are public datasets that violate the assumptions of our model in various ways. FROG and SCHOLAR, from [87], contain 47–48 images lit under varying directional lights that do not cover the full hemisphere. FACE

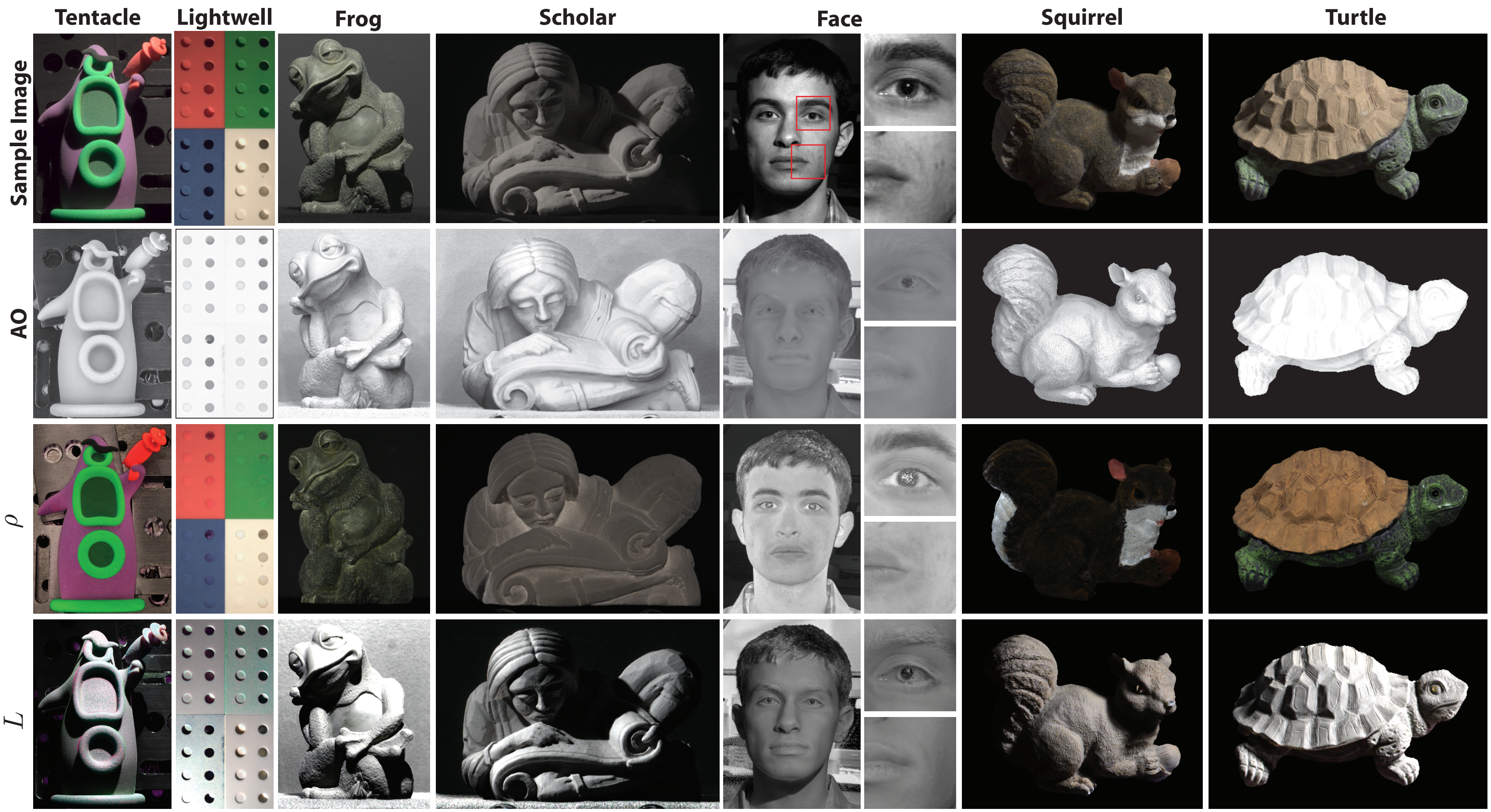


Figure 3.6: Results of our algorithm (2nd estimate). Each column shows results from a different dataset. The rows show 1) sample images from the original dataset, 2) our estimated AO, 3) albedo, and 4) the illumination in the sample image.

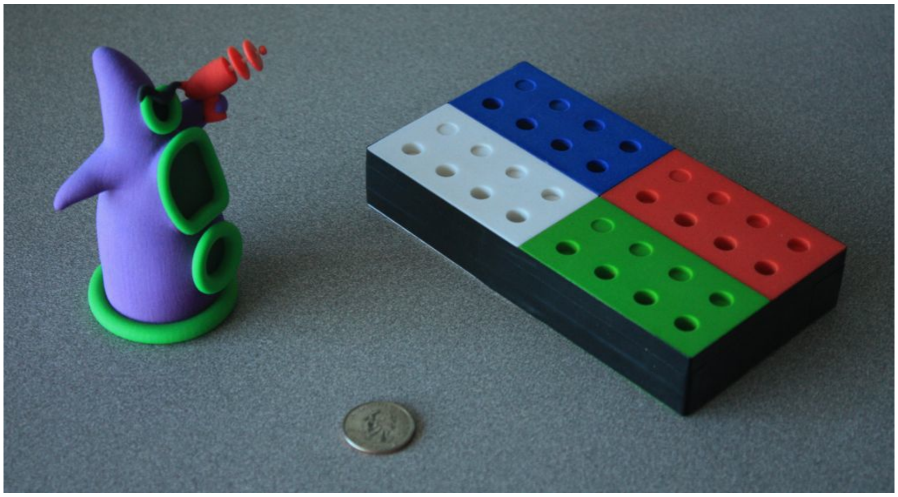


Figure 3.7: 3D printed test objects TENTACLE and LIGHTWELL, together with a quarter dollar coin (for scale). Black tape surrounding LIGHTWELL was added to reduce subsurface scattering that resulted from light shining on the side of the object.

from the Yale Face Database B+ [56], contains 64 images with light positions over a range of angles. This scene violates our assumptions in that skin is not strictly Lambertian, and exhibits significant subsurface scattering. Nevertheless we see from the images for AO and L in Figure 3.6 that our technique can qualitatively separate geometry and reflectance quite well. In particular, one can see from the area around the mouth that our AO image does not contain texture due to facial hair. Finally, we show results for TURTLE and SQUIRREL, from the MIT Intrinsic Image Dataset. Here the main challenge is that there are only 10 images of each object lit by a point light source.

Discussion. Figure 3.6 shows that the recovered AO seems to match our expectation of local visibility for these scenes. The recovered albedos are mostly free of shading and the ambient occlusion map is mostly free of albedo (e.g., the frog's nose and the mouth in FACE). It is also interesting that the pupil in the FACE dataset is black in the AO image and a light gray in the albedo.

3.4.2 Ambient Occlusion

We validated our estimate of AO using two objects of known geometry. In addition to TENTACLE, we 3D printed another object with a more regular shape, which we refer to as LIGHTWELL (Figure 3.7). This object is a solid block of material with a series of cylindrical holes of varying but known depth [1]. We printed this object in four colors: white (original material color), red, green, and blue to evaluate the impact of different albedos on our estimates. The acquisition setup for LIGHTWELL is the same as for TENTACLE (see Section 3.4.1). It is worth mentioning that although 3D printing offers good control over the geometry, material properties cannot be fully specified. The selected material (sandstone) was the most diffuse of the available materials, but is still not perfectly diffuse, and exhibits a fair amount of subsurface scattering (see the red ray gun of TENTACLE).

Figure 3.8 compares our AO result for TENTACLE to the ground truth (a more detailed analysis for the LIGHTWELL object is presented in Section 3.5). We can see qualitatively that our estimate of AO is very similar to ground truth. One difference is that our estimate appears smoother; we believe that this is caused in part by subsurface scattering, as the effect is most noticeable in the thin areas of the gun. Another difference is that our estimate is in general darker, meaning that our algorithm is predicting that locally the geometry is more occluded than really is. We attribute this in part to the material roughness from the 3D printing process. At a meso-scale level the structure can be thought of as being composed of many small crevices, and a single pixel in our κ image is an average of all these contributions.

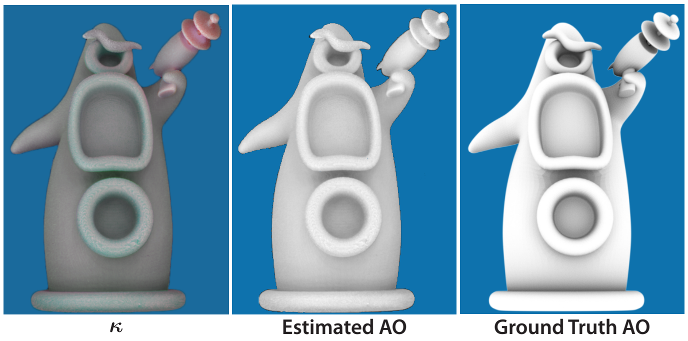


Figure 3.8: Left: the statistic κ computed for TENTACLE. Right two images: Comparison of estimated AO with ground truth (computer-generated). The background clutter is masked.

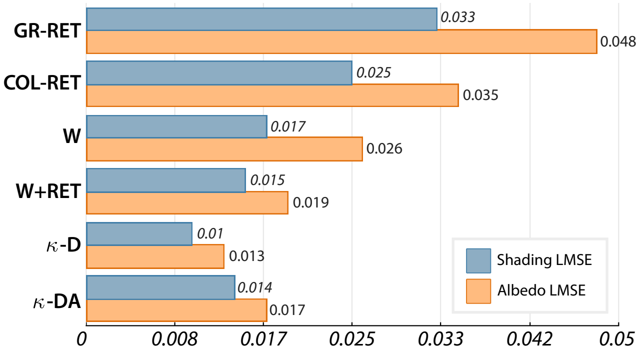


Figure 3.9: Comparison of LMSE error on the MIT intrinsic image dataset [32] (shorter bars are better, indicating less error). Compared algorithms are: Grayscale Retinex (GR-RET), Color Retinex (COL-RET), Weiss (W), Weiss+Retinex (W+RET), ours with only direct term (κ-D) and our second estimate containing direct and ambient terms (κ-DA).

3.4.3 Albedo

We ran our algorithm on the MIT Intrinsic Images benchmark [32] to measure the quality of our albedo estimates. This benchmark consists of 16 objects each with 11 images, and uses the local mean squared error (LMSE) defined in [32] to evaluate performance. Some methods evaluated by the benchmark (e.g., Retinex) operate on a single image, usually by imposing priors on the illumination and albedo images or by using heuristics to classify gradients. However, the best-performing reported prior method operating on multiple images combines Retinex [53] with Weiss's method [89] which, like our own, requires a stack of images.

We obtain the shading image for each of the input images by simply dividing the input image by our estimated albedo (see Eq. (3.4)). Figure 3.9 shows our method's performance compared to others included in the benchmark. In Figure 3.10 we show a subset of results against the Weiss+Retinex multi-image method. We note that our approach outperforms the competing methods. Interestingly, our initial estimate (i.e., $r = 0$) performs better than our refined one. We believe that this is a result of the setup, which indeed contains no ambient illumination (as assumed by the first estimate of our algorithm, but not by our refined estimate, leading to overfitting), and the fact that most objects have a very high albedo, resulting in a larger contribution due to interreflections, which are not modeled by our algorithm. For completeness, in Table 3.1 we also compare our method to recent single-image algorithms [6, 80, 82], and report results on the different subsets of the benchmark dataset used in each prior evaluation. Our method compares favorably to these methods (but also uses more than a single image).

3.5 Analysis

In this section we present a more in-depth analysis of various aspects of our algorithm on the specially created LIGHTWELL object (Fig. 3.7). This object has a very regular shape, with cylindrical holes of various depths that match our

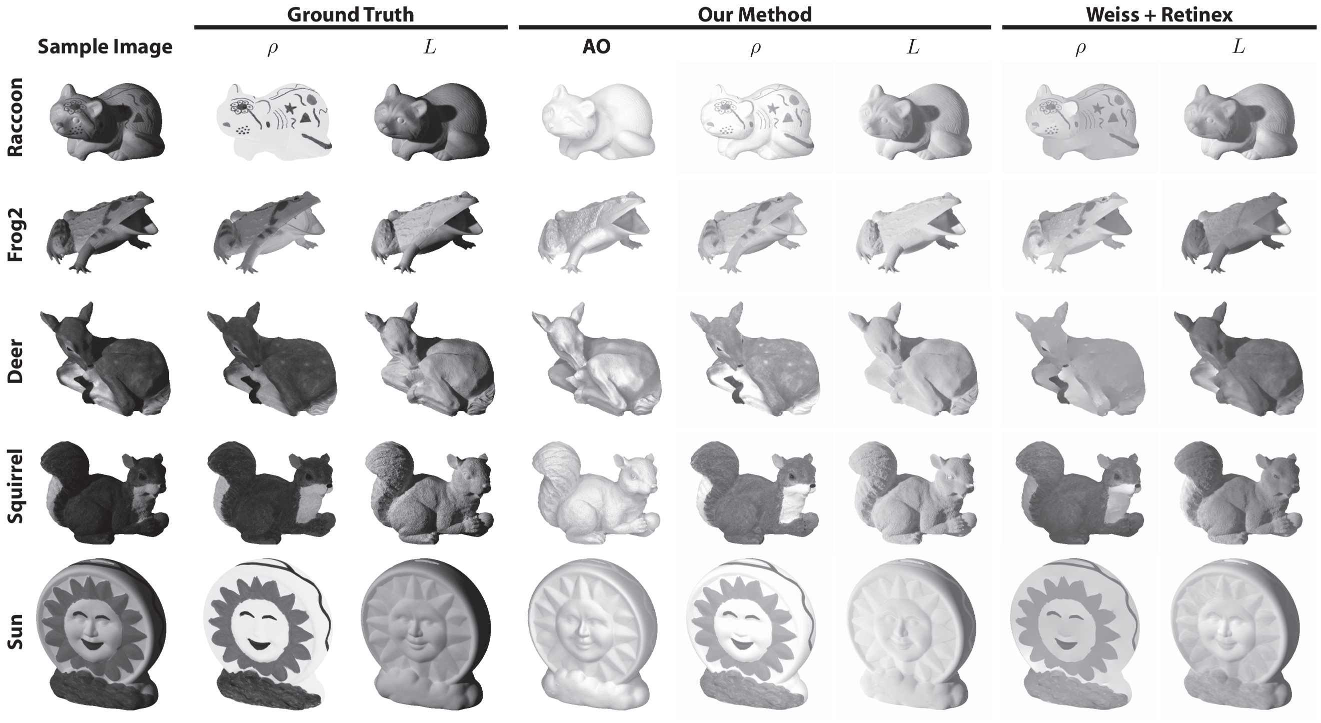


Figure 3.10: Comparison of our method with W+Ret from the MIT benchmark. Results are for our first estimate of the albedo (i.e., ambient illumination is assumed to be zero) as this gave us the best results on the benchmark. We show here grayscale images as the benchmark uses grayscale versions of the decomposed images in its evaluation metric.

Table 3.1: Local mean squared error (LMSE) for individual images of our algorithm for the 1st (only direct light) and 2nd estimates (direct and ambient term), together with results from other work when available. In the last four columns of the last row we show our average on the same subset of images as reported by [6, 80, 82, 32, 10]. On all cases our algorithm outperforms these prior methods. Note that the numbers marked with † are the geometric mean as reported in [10], other averages are arithmetic means.

	Ours (1st estimate)			**Ours (2nd estimate)**							
	refl	**shading**	**avg**	**refl**	**shading**	**avg**	**Weiss + Retinex [32]**	**Barron and Malik [6]**	**Shen and Yeo [82]**	**Shen, Yang, Li, and Jia [80]**	**Barron and Malik [10]**
apple	0.006	0.0060	0.006	0.006	0.006	0.006	0.016			0.010	
box	0.004	0.0040	0.004	0.005	0.005	0.005	0.010		0.002	0.011	
cup1	0.003	0.0020	0.002	0.003	0.002	0.002	0.005		0.004	0.005	
cup2	0.003	0.0010	0.002	0.003	0.001	0.002	0.002	✓	0.005	0.007	✓
deer	0.027	0.0160	0.021	0.037	0.021	0.029	0.043	✓		0.032	✓
dinosaur	0.015	0.0120	0.014	0.016	0.007	0.012	0.015			0.021	
frog1	0.020	0.0180	0.019	0.029	0.026	0.027	0.043		0.053	0.029	
frog2	0.056	0.0120	0.034	0.053	0.017	0.035	0.053	✓	0.043	0.024	✓
panther	0.008	0.0060	0.007	0.024	0.014	0.019	0.005		0.008	0.005	
paper1	0.004	0.0040	0.004	0.010	0.008	0.009	0.003		0.001	0.013	
paper2	0.007	0.0040	0.006	0.009	0.006	0.008	0.005	✓	0.003	0.016	✓
pear	0.006	0.0050	0.005	0.006	0.004	0.005	0.006	✓		0.010	✓
phone	0.011	0.0080	0.010	0.035	0.013	0.024	0.008			0.011	
potato	0.011	0.0080	0.009	0.006	0.006	0.006	0.010	✓		0.014	✓
raccoon	0.011	0.0090	0.010	0.015	0.011	0.013	0.005	✓	0.005	0.008	✓
squirrel	0.019	0.0240	0.022	0.020	0.025	0.023	0.027			0.037	
sun	0.004	0.0050	0.005	0.007	0.005	0.006	0.003	✓	0.002	0.007	✓
teabag1	0.007	0.0160	0.012	0.012	0.033	0.023	0.014	✓	0.027	0.063	✓
teabag2	0.003	0.0110	0.007	0.012	0.020	0.016	0.006		0.015	0.031	
turtle	0.017	0.0200	0.019	0.020	0.026	0.023	0.015	✓	0.017	0.025	✓
average	0.012	0.0095	0.011	0.016	0.013	0.015	0.015	0.019	0.015	0.019	0.021†
our avg on same subset							0.011	0.012	0.010	0.011	0.009†

physical model, allowing us to evaluate in more detail how different aspects of our model impact the performance of our algorithm.

3.5.1 Impact of Albedo on Ambient Occlusion Estimates

Because the 3D printed LIGHTWELL object consists of four different albedos, we can obtain a quantitative error measure of the local visibility angle α for different albedos. We report this error in Figure 3.11, computed by measuring the average error for α at the center of the crevice for LIGHTWELL compared to ground truth, for varying α angles corresponding to the crevice depths for the printed object. This figure shows four curves, one for each color of LIGHTWELL. In the plot three trends are evident. First, the error is larger for brighter albedos (red and white, in this case). We suspect that this is due to the increase in light interreflections for higher albedos. Since our model does not account for this effect, a patch at the bottom of a deeper hole looks brighter than our model would predict.

Second, we note that error increases for the more shallow crevices. We suspect this is due to roughness in the printed object as discussed in Section 3.4.2.

A third trend is that deeper holes have the largest errors. This can be explained by the fact that κ is the quotient of two expectations, and that for regions that receive light less frequently, we expect these averages to stabilize more slowly, a property we examine next.

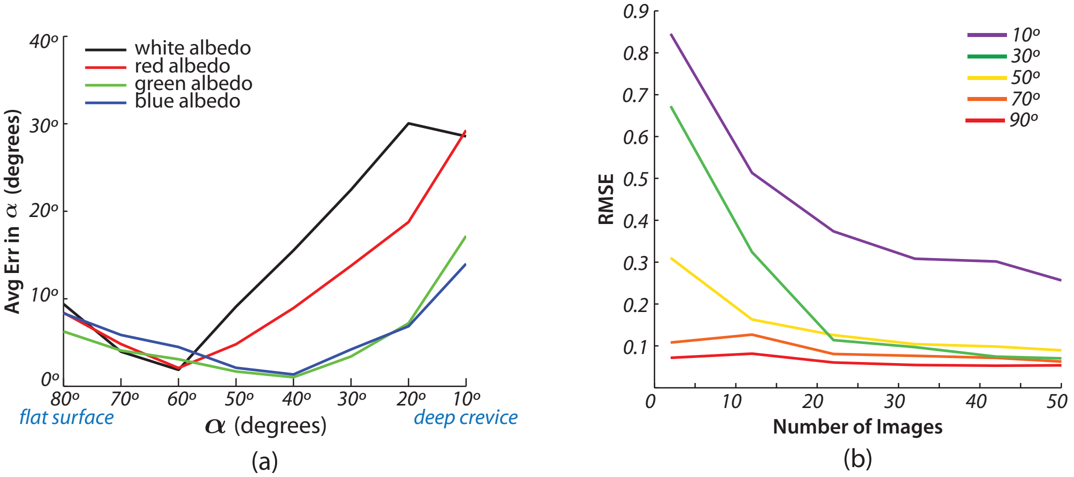


Figure 3.11: (a) Error in the estimated local visibility angle α vs. the true local visibility angle for the LIGHTWELL object printed in different colors (shown in the left). (b) Average Root Mean Square Error (RMSE) for our estimate of ambient occlusion vs. number of images used in the estimate. Each curve represents a different crevice depth and a corresponding local visibility angle α.

3.5.2 Convergence Rate

We now consider the impact of the number of images and the visibility angle in estimating ambient occlusion. Figure 3.11 shows the root mean squared error (RMSE) of our ambient occlusion estimate as a function of the number of input images for different crevice depths (and hence local visibility angles). For each hole depth, we estimate AO at the center of the hole using rendered images of the blue LIGHTWELL (generated using a physically based renderer [38]). We compare our estimate to the ground truth AO in that hole using MSE, and repeat this process 100 times (sampling different light source positions at random each time) to compute an average RMSE. We observe that rate of convergence is strongly dependent on the depth of the crevice, but our method performs well even with a relatively small number of images on scenes where $\alpha \geq 40^{\circ}$.

3.5.3 Impact of Interreflections

To understand how global illumination affects our method, we conducted a simple experiment with computer-generated images of LIGHTWELL. The object, which was rendered with a perfect diffuse material and a reflectance of 0.5, was captured with an orthographic camera that aims directly towards the crevices, while illuminated with an ideal directional light source plus an ambient term (with $r = 0.25$). We used a physically based renderer [38] to produce 1000 images sampling the light direction uniformly at random over the hemisphere. Two sets of images were rendered: one with only the direct (single-bounce) component of light (that is, with interreflections and other indirect effects disabled in the rendering) and the other with both direct and indirect illumination components.

The error in the estimate of AO at the center of each crevice is shown in Figure 3.12. For the ideal case (only direct component of illumination) the ambient occlusion is in general very close to ground truth, with a max absolute error of 0.0172 (where the max possible error would be 1.0). When indirect illumination is present the local visibility angle is overestimated for the holes in the center of the range, reaching an error of 0.0753 for $\alpha = 30°$. This happens because at the middle of the α range the contribution from the indirect light is closest to that of direct light, which means that the discrepancy between our model and what is observed is at its maximum. At the extremities of the range two different phenomena decrease the effect of global illumination. For the shallower crevices most light paths that reach the bottom of the crevice are direct ones, decreasing the relative effect of indirect illumination. For deeper crevices, on the other hand, most paths that reach the bottom of the crevice do go through multiple bounces of light. Nevertheless, at each bounce the light is attenuated by the cosine factor (due to the angle of incidence and the surface normal) multiplied by the albedo, so by the time it reaches the bottom of the crevice it is attenuated so much that the total contribution from all indirect paths is still much smaller than that of the direct ones.

Albedo also plays a role in the error when global illumination is present. In this experiment we used $\rho = 0.5$. For larger values of albedo the mode of the error will shift towards deeper crevices, because the light is attenuated less after each bounce.

3.5.4 Color in κ

When introducing κ in Section 3.2 we focused on monochromatic images; we now discuss a property of the statistic κ that arises when dealing with color images. In this case κ is computed independently for each channel resulting in one r per channel: r_R, r_G, and r_B for the red, green, and blue color channels. In Eq. (3.5) we showed that the statistic κ is independent of the albedo ρ, yet a red tint appears on the ray gun of TENTACLE (Fig. 3.8 left). What does this color reveal about lighting? Because the direct component of light in our setup was white, color in κ is due to an ambient term.

Let's focus on a single point in the scene. Geometry at a point is the same across color channels so we can restrict our analysis to a single angle α without loss of generality. For instance, in Fig. 3.4 let's focus on $\alpha = 90°$. If the ambient term is blue, with $r_R = 0$, $r_G = 0$, and $r_B = 0.1$, what we will see in κ is that $\kappa_R = \kappa_G < \kappa_B$, so the color in the κ image reflects the color of the ambient term.

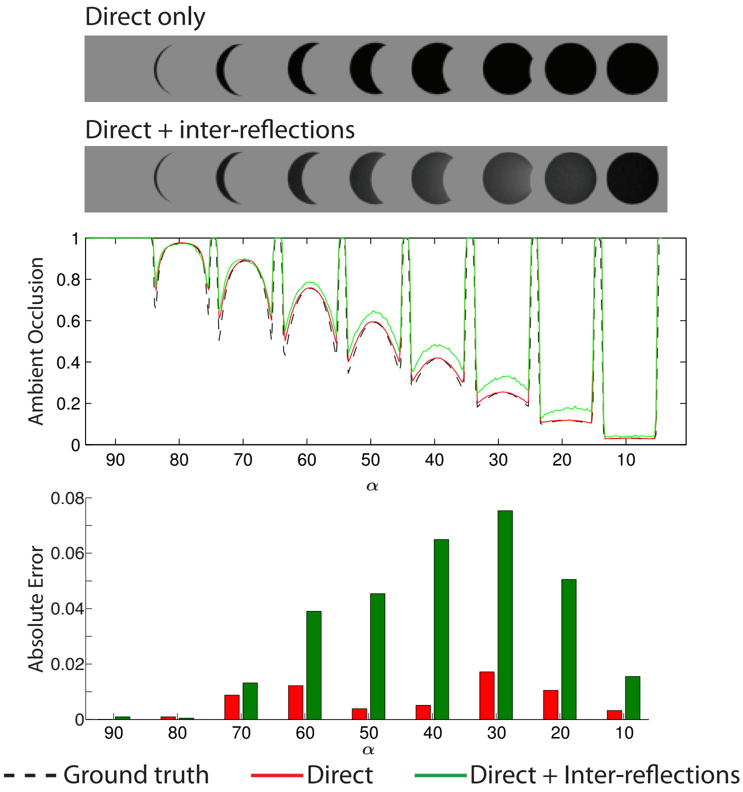


Figure 3.12: Impact of interreflections in our estimates. Top: two renderings of LIGHTWELL, only the direct component of light and then with interreflections included (notice that shadows become brighter due to reflected light). Middle plot shows a cut through the rendered LIGHTWELL ambient occlusion estimate versus ground truth. Interreflections cause our method to over estimate ambient occlusion. Bottom bar plot shows the absolute error in the estimated ambient occlusion at the center of the crevices.

This allows us to estimate the hue of the ambient term directly from κ. In the acquisition setup for Fig. 3.8 there was no ambient light; only a direct source was present. We believe that the red tint in the κ image on the ray gun is due to interreflections and subsurface scattering. Even thought this source of light is not constant, it varies much slower that the direct term so it can be thought of as a "local ambient" term.

CHAPTER 4

REASONING ABOUT PHOTO COLLECTIONS USING MODELS OF OUTDOOR ILLUMINATION

Natural illumination plays a critical role in the appearance of outdoor scenes, and in the variation of scene appearance over time; for example, see Figure 4.1 for images from a photo collection of the Statue of Liberty illustrating appearance changes under different illumination conditions. Many vision tasks, such as photometric stereo and instrinsic image decomposition, require reasoning about this illumination and how it interacts with the scene.

Although outdoor illumination is highly variable, it is far from arbitrary; in fact, it is dominated by a few elements—sun, sky, and weather—which in turn depend fundamentally on scene location, time, and atmospheric conditions. Indeed, the computer graphics community has developed increasingly sophisticated models of outdoor illumination that, given parameters such as geolocation and time, compute a predicted outdoor environment map. Surprisingly, such illumination models are not yet widely used in computer vision despite illumination's importance in the appearance of outdoor scenes.

The work in this chapter explores the connection between community photo collections of an outdoor scene at a given location on Earth, and the distribution of lighting conditions for that scene predicted by these illumination models. The usage of these predictive models to reason about scenes from unstructured photos is still a major challenge, in part because timestamps are often missing or erroneous—community photos represent a "soup" of different observations of the scene under varying but unknown illumination. Our insight is to match *statistics* of outdoor illumination with pixel statistics derived from photo

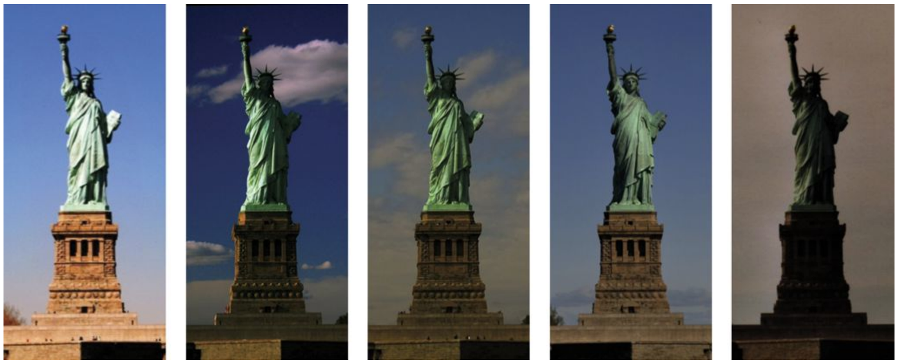


Figure 4.1: The Statue of Liberty under a variety of natural illumination conditions

collection. We build on the photometric ambient occlusion work presented in Chapter 3, which explored the connection between pixel statistics and simple illumination distributions in relation to the local visibility (or ambient occlusion) of each scene point. This chapter generalizes the model to handle the more realistic scenario of varying illumination in outdoor scenes, a challenging setup for the method presented in Chapter 3.

This chapter includes an analysis of how the geographic position, surface normal, and local geometry of a point interact with illumination models, and how multiple measurements of a point's appearance over time can be used to estimate albedo and local visibility for points in a scene. Since our photometric approach relies on varying illumination, we analyze the conditions under which a lack of sufficient variation can arise, and use this analysis to detect areas of insufficient variation in illumination in a given scene (e.g., surface points that are almost always pointed away from the sun). Hence, in addition to estimating scene albedo, we also estimate areas where such estimates are unreliable.

Our albedo estimation has further practical value in estimating sun positions in uncalibrated Internet photos with missing or erroneous timestamps (timestamps and sun position being two sides of the same coin). Such estimated sun positions are useful in the analysis of outdoor illumination in photos, such as in photometric stereo, shadow detection, or grouping photos by light similarity. Timestamps are useful in correcting clocks on consumer cameras, and discovering patterns of photography (e.g., which time of day is most popular for taking photos of a given landmark).

4.1 Modeling Illumination for Outdoor Points

As discussed above, outdoor illumination exhibits great variability, but is nonetheless highly structured. The illumination reaching a point in an outdoor scene is influenced by a few key factors. The primary source of illumination during the day is the sun, whose position in the sky is a function of **geographic location** and the **time and date** of an observation. The location and date constrain the sun position to a well-defined path, while the time of day determines where the sun lies on that path. We denote location using latitude (ϕ) and longitude (λ), and time and date as t. The intensity and color of the sun, as well as the light from the sky caused by atmospheric scattering, vary as a function of both sun position and weather.

Weather adds immense complexity to outdoor illumination; the degree of variation increases greatly with the variety of clouds, fog, haze, and other atmospheric effects. State-of-the-art outdoor illumination models largely ignore weather and assume clear skies; since we are taking advantage of these models, we leave the incorporation of more varied weather conditions as future work. Furthermore, clear, sunny skies provide the most informative illumination for photometric methods such as ours that rely on varying and strongly directional

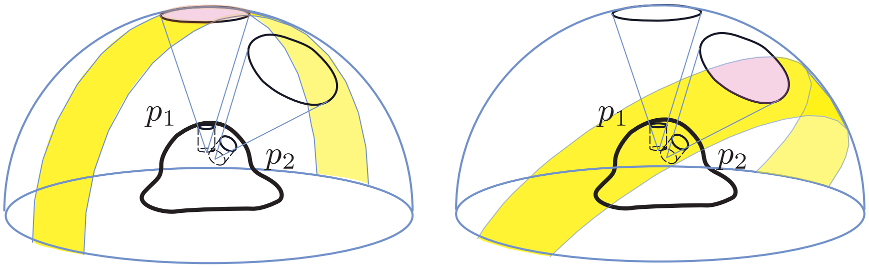


Figure 4.2: Influence of geolocation, date and time, orientation, and local visibility on the illumination at a point. On the left, an object located near the equator sees a band of sun paths (shaded in yellow) that is centered directly overhead. In this location, the point p_1 at the bottom of a crevice can sometimes see the sun, whereas p_2 cannot. On the right, we see a location farther from the equator, with a different sun path, where the reverse is true.

illumination. We discuss later how to use weather records to discard cloudy images.

Having discussed the factors affecting the illumination coming from the sun and sky, we now consider scene-related properties that affect how much of the light hits a given point in a scene. The **surface orientation** at a point affects how much and which portion of the sky's illumination reaches it. For instance, if the normal is facing away from the sun's path it will never receive direct sunlight (see Figure 4.2). Further, a point in an open field pointing upwards towards the sky will see the entire sky dome, while a point facing downward will see less of the sky dome and more of the ground.

Finally, the illumination arriving at a point can be affected by its **local visibility**—the extent to which surrounding geometry occludes its view of the sky dome. As a simple way to describe the potentially complex local geometry around a point, we adopt the crevice model from Chapter 3. By modeling local geometry as a single cylindrical hole, we are able to describe the extent of occlusion using a single parameter, α, representing the angle from the point's surface normal to the opening of the crevice (see Figure 3.3).

In summary, our model considers the illumination of an outdoor scene point on a clear day as a function $L(\phi, \lambda, t, \alpha, \vec{n})$ where (ϕ, λ) are the geographic latitude and longitude, t is the time and date, $\vec{n}$ is the normal vector, and α is the local visibility angle given by our crevice model. To make predictions based on our model, we use the physically-based sun/sky model proposed by Hosek and Wilkie [36], which produces a sunny environment map given geographic location and time of day (ϕ, λ, t). We can then choose any surface normal and visibility angle $(\vec{n}, \alpha)$, and integrate the irradiance over the visible portion of the environment map to acquire a value for $L(\phi, \lambda, t, \alpha, \vec{n})$.

4.2 Albedo and Sun Position in Photo Collections

In this section, we describe how to use the model described above to estimate local visibility (ambient occlusion) and albedo of scene points; we then describe a method for using the albedo to estimate the illumination and timestamp of individual photos.

Our method takes as input a set of photos of an outdoor scene from different viewpoints and varying, unknown times.[1] We first create a sparse 3D reconstruction using SfM and multi-view stereo, and then georegister the reconstructed scene. We project each reconstructed point into the images in which it appears to retrieve a set of observed color values for that point. The geo-

[1] Later, we assume we know the date for each photo, but not the time of day. This is consistent with our experience of errors in image timestamps.

registered model gives us the location (ϕ, λ) and surface normal $(\vec{n})$ for each point in the scene.

4.2.1 Estimating Albedo for Sunlit Outdoor Scenes

We assume that all surfaces are Lambertian and that a point's albedo does not change over time. Our method works on color images by treating each channel independently, so for simplicity we refer only to intensity. We use a simplified image formation model where a single observation of a point x is given by:

$$I_x = \rho_x L(\phi_x, \lambda_x, t, \alpha_x, \vec{n}_x) \tag{4.1}$$

where I_x is the observed intensity for scene point x in image I, and ρ_x denotes its albedo. If we could find accurate values for ϕ_x, λ_x, t, α_x, and $\vec{n}_x$, we could recover the albedo ρ_x by dividing the observed intensity by the predicted illumination. The 3D reconstruction provides values for ϕ_x, λ_x, and $\vec{n}_x$, but local visibility α_x remains unknown since we are dealing with sparse point clouds. Furthermore, since the images come from Internet photo collections we generally have unknown or uncertain time t. For this reason, we cannot directly predict illumination for a single image in practice.

However, we have many observations of x across different images, which can provide insight about the *distribution* of intensity values observed at that point. Likewise, our lighting model can be used to predict the expected distribution of illumination conditions over the course of a year. Building upon the method proposed in Chapter 3, we match predicted statistics to observed statistics in order to estimate the local visibility and albedo of each point in the scene. Given many images that view x distributed over the year, we can estimate the expected intensity of x, $\mathcal{E}[I_x]$, by averaging the observed samples. If ρ is constant over time, then Eq. (4.1) implies:

$$\mathcal{E}[I_x] = \mathcal{E}[\rho_x L(\phi_x, \lambda_x, t, \alpha_x, \vec{n}_x)] = \rho_x \mathcal{E}[L(\phi_x, \lambda_x, t, \alpha_x, \vec{n}_x)] \tag{4.2}$$

The expectation above is computed over all light source positions, which in the case of outdoor illumination, equates to time. Therefore $\mathcal{E}[I_x]$ is independent of time.

Suppose that we know x's local visibility angle, say $\alpha_x = 90°$. In this case, we have all the information we need to compute $\mathcal{E}[L]$ using the sun/sky model, and we can compute ρ_x as:

$$\rho_x = \frac{\mathcal{E}[I_x]}{\mathcal{E}[L(\phi_x, \lambda_x, t, 90°, \vec{n}_x)]} \tag{4.3}$$

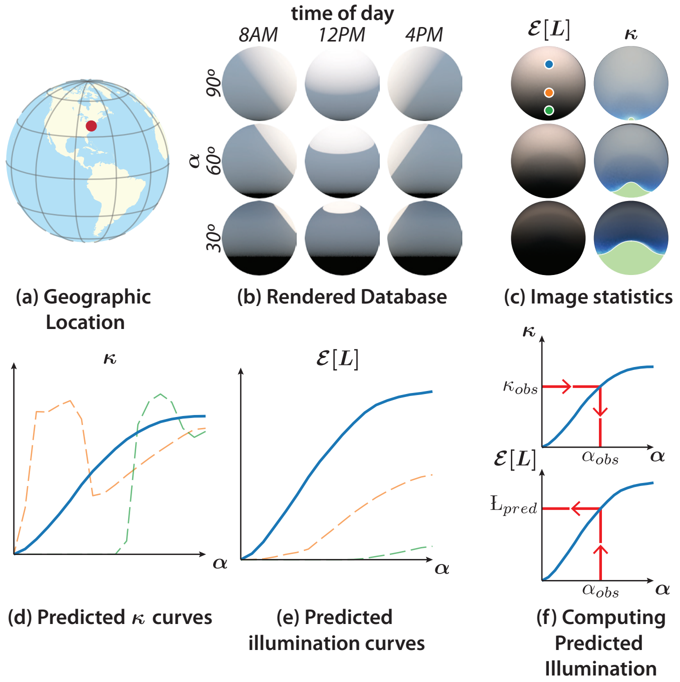
where the expectation of L is computed over a set of times t sampled throughout a full year.

4.2.2 Estimating Local Visibility Angle

We can now compute albedo for a point if its local visibility angle is known, but in practice α_x is unknown and must be estimated as well. In Chapter 3 we proposed a technique to estimate α directly from image observations by computing a statistic over image observations I_x that is independent of ρ:

$$\kappa_x = \frac{\mathcal{E}[I_x]^2}{\mathcal{E}[I_x^2]} = \frac{\mathcal{E}[L_x]^2}{\mathcal{E}[L_x^2]} \tag{4.4}$$

By assuming point-source illumination that moves uniformly over the hemisphere, we derived an analytical relationship between κ and α, which allowed us to compute α based on the observed value of κ.



(a) Geographic Location

(b) Rendered Database

(c) Image statistics

(d) Predicted κ curves

(e) Predicted illumination curves

(f) Computing Predicted Illumination

Figure 4.3: Pipeline for albedo estimation. Given a geographic location (a), we tabulate $L(\phi, \lambda, t, \alpha, \vec{n})$ over all parameter values (b), where we show one sphere per combination of α and time of day, with each point in the sphere representing a different normal direction $\vec{n}$. We then compute $\mathcal{E}[L]$ and κ for each α and $\vec{n}$ (c), producing the curves for $\kappa(\alpha)$ in (d) and $\mathcal{E}[L](\alpha)$ in (e) for each normal. For an observed value of κ, we look up α and then predicted average illumination $\mathcal{E}[L]$ (f), allowing us to estimate albedo. Green regions in (c) correspond to combinations of $\vec{n}$ and α for which we cannot reliably recover albedo.

Under our more sophisticated illumination model, a closed-form relationship is harder to find; κ now depends on location, α, and $\vec{n}$. However, the model we introduce in this chapter allows us to predict, for a given scene location, the expected value of κ over all normals and visibility angles. In particular, for a fixed location (ϕ, λ), we can compute the relationship between κ, α and $\vec{n}$ as:

$$\kappa(\alpha, \vec{n}) = \frac{\mathcal{E}[L(\phi, \lambda, t, \alpha, \vec{n})]^2}{\mathcal{E}[L(\phi, \lambda, t, \alpha, \vec{n})^2]} \tag{4.5}$$

where expectations are computed over light source position/time.

For a given point x with normal $\vec{n}_x$, we compute its observed κ value κ_x using Eq. (4.4). The visibility angle α_x is chosen to be the value of α such that the predicted $\kappa(\alpha_x, \vec{n}_x)$ most closely matches the observed κ_x. Figure 4.3 (c) shows images of κ and expected illumination $\mathcal{E}[L]$ for several α angles; (d) and (e) show examples of κ and predicted illumination curves for three different normals (colored points marked in (c)). For a monotonically increasing κ curve such as the blue curve in Figure 4.3(d), we can simply take the observed value κ_x and look up the corresponding value of α to assign an estimated local visibility.

Sun Visibility and κ

Let's analyze the $\kappa(\alpha)$ curves for normals that do not directly face the sun path, such as the orange and green curves in Figure 4.3(d). Note that they do not increase monotonically over all values of α as is the case with the blue curve (which corresponds to a normal that does face the sun path). In particular, we see the curves go very quickly from 0 to approx. 1, and then go down again and begin a slow monotonic increase after a certain α. The case $\kappa(\alpha) = 0$ occurs for crevices that never receive direct sun light, so $\mathcal{E}[L] \approx 0$. While $\kappa(\alpha) = 1$ occurs when $Var[L] \approx 0$ and $\mathcal{E}[L] > 0$, i.e., crevices that receive very little direct sun light[2].

Because these curves are not invertible we cannot obtain a single α given κ. We deal with this issue by identifying the crevice angle α_{min} bellow which a point at the bottom of the cylindrical crevice either receives no direct sun light or so little reaches it that $Var[L] \approx 0$ (in practice we set α_{min} based on the fraction of the time that the sun directly illuminates the bottom of the crevice). When finding

[2] A brief discussion of these two cases is presented in Chapter 3

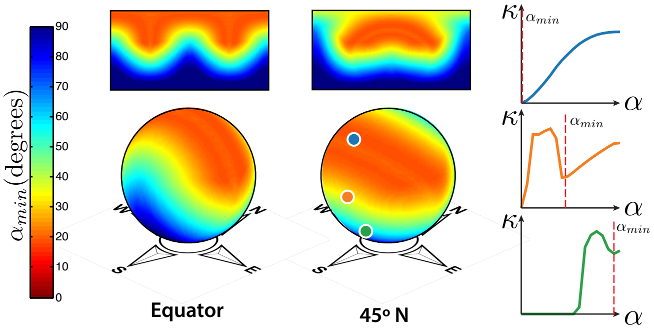


Figure 4.4: How α_{min} varies with geographic latitude and normal. Each point on the sphere represents a surface normal direction, and its color encodes the α angle of a crevice that sees the sun 10% of the time during daylight. Normals pointing near the sun path (as determined by latitude) have lower α_{min} values and are more informative because κ is meaningful over a greater range of α. The rightmost column shows κ curves and α_{min} for the three different surface normals.

α given κ we discard the portion of the $\kappa(\alpha)$ curve corresponding to $\alpha < \alpha_{min}$, leaving only the slowly increasing monotonic regions of the curve (which are invertible), this implicitly assumes that crevices deep enough that their opening is smaller than α_{min} do not occur in the scene. For certain normal directions, ones facing further away from the sun path, α_{min} is large enough that it is no longer reasonable to assume that deep crevices with $\alpha < \alpha_{min}$ do not occur in the scene. We deal with this by discarding normals for which $\alpha_{min} > \alpha_0$, where α_0 determines the maximum depth of a crevice in the scene. It is important to note that this limitation, imposed by lack of variability in illumination, will affect any photometric method, because the observations lack sufficient information to disambiguate between albedo and illumination.

4.2.3 Estimating Time of Day

We now consider the task of determining the time of day a given image was captured. This is a problem of practical importance for Internet photo collections as it is very common for images to have the incorrect time stored as metadata (e.g., caused by camera owners that travel to different timezones but forget to set the camera time). Recovering this information can help reveal patterns of human activity around monuments, and since it is linked to the sun position it can be used to estimate material and geometry of the scene. We do assume that the associated date is roughly correct. This constrains the problem and is more forgiving of errors as sun position varies much less with date than with time of day.

Our strategy is to use the computed albedo ρ_x to estimate the lighting for a set of visible points in the scene, we then compare such a lighting estimate for a single image to a set of predicted illumination conditions over a range of times and choose the timestamp where the lighting matches most closely. This is to some extent similar to the mechanism used by sundials to determine time. One fundamental difference is that sundials rely on the position of cast shadows, while our method relies on the shading caused by attached shadows to determine the position of the sun in the sky.

Under our simple image formation model we can compute lighting by simply dividing the observed intensity by the albedo:

$$L_x^{obs} = \frac{I_x}{\rho_x + \epsilon} \tag{4.6}$$

where ϵ is a small constant to achieve robustness to noise. We collect the estimated illumination for all visible points in an image I into a vector $\vec{L}_I^{obs}$, and generate a corresponding predicted illumination vector $\vec{L}_I^{pred}(t)$ for each hypothesized timestamp t using our model.

We found it important to perform a normalization before comparing illumination vectors to increase contrast and overcome noise in our lighting estimates. The most effective strategy was to normalize each vector so that the bottom and top 10 percentiles span the range $[0, 1]$. We compute the cost $c(I, t)$ for time t as a robustified L^2 distance: we sort the element-wise differences $\vec{L}_I^{obs} - \vec{L}_I^{pred}(t)$ and discard the top and bottom 10%. Our final cost function is:

$$c(I, t) = ||R(\vec{L}_I^{obs} - \vec{L}_I^{pred}(t))||_2 \tag{4.7}$$

where $R(\cdot)$ is the robustification operator above. The need for robustness in this distance measure is mainly due to phenomena not captured by our model, such as cast shadows (where L^{obs} is darker than L^{pred}) and specular highlights (where L^{obs} is brighter than L^{pred}). Finally, the timestamp for an image is chosen by finding the time t that minimizes the cost $c(I, t)$.

4.3 Implementation details

We start with a large collection of Internet photos, and use SfM to obtain camera extrinsics and intrinsics along with a sparse set of 3D points [4], the model then is manually georegistered. We then use PMVS to compute a larger point set with surface normals [27] and recompute the visibility list for each camera using a z-splatting algorithm (to increase the number of observations per point). We approximate the response of all cameras as a gamma curve with $\gamma = 2.2$, as done in prior work [33] (in Chapter 5 we use more sophisticated models for the tone mapping curve). We also discard pixels that are too bright or too dark.

Because current sun-sky illumination models are limited to clear skies, we restrict the input to our albedo estimation phase to images taken on days with limited cloud cover. For STATUE, we use weather records provided by NOAA [65] to select days when cloud cover was no greater than 25%. CASTLE had very few cloudy images, so weather-based pruning was unnecessary. When tabulating the values of L we set the ground albedo in [36] to 0.15.

Until now we have assumed a uniform distribution of timestamps among our input images. However, any known distribution can be modeled by weighting the predicted illumination values before computing statistics. For TENTACLE, we used only the L values from the times when images were captured. To better match the distribution found in STATUE and CASTLE, we weight all predicted illumination values from each date by the number of input images captured on that date.

4.4 Experiments and Applications

Evaluating our method is a challenging task, as ground truth albedo and timestamps are difficult to acquire for photo collections. To evaluate our method in a more controlled setting we created the TENTACLE dataset with 100 images of a 3D-printed object taken outdoors over the course of a sunny day. We also created an analogous synthetic dataset, TENTACLER, by rendering the same object using a physically-based renderer under the sun-sky model [36] at times sampled throughout a day.

We also gathered two photo collection datasets from Flickr, STATUE and CASTLE. STATUE contains 78K images of the Statue of Liberty in New York, USA. CASTLE contains 33K images of a theme park attraction in Florida, USA. Ground truth timestamps for 347 CASTLE images were manually entered by reading the time from a clock in the scene. For STATUE we evaluate our timestamping method on an additional set of 265 images from the AMOS webcam dataset [37] with known ground truth timestamps (STATUEA). STATUEA is distinct from STATUE in that we did not use it for albedo estimation.

4.4.1 Albedo

A comparison of the albedo obtained with the technique proposed in Chapter 3 (ρ-unif) against the technique proposed in this chapter (ρ-sunsky) for the TENTACLE dataset is shown in Fig. 4.5. ρ-sunsky recovers a significantly flatter albedo and successfully identifies and discards points which cannot be recovered accurately. We measure the Local Mean Squared Error (LSME) [32] of ρ-sunsky and ρ-unif and verify that ρ-sunsky has a lower error of 0.0303 versus 0.0586.[3] We also computed LMSE on the result of ρ-unif with the mask generated by ρ-sunsky (which discards normals for which $\alpha_{min} > \alpha_0$), showing that a significant improvement can be made, with the error dropping to 0.0447, by identifying for which points the albedo cannot be estimated accurately using photometric methods, as discussed in our analysis in Section 4.2. The method we introduce in this chapter, ρ-sunsky, which performs best, combines this analysis with our more realistic lighting model.

[3] The TENTACLE "ground truth" is the albedo sent to the 3D printer, but colors are not reproduced perfectly in 3D printing. Therefore, we use the LMSE on grayscale images to evaluate the piecewise constant albedo without penalizing the color mismatch.

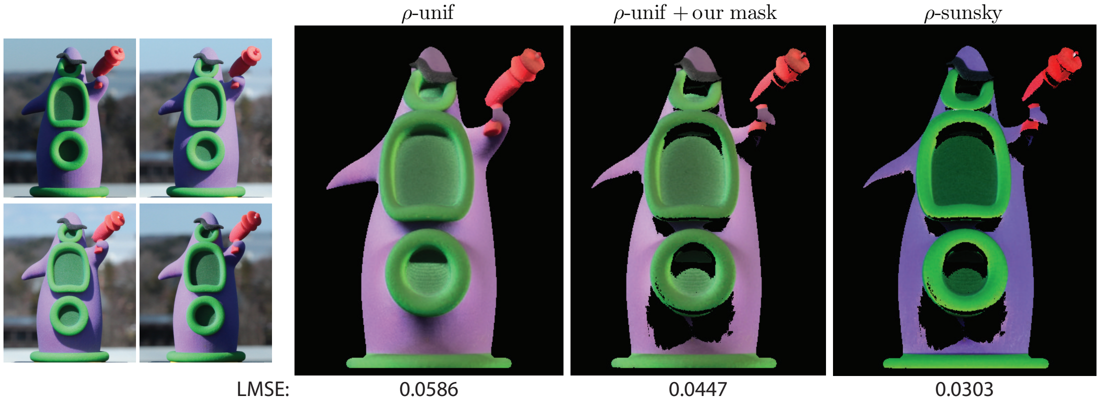


Figure 4.5: Results of ρ-sunsky compared to ρ-unif, with and without the mask generated using ρ-sunsky (which discards normals for which $\alpha_{min} > \alpha_0$). We also list the Local Mean Squared Error for each result.

4.4.2 Timestamps

Quantitative results on the timestamping task are shown in Table 4.1. To illustrate the impact of the albedo in the timestamping task we run our timestamping method with the two different albedos ρ-sunsky and ρ-unif, shown in the table as t-sun and t-unif. As baselines we also compare to raw Exif timestamps (CASTLE only) and "chance" error (Rand), the average expected error given by guessing a random time between sunrise and sunset. Note that because Rand is restricted to daylight hours its average error is actually lower than that of Exif timestamps, which are totally unconstrained and reflect the data that came from the images.

In the table we see that ρ-sunsky performs best on all datasets, with errors increasing as the data becomes less structured, starting with a median error of 9.8 minutes for the computer generated dataset TENTACLER all the way to 57.3 minutes median error for the Internet photo collection CASTLE. The most

Dataset (# images)	TENTACLER (200)		TENTACLE (100)		STATUEA (265)		CASTLE (347)	
	Avg	Med	Avg	Med	Avg	Med	Avg	Med
Rand	248.6	237.0	218.2	208.6	230.8	216.3	231.6	219.4
EXIF	–	–	–	–	–	–	287.5	211.0
LEN	–	–	–	–	249.3	223.0	195.0	150.0
t-unif	27.6	29.3	58.0	61.8	133.3	80.5	114.7	74.4
t-sun	9.9	9.8	53.1	46.9	136.9	49.3	87.0	57.3

Table 4.1: Average and median timestamp error (in minutes) for various methods on our datasets. Rand represents chance over daylight hours, while LEN is the method of Lalonde et al. [50]. Sun position error in degrees can be calculated approximately by dividing minutes by 4.

significant increase in error occurs when going from TENTACLE to TENTACLER, when median error jumps from 9.8 to 46.9 minutes, this could indicate that there are still significant differences between the true natural illumination and that provided by Hosek and Wilkie [36] given that both datasets are very similar and ground truth geometry is known in both cases.

The difference between *t*-sun and *t*-unif is smaller on Internet datasets, where noise in the input data affect *t*-sun more. For example, *t*-sun relies on the surface normal estimates from multiview stereo, which we have observed to contain significant noise. Another source of error on Internet datasets is tone mapping, which we have modeled using a simple gamma curve. Future improvements in radiometric calibration and better normals will improve results for *t*-sun. Finally, a common failure case occurs when our algorithm mistakenly assigns a cloudy image to either sunset or sunrise, when shadows are also very diffuse.

We significantly outperform the method of Lalonde et al. [50], whose lowest median error of 150 minuntes is achived on CASTLE against 57.3 minutes for *t*-sun. This demonstrates the value of using many images to reason about a scene.

It is worth noting though that the two methods are orthogonal in the information they use to estimate sun position. Our method requires sparse geometry and large photo collections, which enable it to make precise measurements of the shading in the scene. Lalonde et al. on the other hand operate on a single image and use cues such as the color of the sky and the position of shadows cast by pedestrians. Therefore, one could potentially combine the two methods to obtain even better results.

Figure 4.6 shows sample results, including an input photo along with the scoring curve used to determine the timestamp. For an alternative visualization of our timestamping results see Fig. 4.7, where we show the average of all images for a given time (after registration using a homography) as estimated using either raw Exif tags or our method.

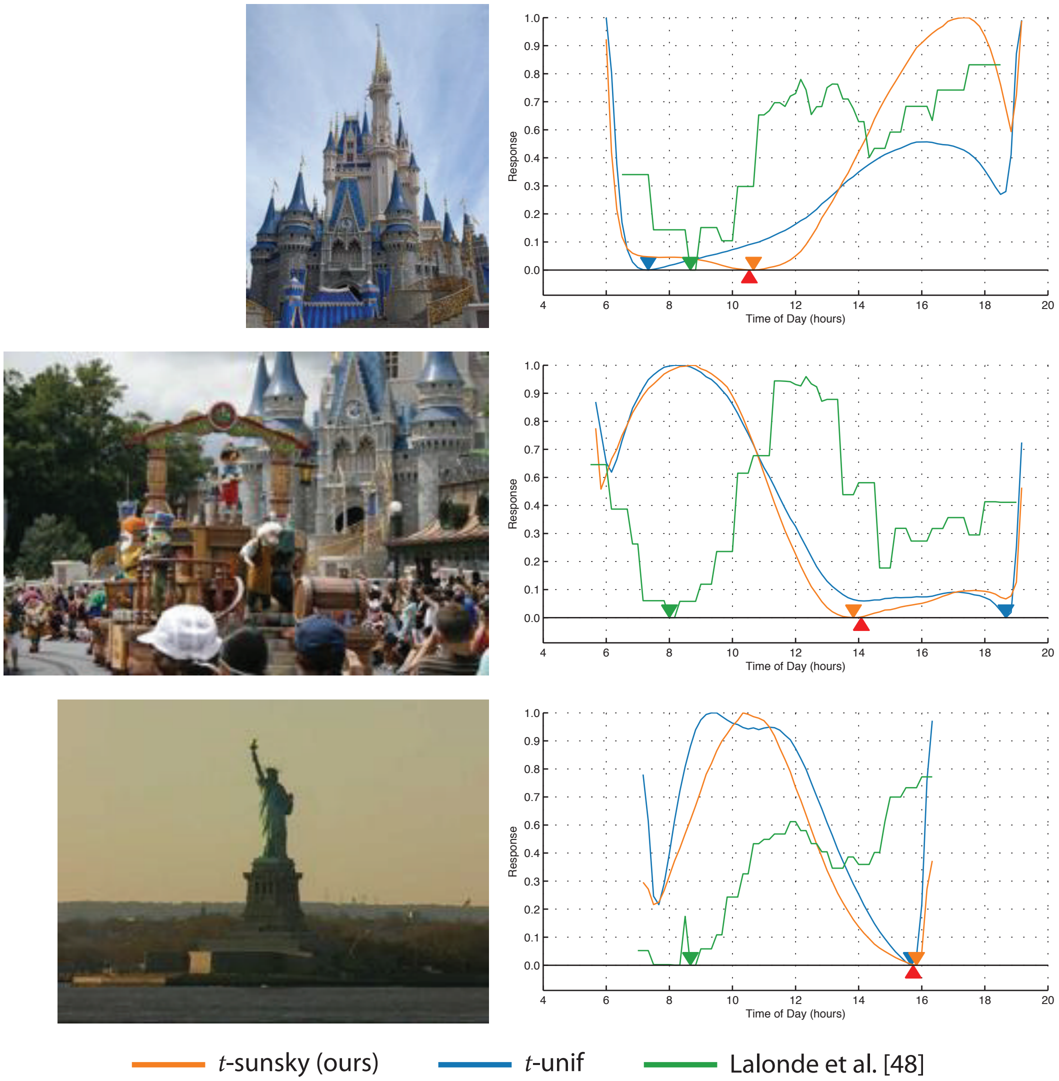


Figure 4.6: Examples of timestamping results. The plot on the right shows timestamp error metrics over the full day for the image on the left. Colored triangles show the global minimum (estimated timestamp) for each method, and the red triangle indicates the ground truth timestamp.

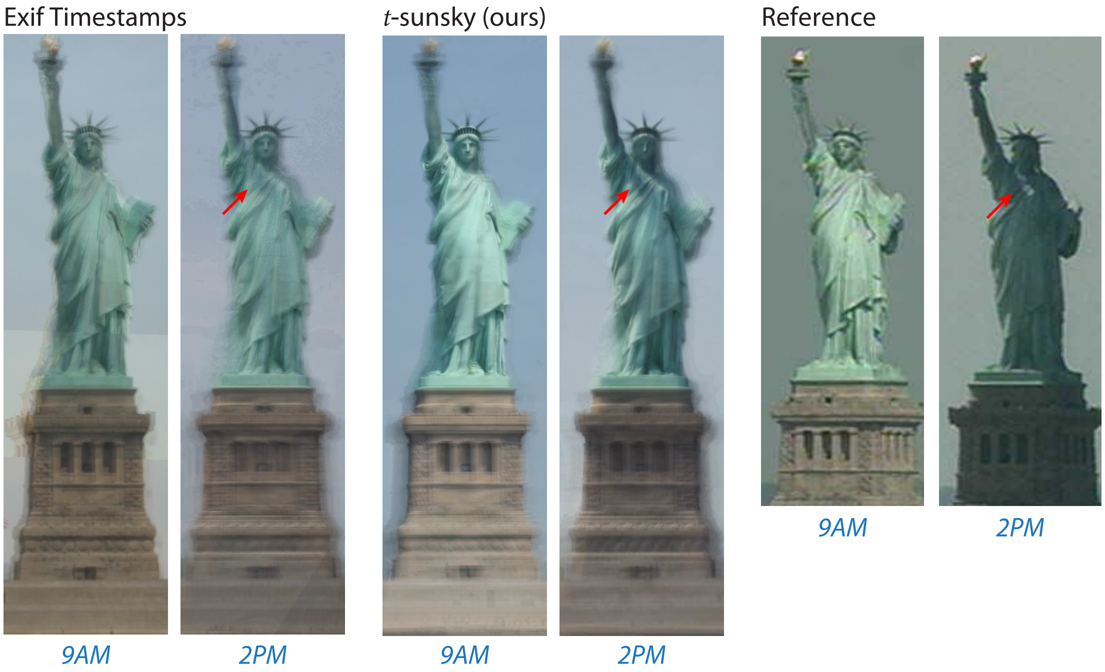


Figure 4.7: Alternative visualization of timestamping results. We select a set of images taken in July with a given timestamp, reproject them to a single viewpoint using a homography, and average them. We show that Exif timestamps (left) do not produce coherent lighting when averaged due to the timestamp errors. Using our timestamps (middle), the average images match the lighting in the corresponding reference images (right), taken from the AMOS dataset where timestamps are known. Note highlighted cast shadow at 2PM and the clear change from 9AM to 2PM.

CHAPTER 5
LIGHT DESCRIPTORS IN PHOTO COLLECTIONS

In this chapter we move our study of light further, starting with an estimate of the scene illumination in an image from a photo collection, using the methods introduced in Chapters 3 and 4, we then proceed to distill this information into a compact representation, effectively creating a light descriptor. This characterization of a scene, which is separate from other properties such as scene geometry or reflectance, enables us to reason about changes in lighting that go beyond the timestamping application we described in Chapter 4. We demonstrate that this descriptor enables indexing and searching for images by illumination, classification of images based on weather conditions, and virtual object insertion. It is also conceivable that this representation could drive other powerful computer graphics applications for Internet photos, such as image relighting, and plausible object cut-and-paste between images.

A key question is one of representation: what kinds of descriptors can we compute from the available data, and which representations are best for various applications? We evaluate a number of representations, including average irradiance maps, spherical harmonics, and a representation that captures spatial variation in illumination over a scene. Further, we use our representation of irradiance to compute more accurate environment maps by leveraging predictive sun-sky models. We show that our new light descriptor representations are simple to derive from Internet photo collections of scenes.

5.1 Light Descriptors

This chapter focuses on computing light descriptors for images, where a *light descriptor* is a compact representation that summarizes how light is distributed in the underlying scene as captured in the photo. In particular, given a collection of images of a scene, we seek to compute a light descriptor for each image in the collection as a vector. Various types of descriptors have been used to characterize overall image appearance, such as color histograms [75] and GIST descriptors [66]. However, we seek to capture the essence of *illumination*, factoring out scene appearance, and modeling important effects such as directionality and color of light. There are many potential applications for such descriptors, including lighting-based search for other images with similar illumination (within the same scene, or across scenes), inferring the location of a dominant light source (the sun) in outdoor images, weather classification (sunny vs. cloudy), among other tasks.

We assume that for each image we have estimates of the irradiance and surface normal for each pixel in a subset of image pixels (Section 5.2 describes how we derive this information). Note that in this chapter we use “points” and “pixels” more or less interchangeably, where it is understood that a 3D point in a scene maps to a 2D pixel in an image via perspective projection. We assume primarily Lambertian scenes, although our methods are largely robust to sparse sets of pixels that violate this assumption (e.g., occasional specular highlights).

5.1.1 Concept

At a high level, a descriptor should be the most compact representation of a signal that is necessary to perform a certain task. It should be invariant to features of the original signal that are not relevant for its intended application. For instance, local image features like SIFT [59] are designed to capture the appearance around an image point, while remaining invariant to scale, rotation, and (to a lesser extent) projective distortion and changes in lighting. In our application, we are interested in matching the lighting conditions of a scene, so the invariants are very different. In particular, we would like to design a descriptor with the following properties:

- **Invariance to geometry and material.** We want to be able to match illumination between images, even if the scenes or materials pictured in the images are very different.
- **Captures important effects.** We want our descriptor to capture effects such as the direction and color of the illumination.
- **Simple to compute.** Our descriptor should be simple and robust to compute from an image collection.
- **Widely applicable.** The descriptor should be general enough to enable a variety of tasks that depend on illumination.

5.1.2 Representation

How should we *represent* the illumination evident in an image? Here, it is instructive to consider representations for illumination used in computer graphics, such as environment maps, light fields, and surface light fields [90]. As a starting point, one way to define a lighting descriptor is as an estimate of the environment map from an image, i.e., a map of radiance indexed by surface orientation. If we had a chrome sphere (i.e., a light probe) in our photos, then computing such an environment map would be straightforward. In the absence of a light probe, however, for most scenes, computing a full-resolution environment map from a photo is an extremely ill-posed problem, because high-frequencies in the environment are lost when reflected from the diffuse surfaces that characterize much of the outdoor world. Instead, we seek to compute an estimate of the *irradiance environment map* (or diffuse environment map) [72].

Note that radiance and irradiance maps are ideal for capturing information about distant light sources, such as the illumination provided by the sky. In order to capture local illumination other representations such as light fields are necessary. These representations though require significantly more information and are much harder to acquire. Therefore, in this chapter we focus on irradiance maps.

Average Irradiance Map (A-IM)

The irradiance environment map records the irradiance from distant illumination at a scene point, indexed by surface normal. For an image I, and for each point $p \in \mathcal{P}(I)$, where $\mathcal{P}$ denotes the set of scene points and $\mathcal{P}(I)$ the set of scene points visible in I, given our estimates of irradiance $L(p)$ and surface normal $N(p)$, and assuming all illumination is distant (and there are no local lighting effects, such as cast shadows or interreflections), we have a direct measurement of the irradiance at $N(p)$. Thus, the descriptor can be computed as follows: First, discretize the space of surface normals into bins. Then for each normal bin *b*, find the subset of points *p* whose normal $N(p)$ lies in *b*, and compute the mean irradiance over that subset of points. This representation has an immediate interpretation as an irradiance environment map, but one with missing data. First, due to scene geometry some normal directions might simply not be present in the scene. Second, we can only observe normals that point toward the camera, so even if all normal directions are represented in the scene we can capture at most a hemisphere of directions. We call this descriptor the Average Irradiance Map (A-IM). Examples of this descriptor are shown in Figure 5.1.

Spherical Harmonic Compression (SHM). In addition to being easy to compute, irradiance maps are also very compact. In fact, Ramamoorthi and Hanrahan showed that for a Lambertian scene, any irradiance map due to distant lighting can be accurately represented using a small number of coefficients of a Spherical Harmonic (SH) basis [72]. This serves as a natural means for compressing our descriptor even further.

Fitting SH coefficients to our descriptors is an effective way to compress the descriptors; but it is also important to consider the effects of missing data on the fitting process. In particular, we found it important to store a mask that specifies which normal bins originally contained data. This mask is then used during reconstruction to avoid using parts of the sphere where the SH reconstruction "hallucinated" data even though there was none in the input. The combination of SH and mask is refered to as SHM throughout the text. Figure 5.2 illustrates the effects of SH compression on our descriptors. Note that the full-sphere reconstruction (right middle) becomes inaccurate in the portions of the sphere where no data was available in the input.

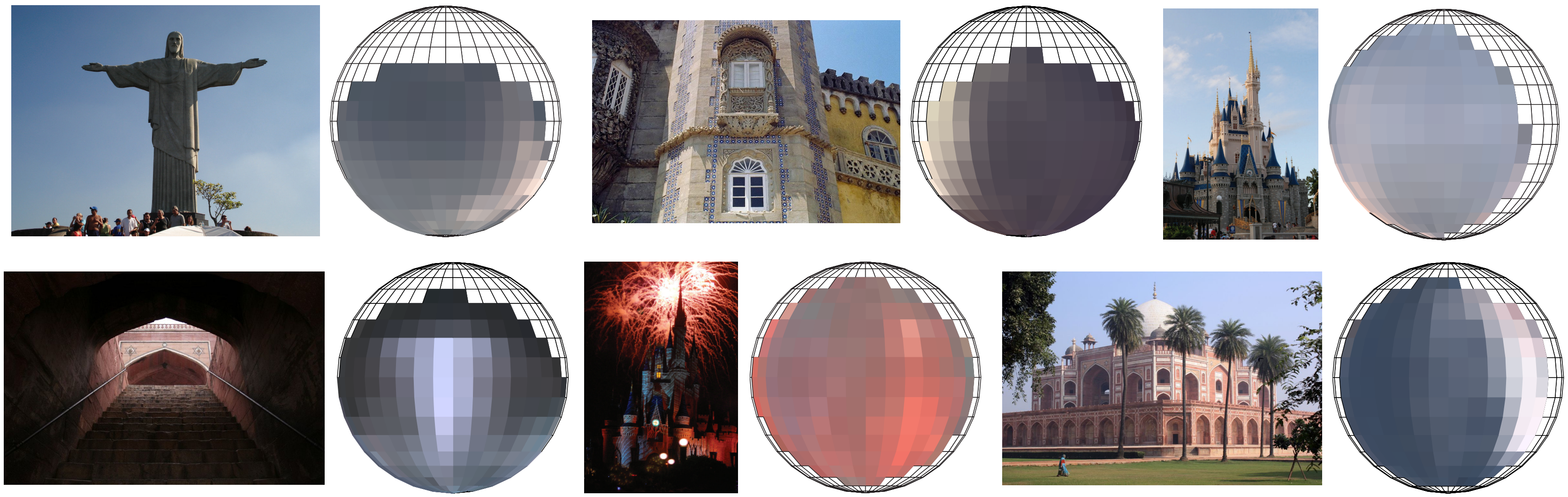


Figure 5.1: *Example lighting descriptors.* This figure shows example photographs from a range of scenes and illumination conditions, along with a visualization of the lighting descriptor we compute. Our descriptors are designed to be invariant to geometry and materials, and to capture important information about the illumination, such as directionality, color, and local effects. Note, for instance, for images with strong directional lighting from the sun, our descriptors are bright in the corresponding directions, and how the color of the surface itself (e.g., the yellow building in the upper middle image) does not appear in the descriptor. Our descriptors are simple to derive from structure-from-motion reconstructions from large Internet photo collections, and applicable in a variety of tasks.

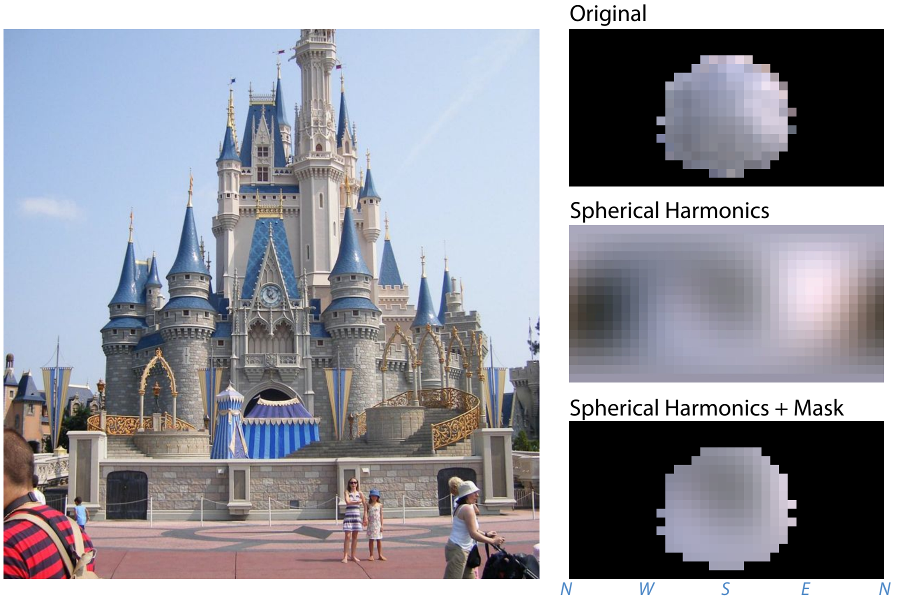


Figure 5.2: The effects of spherical harmonic compression on our descriptors. On the left, we show a sample image. On the right, we show the original descriptor as computed from the image (top), the full unwrapped spherical function reconstructed after fitting using nine spherical harmonic coefficients (middle), and the same reconstruction masked according to which normals were missing data in the input (bottom).

The key benefit of SHM over A-IM is space. A-IM stores 3 floats per normal bin, while SHM stores floats for 9 coefficients × 3 color channels (27 floats), plus 1 bit per normal bin for the mask.

Average-Variance Irradiance Map (AV-IM)

Averaging the irradiance values (as in A-IM) yields a simple, compact summary of the scene irradiance. However, this process conflates information from different points that share the same normal, thus losing potentially valuable information about lighting variation in the scene. For some applications, such as sun visibility classification, variance within a normal bin, (e.g., due to shadowing) can provide a useful cue. To capture this, we can store both the average and the variance per normal bin. This descriptor captures the extent of the variation of irradiance across points with the same normal while only using twice the memory as A-IM. We call this descriptor the Average-Variance Irradiance Map (AV-IM).

This idea of storing more than just one average per bin can be generalized further to store a more descriptive representation of the values in each normal bin, but descriptive power comes at a cost of increased storage space. For example, we experimented with storing a histogram of irradiances per normal bin to capture the full variation across normals, but we found the descriptor did not perform as well as the descriptors above, and so the extra space requirements were not justified.

5.2 Computing Light Descriptors from Photo Collections

We now describe how we estimate the light descriptors for images, starting from a large set of Internet photos $\mathcal{I}$. As done in Chapter 4, we first run structure from motion [85] for the photos and multi-view stereo (PMVS [26]) to derive a 3D reconstruction consisting of a point cloud $\mathcal{P}$ and 3D camera parameters for each image. PMVS also estimates a surface normal $N(p)$ for each 3D point $p \in \mathcal{P}$, as well as a set of visible points $V(I) \subset \mathcal{P}$ for each image I. After building a reconstruction, we manually georegister the scene to the correct world reference frame (latitude, longitude, orientation, and scale).

To compute our lighting descriptor for image I_i, as described in Section 5.1, the key information we need is the irradiance at a sparse set of pixels in I_i. In what follows, we assume that the image formation process for I_i is approximated by the equation

$$I_i(p) = f_i(\rho(p)L_i(p)) \tag{5.1}$$

where $I_i(p)$ is the measured radiance at the location of the projected point p in I_i (note that our notation $I_i(p)$ also incorporates the camera projection), f_i is the camera response function for camera i, $\rho(p) \in [0, 1)$ is the diffuse reflectance (albedo) at scene point p, and $L(p)$ is the diffuse irradiance at point p at the time the photo was taken.[1] The albedo of a point $\rho(p)$ is assumed constant across the image collection, but $L_i(p)$ varies from image to image. Under this model, to estimate $L_i(p)$, we need to determine both $\rho(p)$ and f_i. To do so, we: (1) use the entire image collection to estimate an albedo for each point $p \in \mathcal{P}$, (2) project the points in $V(I_i)$ into each I_i and look up their color values in that image, (3) estimate the camera response function f_i and invert it to map these color values to a linear space, and (4) divide these colors by albedo to derive irradiance. This pipeline is illustrated in Figure 5.3, and described in more detail below.

5.2.1 Estimating Camera Response

We first describe how we estimate the camera response function f_i for each image. This step is critical, as the algorithm we use to estimate albedo assumes a linear camera response. While others have posed radiometric calibration as a non-linear optimization over a specific image collection [23], we instead opt to pre-calibrate a large number of cameras via a data-driven approach based

[1] While this description treats I, ρ, and L as scalars per-point for simplicity, in practice we treat them as RGB values that multiply element-wise.

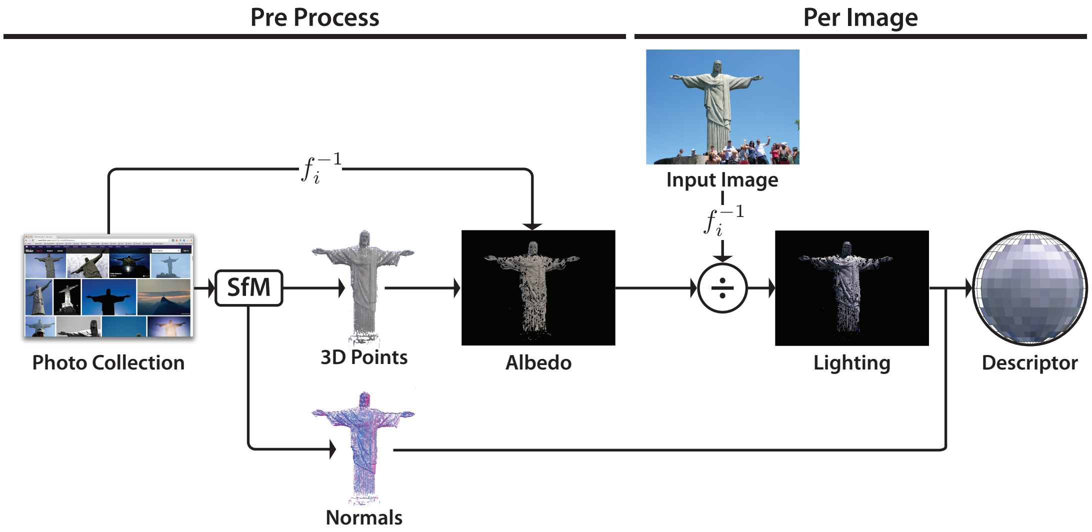


Figure 5.3: *Our pipeline.* Beginning with a large photo collection of a scene, we use structure from motion (SfM) to reconstruct a 3D point cloud with surface normals. We also assign a camera response curve to each photo in the collection, and use all of this data to compute a surface albedo for each scene point. These albedos can be factored out of a new photo to produce irradiance values, which, along with surface normals, are digested into our lighting descriptor. Here, the lighting descriptor captures the strong directional component of the illumination.

on [47].

A camera's response function can in general be very complex [43]: Chakrabarti et al. recommend the use of a model with 24 parameters [16]. In practice, a common approximation is to assume that $f_i(x) = x^\gamma$ with $\gamma = 2.2$ [33, 49]. We found that this approximation performed poorly, and we achieved better results with a data-driven calibration approach. Computing an accurate response function for a camera is a challenging problem, and many methods require controlled image acquisition of a static scene [21]. Internet photo collections are taken with too many cameras to rely on lab-calibrated response curves. Instead, we create a *hierarchy* of response curves, from a generic, "average" camera, to a set of specific camera makes and models using the method proposed by Kuthirummal et al. [47].

The algorithm of Kuthirummal et al. takes as input a large collection of natural images captured with the same camera model, and produces a response curve. The method relies on stationary statistics over many natural images and how camera response alters these statistics. To produce a good estimate, their method requires a large number of images, so newer or less popular cameras might not have enough images for a reliable estimate of the response function. This is where our hierachical structure of camera response functions comes into play. In particular, we organize camera response curves into a tree where each node corresponds to a curve. The leaves of this tree correspond to specific camera models (e.g., a Canon Digital Rebel T1i); higher nodes are computed with larger sets of images corresponding to all of the camera models rooted at that subtree. One level above the leaves, we have nodes that each represent a family of camera models (e.g., all Canon Digital Rebels), above that are all images from a given manufacturer (e.g., Canon), and at the root the response curve is computed from all images in our photo collection, resulting in a response curve for an "average" camera. We only create leaf nodes for camera models for which we have at least 1000 images in our training set. This approach assumes that the response curve for similar camera models is also in general similar.

We used this approach to build a collection of 205 camera response curves from 864,055 training images. Figure 5.4 shows the curve for the green channel for a specific camera. In the plot we show the curve for $\gamma = 2.2$, one we obtained from images acquired in a lab setting [16] using the method of [29], labeled *gt*, and the set of curves of our hierarchical structure, starting with the response computed from all images in the collection, *avgcam*, down to all images from the specific maker of the camera, *make*, to the most specific set of images, the leaf in our hierarchical structure, *make and model*.

To evaluate this method, we compared the response function at each level of the hierarchy with the curve obtained with the method of Grossberg and Nayar [29] on the 34 cameras present in the dataset of Chakrabarti et al. [16] (we discarded two cameras for which the exposure stack did not vary exposure, only aperture). We measured error as the area under the curve of the absolute difference between the two response functions. The average error for all images across all color channels is: 0.1886 ($\gamma = 2.2$), 0.1445 (*avgcam*), 0.1409 (*make*), 0.1426 (*make and model*). As the errors indicate in general it is sufficient to use the curve up to the *make* of the camera, the more specific *make and model* curve offers no benefit in our tests.

When linearizing a new photo, we use the most specific curve given the information available in that photo's Exif metadata. For images that have no information about camera make or model, we use the *avgcam*.

5.2.2 Computing Descriptors

Given a collection of linear images of a scene, a number of global algorithms can be used to estimate albedo [49, 78]. In this chapter, we use the algorithm described in Chapter 3 (we found that noise in the estimated normals from PMVS was severe enough that the method from Chapter 4 did not perform well).

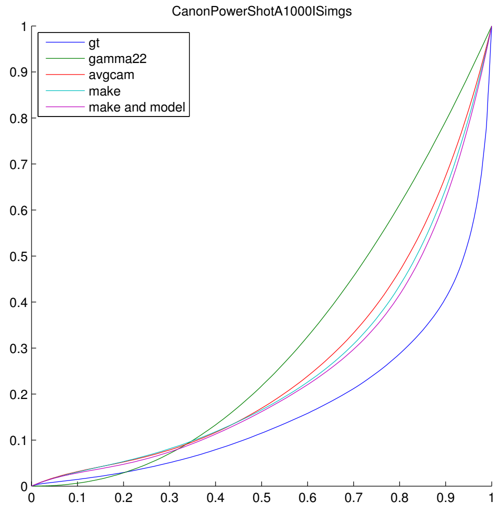


Figure 5.4: The camera response curve for a Canon Power Shot A1000 IS for the green channel. We compare here the curve recovered from a registered set of images with varying exposure (from the Middlebury dataset) with the simple $\gamma = 2.2$ curve and the curves at different levels of our hierachical camera response data structure.

Estimating Illumination. Now that we have estimates for the response function f_i for I_i and the reflectance ρ at each point, we can recover our estimated irradiance at a point p:

$$\frac{f_i^{-1}(I_i(p))}{\rho(p)} = \frac{\rho(p)L_i(p)}{\rho(p)} = L_i(p) \tag{5.2}$$

Descriptor Computation. Our descriptor is computed for an image I_i by taking each point p visible to I_i, binning the irradiance $L_i(p)$ according to the point's normal azimuth θ, and elevation ϕ angles, into a 2D grid of size $b_\theta \times b_\phi$ covering the sphere. If, after binning each point p in this way, a cell in this 2D grid has fewer than t (we used 10) observations, we treat that cell as empty. For each cell that remains, we compute the irradiance in RGB color space by averaging the contribution of all points that fall within that cell. This results in a $b_\theta \times b_\phi$ dimensional descriptor. In our experiments we set $b_\theta = 36$, $b_\phi = 18$, resulting in a 648 dimensional vector). This results in a 7.6KB descriptor for A-IM, and 15.2KB for AV-IM. However, missing data typically means that the descriptor is well over 50% sparse, so their size on disk is typically much smaller. SHM, being highly compressed, can be stored in 189 bytes.

Comparing Descriptors. When comparing two descriptors, we must define a distance function between the descriptor vectors. We found that a simple L_2 distance worked well across our applications, and this is what we use throughout the rest of the chapter, although other distance metrics, such as L_1 or Earth-Movers could also be used. Note that since any given descriptor could have missing data, we only compare elements in descriptors where both have content, and normalize the distance by the number of overlapping elements.

Descriptor Coordinate System. One key question is the coordinate system in which we place our descriptors. One option is that a descriptor is defined with respect to the camera (**camera-frame**), e.g., if light is coming from the left *with respect to the camera*, the left side of the descriptor "lights up." Alternatively, we can orient the descriptor with respect to the world coordinate frame (**world-frame**), since we know the absolute camera orientation from SfM. In this case, the descriptor is fixed in the world so that if light is coming from the south in a particular image, the same parts of the descriptor light up independent of the camera orientation. We also experimented with a variant of each orientation type: an **upright-camera-frame**, in which we first factor out a camera's pitch and roll (but not heading), and an **object-frame**, in which, for scenes with a definite "front" (such as a statue), we orient the descriptor with respect to that object. Note that, up to quantization, each of these frames are related by a 3D

Method	# Photos	# 3D Points
Rio	3515	219,183
Castle	33219	817,899
Heaven	6054	906,647
Humayun	7360	1,308,456
Güell	15586	10,670,473
Peña	8283	2,197,787
Sagrada	5880	6,947,304

Table 5.1: Our landmark datasets taken from Internet photo collections. The number of photos and 3D points extracted are given. The last column is the average uncompressed double-precision descriptor size.

rotation.

5.3 Applications

In this section, we demonstrate the utility of our light descriptors in four applications. We show that we can use our light descriptors to derive higher- level information about illumination, such as the sun position and visibility (Sections 5.3.1-5.3.2). We then show our descriptor's utility in two graphics applications: multi-view object insertion and search-by-illumination (Sections 5.3.3-5.3.4). To demonstrate these applications, we have created seven datasets of landmarks (Rio, Castle, Heaven, Humayun, Güell, Peña, Sagrada) created from large Internet photo collections (for more details on the datasets see Table 5.1).

5.3.1 Sun Position

Analogous to what was done in Section 4.2.3 we demonstrate that our light descriptors can be used to determine the sun position. Our approach and assumptions are also very similar to those of Section 4.2.3, we rely on Exif data to obtain day of year but not time of day, which reduces the search space significantly. One fundamental difference to what was done before is that now we use the physically-based model of sun/sky illumination to predict a *canonical* light descriptor for different times of day, whereas before we used the illumination model to predict the illumination for each point in the 3D reconstruction. The advantages of this approach are two fold. First, the light descriptors are a much more compact representation of the illumination than the generated predictions for all visible points in the reconstruction. Second, the generated *canonical* descriptors can be shared with other scenes that lie at the same latitude, reducing the amount of data that needs to be pre-computed and cached.

To generate predicted descriptors, we use the sun-sky model of Hosek and Wilkie to generate a physically-based clear-sky environment map [36]. For each normal on a sphere, we integrate over the environment map to create an irradiance map, then use these normals and irradiance values to compute a world-space A-IM light descriptor. We generate predicted descriptors for timestamps spaced 10 minutes apart every 5 days over a full year, since the sun direction for a given time varies only slightly from one day to the next. To estimate the sun position for an image, we compute that image's world-frame descriptor, then predict the sun position corresponding to the most closely matching predicted descriptor. Our best results were achieved when searching using a simple Euclidean distance metric with A-IM.

	Sunny (803)		Cloudy (111)		All (814)	
	Mean	Median	Mean	Median	Mean	Median
A-IM	15.70°	9.96°	30.75°	24.75°	17.75°	11.22°
Random	52.83°	50.40°	53.54°	50.72°	52.93°	50.41°

Table 5.2: Quantitative results for our sun direction (measured in degrees) on the ground truth CASTLE evaluation set. We compare to the expected error given by randomly chosen times of day.

We evaluated our method using 814 images with timestamps manually read from the clock on the front of the CASTLE dataset. For each image, we search over all the predicted descriptors corresponding to the date found in the Exif timestamp. Quantitative results are shown in Table 5.2. Predicting sun direction on cloudy days is significantly more challenging, so we report results separately for the 111 cloudy images in the ground truth dataset. On sunny images, our method is able to predict the sun position with a median error of less than 10 degrees, while cloudy images are considerably more difficult. Nonetheless, we still significantly outperform random guessing on cloudy images which lack strongly directional illumination.

5.3.2 Sun Visibility

In addition to sun position, another key aspect of outdoor daytime illumination is weather. In particular, occlusion of the sun by clouds can dramatically change the appearance of a scene, making knowledge of this lighting characteristic important in many applications.

Our approach to determining sun visibility is to label a small number of images and train a classifier on our descriptor to predict whether an image is sunny or cloudy. We could use the full light descriptor itself as a feature, but sunny illumination produces very different descriptors throughout the day due to color and directionality. For example, a sunny morning descriptor might have sunlight reaching the left-facing normals and sky-lit right-facing normals, while the situation is reversed in the afternoon. For this reason, we build a histogram of the irradiance values, discarding the normal information.

One limitation of A-IM is that it averages together the illumination of all points with a given normal. In cases where some points with a given normal fall in a shadow and others are illuminated by the sun, this A-IM will not capture this variation well. For this reason, we use AV-IM, which includes the variance within each normal bin. Our final feature vector is two concatenated 10-bin histograms, one for grayscale average irradiance and one for variance. We use these features in a classifier trained on a set of about 200 manually labeled images for each of our seven datasets. Using 10-fold cross-validation, we found that a k-nearest-neighbor classifier using a χ^2 distance metric performed best, outperforming Support Vector Machines with linear and RBF kernels.

We compared the performance of our features to the features proposed in [52] for webcam images. Their approach divides the saturation and value channels by their averages over a stack of webcam images as a way to isolate the illumination, then computes a joint 2D histogram of these saturation and value ratios. Our descriptor takes a more principled approach, isolating illumination by factoring out albedo. For comparison, we implemented their ratio histogram approach on the image values of the 3D points extracted from each image. The greater variability in photo collections makes our datasets significantly more challenging, but our features outperform theirs, achieving between 72% and 81% cross-validation accuracy as shown in Table 5.3. We also tried a method based on the sun visibility technique proposed in [81] in the context of photometric stereo. They classify sun visibility by thresholding the ratio of second-order to first-order energy of the spherical harmonic coefficients. Even using cross-validation to choose a threshold for each dataset, we found this approach did not perform well.

5.3.3 Object Insertion

We now show that our light descriptor can be used to consistently insert an object into many photos in a collection, with realistic lighting for each image. We estimate a timestamp as described in Section 5.3.1, then render a virtual object to be inserted and a neutral ground plane under a sun/sky environment map generated using [36]. For each image, we render from the camera's viewpoint (known via SfM). We then composite the object into the image using Debevec's differential compositing method [20] which realistically transfers shadows onto the ground in the image. Figure 5.5 shows example results. The four images on the left illustrate that we can easily place an object in a consistent location across many photos from the collection.

Our light descriptor, coupled with the sun/sky model, takes the place of the light probe used by Debevec [20], allowing us to perform object insertion after the fact without specialized equipment at capture time. Whereas [41] requires user input to build a 3D model of the scene and lighting in each image, we use our light descriptor and SfM reconstructions to set the geometry once in 3D space, and automatically generate an environment to match the lighting in any sunny photo in the collection.

Method	Rio	Castle	Heaven	Humayun	Güell	Peña	Sagrada	Average
Ours	**0.67**	0.75	**0.82**	**0.79**	**0.71**	**0.79**	**0.77**	**0.757**
[52]	0.66	**0.83**	0.76	0.72	0.68	0.71	0.78	0.734
SHM Energy	0.63	0.59	0.55	0.70	0.64	0.68	0.71	0.643

Table 5.3: Cross-validation accuracy for sun visibility classification using our method and the method proposed by Lalonde et al. Our method outperforms theirs on all but Castle, and achieves an average accuracy of 75.7%. We also tried the approach from Shen and Tan, which uses the ratio of second to first order spherical harmonic energies. Note that the validation set for Castle is not the same as the set used for sun direction and is not heavily biased towards sunny images.

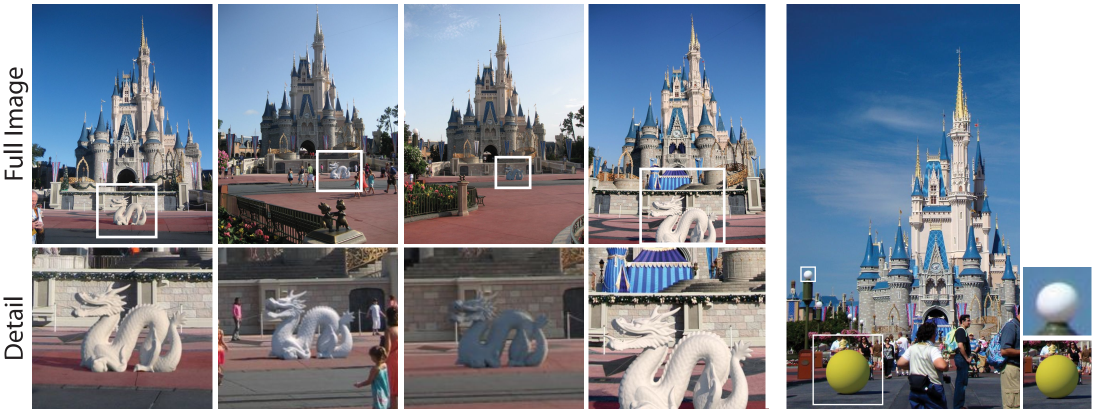


Figure 5.5: Inserting an object into multiple images of a scene with consistent lighting. On the left, we show an object inserted in a consistent 3D location across four images with consistent lighting within each image. On the right one more example, notice that the shading on the synthesized yellow ball matches the real-world white spherical lamp on the left side of the image.

Although this example uses a sun/sky model, other approaches are possible. In principle, a full irradiance map (e.g., derived from a spherical harmonic fit of A-IM) could be used to render Lambertian objects into a scene under arbitrary distant illumination.

5.3.4 Search

Our light descriptors allow us to search for images by *light similarity*. The illumination in a scene is a rich visual element, and we enable searches based on similar illumination within the same scene as well as *across* scenes (since our descriptor is designed to be geometry invariant).

We illustrate several such searches in Figure 5.6. For each example search, we show a query image, and the top matches by light descriptor similarity ei-

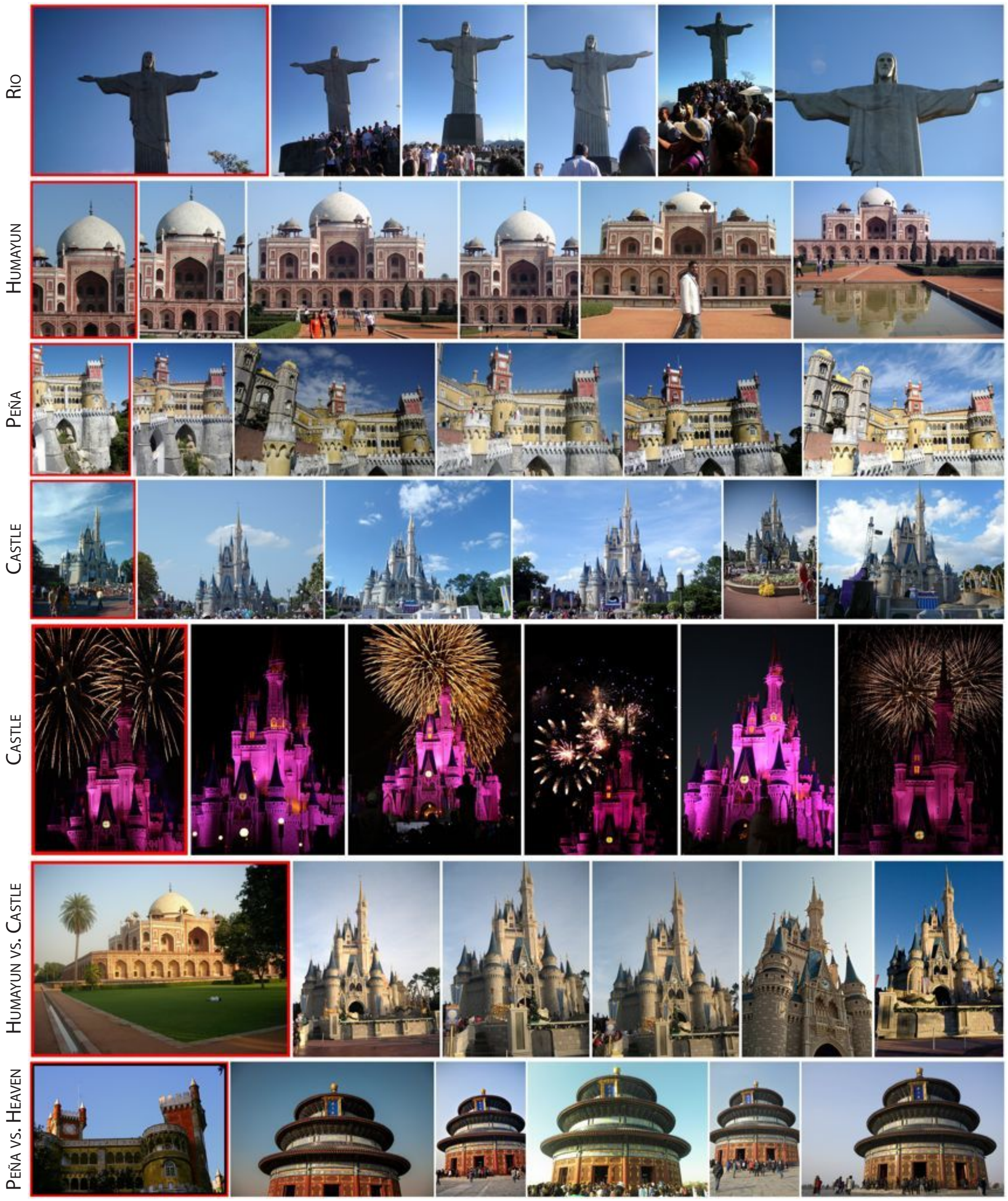


Figure 5.6: Example light similarity search results. Queries were matched against database images which had at least 320 occupied histogram bins in common. Each row contains one query image (leftmost image, marked in red) and the nearest search results following on the right. The first five rows show results for queries within a dataset (Rio, Humayun, Peña, Castle-day and Castle-night) while the remaining rows show results for cross-dataset queries (Humayun ↔ Castle and Peña ↔ Heaven).

ther within the same dataset, or across two different datasets. Note how our method is able to match not only the overall color of the lighting but also its direction. In the last two rows we also see that the method correctly finds images with similar lighting across datasets. In the first cross-dataset match the soft lighting of HUMAYUN is matched in the CASTLE images. In the last row the sunny lighting from the left in PEÑA is matched in HEAVEN (the PEÑA building face is darkened since it faces away from the light). In both cases the sky conditions are matched very closely.

Evaluation with human-judged comparisons. How well does our descriptor work at matching illumination between images? To quantitatively evaluate the effectiveness of our descriptor, we manually labeled a set of image triplets to form a set of human-judged illumination similarity inequalities. Each triplet consisted of a query image C and two other images, A and B, and the task was to select whether A or B has more similar illumination to C (an option was also provided for cases where neither image was clearly more similar). Each triplet was annotated by three people and any triplet not unanimously marked A or B was discarded. The remaining triplets give us a set of inequalities according to a "human" distance D, e.g., $D(A, C) < D(B, C)$. We use these inequalities to judge how well distances in our descriptor space accord with human judgement.

We collected such inequalities for four experiments, three using image triplets all from the same dataset—RIO (106 human-judged inequalities), HEAVEN (46 inequalities), and CASTLE (107 inequalities)—and one where a query from one dataset (RIO) is tested against two images from a second dataset (CASTLE) (113 inequalities). In each experiment, we measure the percentage of triplets for which the descriptor distance agreed with human judgement. We found that an L_2 distance and **object-frame** orientation had an average accuracy of **72.1%**. We compared to the performance of SHM (64.8%) and a simple histogram matching baseline (65.6%). Note that since each triplet has two choices, chance would be 50%. The baseline simply uses a 3D color histogram of the reprojected 3D point values in each image as a descriptor. This can work reasonably on similar views of similar scenes, but fails on different viewpoints and across datasets.

5.4 Discussion and Limitations

Our current implementation requires a large number of photos of the scene in order to recover the albedo at each point. Even though there are many photo collections of famous places, we would like to extend our method to scenes where that is not the case. To do that we could experiment with new methods for estimating an intrinsic image decomposition together with information about the geometry of the scene [40]. We could also use more advanced sensors (RGBD), such as depth cameras, to recover geometry.

Our AV-IM descriptor captures some elements of variation in lighting, but loses some information as well, such as the exact position of shadows. More sophisticated descriptors could capture this information as a function of both normal and geometry. Finally, we only use 3D reconstructed points in computing our lighting descriptors; images often have other strong cues to illumination, such as the sky and ground (which are often both unmodeled in SfM reconstructions). In the future, we plan to combine our method with more holistic image understanding techniques in computing better descriptors—for instance, the sky can reveal information about light coming from behind an object.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In reasoning about light we started this dissertation by introducing a novel method for intrinsic image decomposition which explicitly models Ambient Occlusion, a measure of local visibility at a point that has largely been ignored by the computer vision community. The method operates in image space and makes the weak assumption that over an image stack the lighting varies but its position in each image is generally unknown. Our approach is to use a simple, per-pixel statistic, κ, based on observed intensities over the set of images; from κ, we recover per-pixel ambient occlusion and albedo values by relating our physical model to this measured statistic through our cylindrical crevice model. Despite its simplicity, we show that this statistical approach works well in practice for a range of real-world image stacks. Furthermore, in Chapters 4 and 5 we demonstrate that it can be generalized to work on sparse 3D reconstructions and Internet photo collections thanks to the lack of a smoothness assumption, which allows the method to process points in isolation, without any connectivity information.

In Chapter 4 we refined our algorithm by using a physically based model of outdoor illumination developed by the computer graphics community [36]. This allowed us to model some of the complexities associated with natural illumination. In particular, our refined model incorporates the fact that the sun does not cover the entire hemisphere as the earth rotates. This means that, together with surrounding geometry, a point's normal direction now also influences how much light it receives, with some normal directions simply not being illuminated enough by the sun to reliably recover albedo. Furthermore, changes in the sun intensity and color are also taken into account. In the results section we show that by using this more refined model our algorithm can better estimate the albedo of sunlit scenes, and correctly discards points for which the variation in illumination conditions is not enough to enable accurate estimation of albedo. We also show the utility of these outdoor illumination models for timestamping images.

One drawback of the timestamping method presented in Chapter 4 is that the amount of data generated per image can be quite large. The algorithm makes one illumination estimate per visible point in the 3D model, which for some models and images results in an amount of data that rivals that of the input image. In Chapter 5 we address that issue by devising a novel descriptor, one that summarizes the lighting information into a compact representation that is invariant to the underlying geometry of the scene. In the results section we show that this novel descriptors has uses beyond timestamping, including object insertion, weather estimation, and image search.

6.1 Future Work

The method presented in Chapter 3, although very robust, could benefit from a more expressive model of the local geometry than the cylindrical model we presented, perhaps one that incorporates anisotropy would better match more general visibility scenarios. We also show in the discussion section that the method's estimates are less accurate when the effects of multiple bounces of light become more pronounced, including these effects into the model would also be an interesting direction to explore. Finally, our method operates on a single statistic from the PID, essentially discarding all other information that comes from this distribution. Using other statistics or reasoning about the entire distribution could potentially enable the method to make fewer assumptions about materials (i.e., allowing for more complex material models than Lambertian that incorporate specularities and subsurface scattering).

The sun-sky model used in Chapter 4 is a big step forward in terms of accuracy when compared with the simple lighting model we used in Chapter 3. Nevertheless, there are avenues for improvement there as well. One limitation is that it only models a sunny sky. Incorporating cloud coverage and weather could potentially improve the methods accuracy.

The descriptors we presented in Chapter 5 are an initial step in making lighting descriptors a first-class citizen in graphics and vision. The representation we derived is compact and simple to compute, and as shown in the applications section is flexible enough to enable multiple different applications. One limitation though is that it can only represent distant lighting, a limitation partially addressed by the AV-IM version of our descriptor. More investigation into different representation could potentially address this limitation. In order to obtain the estimate of the illumination in a scene we made use of large photo collections and the method developed in Chapter 3, other methods for intrinsic image decomposition exist that operate on a single image and recover a rough estimate of shape [10]. It would be interesting to use our descriptor with these methods so that more scenes could be used.

Part II

Symmetry

CHAPTER 7

IMAGE MATCHING USING LOCAL SYMMETRY

Symmetry, at many different scales, and in many different forms, is a powerful feature in the structure of our world, evident in the shape and appearance of many natural and man-made scenes. Humans have an innate ability to perceive symmetries in objects and images, and tend to construct objects that exhibit a range of local and global symmetries. For computer vision applications, analysis of symmetry is attractive for a number of reasons: symmetries are potentially a stable and robust feature of an object, yet, when considered at all scales and locations, are also potentially quite descriptive.

For instance, consider the pairs of images shown in Figure 7.1. Even though each pair is perfectly registered, the pairs exhibit large changes in appearance due to varying illumination, age, or style of depiction. These factors can lead to large differences in low-level cues such as intensities or edges. However, each of the structures depicted can be described in terms of a nested hierarchy of local symmetries (and repeated elements). These symmetries are—to some degree—preserved in these pairs.

In this chapter, we seek to exploit such local symmetries for robust image matching through local features based on such symmetries. While other local features, such as SIFT [60], are highly invariant to a range of geometric and photometric transformations, we find that they often perform poorly given the kinds of dramatic variations shown in Figure 7.1. Our hypothesis is that taking advantage of local symmetries can help with matching in these kind of difficult cases. In the case of SIFT, for instance, the local gradient information around a point may be very different between two depictions, whereas the symmetries

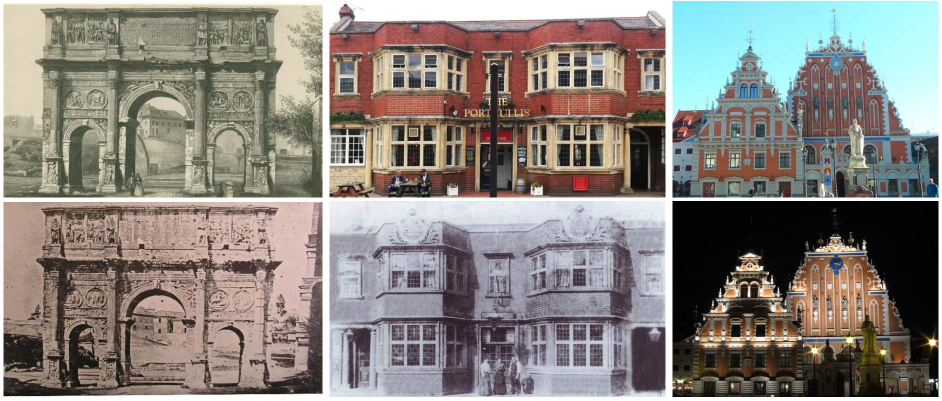


Figure 7.1: *Difficult image pairs.* Each pair of images in this figure shows a registered view of a building. Despite the geometric consistency, these images are difficult for feature matching algorithms because of dramatic changes in appearance, due to different depictions (drawing vs. photo), different time periods (modern vs. historical), and different illumination (day vs. night). While these images are dissimilar at a pixel level, each image in a pair exhibit similar symmetries, which we seek to exploit for matching. These images are selected from our dataset.

may be more resilient (if we can measure them). Hence, we propose both a feature detector and a feature descriptor built from a simple function that detects local symmetries, across an image and in scale space. Our symmetry features are local, and retain some of the advantages of local features, but in some sense they aim for a more "mid-level" scene description than current local features. Our features are primarily designed for architectural scenes where symmetries are common.

Our proposed features leverage a simple measure of local symmetry based on analyzing image differences across symmetry axes, computed densely in regions across the image; we score each patch in the image, and patches at different scales, based on three types of symmetries (horizontal, vertical, and rotational). We develop a way to detect scales for such detected local symmetries, and accordingly define a feature detector that returns a set of maximally locally symmetric positions and scales. We also develop a feature descriptor based on these same symmetry measures.

We evaluate our method on a challenging new dataset created from registered pairs of images with difficult appearance changes (including the images in Figure 7.1). Some of these pairs are rephotographs of the same scene many years apart, while others represent differences in illumination or style. These types of images are of interest in applications such as large-scale 3D reconstruction from heterogeneous image sources [85], especially settings that incorporate historical imagery [76]. The fact that the images are registered allows to focus on testing local features based on appearance changes alone. We evaluate two versions of our method, one based on raw intensities, and another based on gradient histograms.

7.1 Related Work

There has been a great deal of work on symmetry detection in computer vision and graphics (Liu et al. [58] present an excellent survey). However, there has been relatively little work on using local symmetries as an explicit feature for image matching. The closest related work to ours is probably the self-similarity descriptor of Shechtman and Irani [79]. That work proposes to use patterns of self-similarity of an image patch in a local neighborhood as a robust feature descriptor for matching across images and videos, and demonstrate good results on difficult matching problems (e.g., matching a rough silhouette of an object to an image). In our case, we use different forms of *symmetry* as cues, rather than repetitions, and use these to define both a feature detector (based on local symmetry patterns) and a descriptor. Other work utilizing repeated patterns includes that of Schindler et al. [77], who detect and match highly repetitive patterns on building facades, and that of Wu et al. [93], who detect large repeating elements of facade images, and use these as robust support regions for computing features. In contrast, we do not try to build a mid-level representation of the image that explicitly reasons about symmetric structures, but instead compute many local features based on a softer notion of a symmetry score.

Loy and Zelinsky also propose a feature detector [61] based on radial symmetries, using a fast radial symmetry transform that accumulates votes for symmetric regions from gradient information. This was demonstrated to work well for finding radially symmetric features such as eyes, but was not demonstrated for matching applications. In an earlier approach, Reisfeld et al. [74] used a similar voting scheme to detect interest points using radial symmetries via a generalized symmetry transform.

Our local symmetry detector is also related to other low-level measures used for symmetry detection in vision. Kovesi observed that local bilateral symmetry in image intensity relates to the response of filters of different phase [45]. Kovesi later extended this concept of "phase-congruency" for detecting features such as edges and corners [46]. Di Gesù et al. proposed the discrete symmetry transform [22], based on axial moments and related to the medial axis transform. Other work is based not on image intensities or gradients, but instead on an initial set of sparse features (e.g., SIFT) finding relationships between these features consistent with local bilateral or radial symmetry [62]. We define a simple symmetry score based on a general measure of image similarity across reflection axes, and compute this densely over the image and across scale space.

Our symmetry score is related to the reflective symmetry transform proposed in the graphics community for 3D shape analysis [69], but we compute these scores locally (and over scales), rather than globally.

Our detector has some similarities with the recently proposed edge foci interest regions of Zitnick and Ramnath [98]. Their edge foci detector robustly fires at blob-like regions (as with the difference-of-Gaussians detector); our detector also tends to find these kinds of regions, but also on broader class of locally symmetries.

Recent work by Shrivastava et al. [83] also addresses matching of difficult images across different domains (e.g., paintings and photographs), but using global features (in their case, a global HOG descriptor) and using linear classification techniques to weight the HOG descriptor. In contrast, we are interested in local-feature level matching so as to derive feature correspondence, and our focus is on features (symmetry), rather than learning.

7.2 Local Symmetry

We now describe how we "score" local symmetries in an image using a simple analysis of image similarity across different types of reflection axes. The goal is to run this score function densely across an image, and at different scales, to characterize all of the local symmetries present; later, we use this score to build features. We start by treating an image as a 2D function f that maps pixel locations (x, y) to intensities. Our work addresses two common types of symmetry: bilateral (across a specified line in the image) and $2n$-fold rotational symmetry (symmetry defined by reflection across an image point). If a 2D function

$f : \mathbb{R}^2 \to \mathbb{R}$ exhibits bilateral symmetry around the origin, then:

$$f(r, \theta_s + \theta) = f(r, \theta_s - \theta)$$

where r and θ are polar coordinates of points on the 2D plane, and θ_s is the angle of the plane of symmetry. Similarly, if a f exhibits $2n$-fold rotational symmetry, then:

$$f(r, \theta) = f(-r, \theta)$$

Note that this is the definition of 2-fold symmetry, where rotating the plane by $360°/2 = 180°$ results in an identical figure (r rotated 180° about the origin is equal to $-r$), and any $2n$-fold symmetric function is also a 2-fold symmetric figure.

Detecting the two kinds of symmetry above can be understood more intuitively if we focus on slices of the plane $\mathbb{R}^2$ (see Figure 7.2). For each type of symmetry, certain 1D slices will be symmetric: in the case of bilateral symmetry, these slices will be perpendicular to the axis symmetry, and for $2n$-fold rotational symmetry they all pass through a single point. By focusing on these slices, both kinds of symmetry are similar, and the property that holds is $g_{1D}(t) = g_{1D}(-t)$, where $g_{1D}(t)$ is a 1D slice of f parametrized by variable t on the line of symmetry. Given a function g, one could check if it is symmetric by comparing each value $g(t)$ to $g(-t)$, either in a window (for local symmetry), or everywhere (for global symmetry).

7.2.1 Scoring Local Symmetries

We now focus on how to use this intuition to compute a local symmetry score. We will develop this score by looking for symmetries using the raw image in-

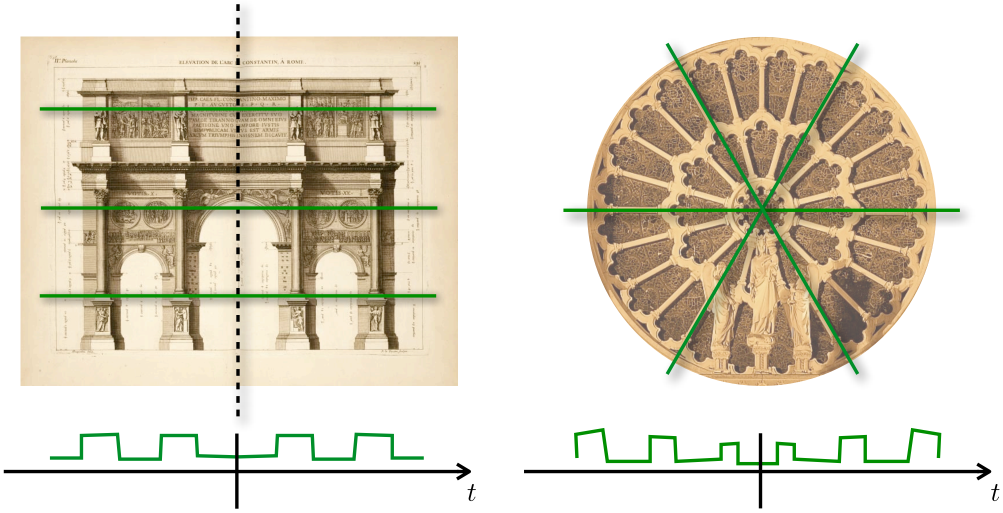


Figure 7.2: By taking a slice of a 2D image f, we see that the problem of detecting symmetry can be posed of that of determining if a set of 1D slices through f are symmetric. Left: an image with (approximate) bilateral symmetry about the vertical axis. All horizontal slices (green) are close to even functions (as shown in the function profile at bottom). Right: (approximate) $2n$-fold rotational symmetry. Most slices through the center of the image are even functions.

tensities themselves, but the extension to other types of per-pixel features is straightforward; later, we evaluate a score function based on intensities, as well as one based on gradient histograms computed densely over the image. Ideally, this score would be efficient to compute and discriminative, yet robust to small asymmetries in a local region due to noise, occlusion, or illumination changes.

Our local symmetry score is defined for each location $\mathbf{p} = (x, y)$ in an image, and characterizes the degree of symmetry at that location using the intuition above. To define the score, we require three components:

Symmetry type. As described above, we consider either bilateral or $2n$-fold rotational symmetries. This type defines a function $g_{s,\mathbf{p}}(\mathbf{p}')$ that maps any other image point $\mathbf{p}' = (x', y')$ into its symmetrically corresponding point with respect to the point $\mathbf{p}$ and symmetry type s. In other words, if the image exhibits symmetry type s at location $\mathbf{p}$, then $f(\mathbf{p}') = f(g_{s,\mathbf{p}}(\mathbf{p}'))$.

Distance function. Next, we need a distance function $d(\cdot, \cdot)$ that measures how well a given pair of corresponding symmetric pair of points $(\mathbf{p}', g_{s,\mathbf{p}}(\mathbf{p}'))$ match each other in appearance. For a symmetry score based on intensities, d could be defined as the absolute difference in the intensities between these two points: $d(\mathbf{p}', \mathbf{q}') = |f(\mathbf{p}') - f(\mathbf{q}')|$.

Weight mask. Finally, we define a function $w_\sigma(r)$ that weights how important each set of corresponding point pairs around the point of interest $\mathbf{p}$ is to determining the symmetry score at $\mathbf{p}$. If we were only interested in perfect global symmetries, the weight mask would have infinite support. To detect local symmetries, one might use a Gaussian mask, giving more importance to pairs close to $\mathbf{p}$. For simplicity, we assume that the weight mask is radially symmetric, and is thus a function just of the distance r from the center point $\mathbf{p}$. The subscript σ denotes a scale for the weight mask, which modulates the size of support region of $w_\sigma(\cdot)$, allowing us to create a scale space for symmetries (as described in Section 7.3). In the case of a Gaussian mask, σ is the standard deviation.

Putting the three components together, we arrive at a function that we call the *local symmetry distance SD*:

$$SD(\mathbf{p}) = \sum_{\mathbf{p}'} w(||\mathbf{p}' - \mathbf{p}||) d(\mathbf{p}', g_{s,\mathbf{p}}(\mathbf{p}')) \tag{7.1}$$

As described above, the simplest symmetry distance function we consider is one built on raw image intensities, where d is the absolute difference in intensity, and w_σ is a Gaussian function. This simply measures, at a given pixel location $\mathbf{p}$, how similar the image is to itself when flipped across a symmetry axis (or point) through **p**, accumulating the differences across a Gaussian support region. We refer to this family of symmetry distances as SYM-I (for "intensity"). In practice, we will consider two versions: bilateral (denoted SYM-IB for generic angles and the shorthands SYM-IH and SYM-IV for horizontal and vertical versions) and $2n$-fold rotational symmetry (SYM-IR).

One potential problem with SD in Eq. (7.1) is that uniform image regions are trivially symmetric, and thus have low symmetry distances. Nevertheless, a characteristic of these regions that differentiates them from less trivially symmetric regions is that their symmetry is not well localized, i.e., SD is low in a wide region. To address this, we find "edges" or "blobs" in the raw SD map, through convolution of SD with an appropriate filter L based on a Laplacian-of-Gaussian (LoG) kernel [57]. For the distance based on rotational symmetry we use the standard LoG kernel, while for bilateral symmetries we use a kernel that has a LoG profile perpendicular to the axis of symmetry and a Gaussian along the axis of symmetry. With the correct sign on the kernel L, this convolution converts the symmetry distance function into a symmetry score function:

$$SS(\mathbf{x}_c) = L * SD(\mathbf{x}_c)$$

where $*$ denotes convolution.

7.2.2 Gradient Histogram-Based Score

The symmetry score above uses raw intensities; we hypothesize that it may be more robust to look for symmetries based on gradient orientations, as gradient orientations tend to be more stable to photometric changes [60, 98], and also may be more discriminative than raw intensities (e.g., oriented edges may accidentally coincide less frequently). Specifically, we define at each image pixel a histogram of local gradient orientations $\mathbf{h}$, then use this vector-valued function in place of the scalar function f in the local symmetry transforms described above (using the dot product of two vectors, rather than the absolute difference in intensities, to compare the appearance of two symmetric points). This vector-valued function is related to SIFT or HOG features [19], but computed densely at each pixel of a given scale. We call this variant of our symmetry score SYM-G (for "gradient").

To compute the per-pixel gradient orientation histogram function $\mathbf{h}(\mathbf{p})$ at pixel $\mathbf{p}$, we first compute the gradient magnitude and orientation at $\mathbf{p}$ using finite differences, slightly blurring the image to remove high-frequency noise (we apply a Gaussian with $\sigma = 0.5$). While we compute the gradient orientation as usual, we apply the local constrast enhancement of Zitnick [97] to the gradient magnitudes, which helps normalize edge magnitudes between high- and low-constrast regions. At each pixel $\mathbf{p}$, we bin the gradient orientations in a small region around $\mathbf{p}$ into an orientation histogram, weighted by gradient magnitude, and Gaussian weighted by distance to $\mathbf{p}$. We use a small Gaussian with $\sigma = 0.5$, and an orientation histogram with eight bins, softly binning each local edge orientation, and treating orientations as "unsigned" (e.g., orientation are only defined up to a 180 degree rotation). This results in a local histogram $\hat{\mathbf{h}}(x, y)$, which we (softly) normalize to form the final vector-valued function:

$$\mathbf{h}(x, y) = \frac{\hat{\mathbf{h}}(x, y)}{||\hat{\mathbf{h}}(x, y)|| + \epsilon}$$

where $\epsilon = 0.05$ is added to the norm for robustness to noise.

To compare the histograms $\mathbf{h}$ at two reflected pixel positions $\mathbf{p}'$ and $g_{s,\mathbf{p}}(\mathbf{p}')$, we compute their dot product after "flipping" one of the histograms by permuting the bins as necessary to form a histogram of reflected orientations. This dot product is large if the two histograms are similar, and so represents a similarity, rather than a distance; hence, this directly gives us a symmetry score function (rather than a distance). This symmetry score is related technique of Loy and Eklundh [62] (who match SIFT features within an image), but ours is computed densely across image (and scale) space. As with SYM-I, we define two versions of this score, SYM-GB (or the shorthands SYM-GH and SYM-GV), and SYM-GR, for bilateral and rotational symmetry types, respectively.

7.3 Scale Space

The previous section defines a local symmetry score function on an image $f(x, y)$. We wish to use this score to detect features; in order to create a good detector, we need to be able to reliably compute interest points in scale space, as well as image space. Our approach will be to simply compute the symmetry score, unmodified, on a Gaussian image pyramid densely sampled in scale space (while Section 7.2 describes scale space in terms of the scale of the weight mask, we implement it as a function applied uniformly across such a pyramid).

Possible candidate function for use in feature detection are the SYM-IR or SYM-GR score functions (the intensity- or gradient-based rotational symmetry score), as these tend to have well-localized large values in the centers of symmetric regions. However, for SYM-IR, we found that using a Gaussian function as w_σ gives good localization in x and y, but poor localization in scale. This is because the Gaussian has too much mass close to the origin, and the support region of the weight mask changes slowly across scales; thus, the Gaussian responds slowly to the inclusion of asymmetric or uniform image regions under its support, and it becomes hard to precisely localize features in scale space. Thus, for the purpose of feature detection we choose a different function for w_σ, a mask resembling a smooth ring, defined as $w_\sigma(r) = Ae^{-\frac{(r-r_0)^2}{2\phi^2}}$, where r is the distance to the origin, A is a normalization constant, r_0 is the ring radius, and ϕ controls the width of the ring (an example ring mask is shown in Figure 7.3).

To better understand the advantages of using this weight mask for scale space localization, we compare it to the Gaussian mask in Figure 7.3. For a set of three simple images, we show how the rotationally symmetric version of the SYM-I score (shown as a fraction of the max possible LoG response) varies at the center of the image as the scale changes. The middle row shows the results of for the ring-shaped weight mask, while the bottom row shows the results for the Gaussian weighting mask. The dashed red line shows the radius of the "interesting" symmetric portion in the center. We can see from the plots that the ring weighting gives a much more pronounced peak around the dashed line than the Gaussian, sharp enough to fire for each individual ring in the third image consisting of concentric circles.

For SYM-G (the gradient-based score), we found that the Gaussian weights worked well, possibly because the overlap between symmetric pairs of gradient histograms generally results in a much sparser signal in the support region of the weight mask. SYM-G scores at two pyramid levels for the clock image in Figure 7.5 are shown in Figure 7.4.

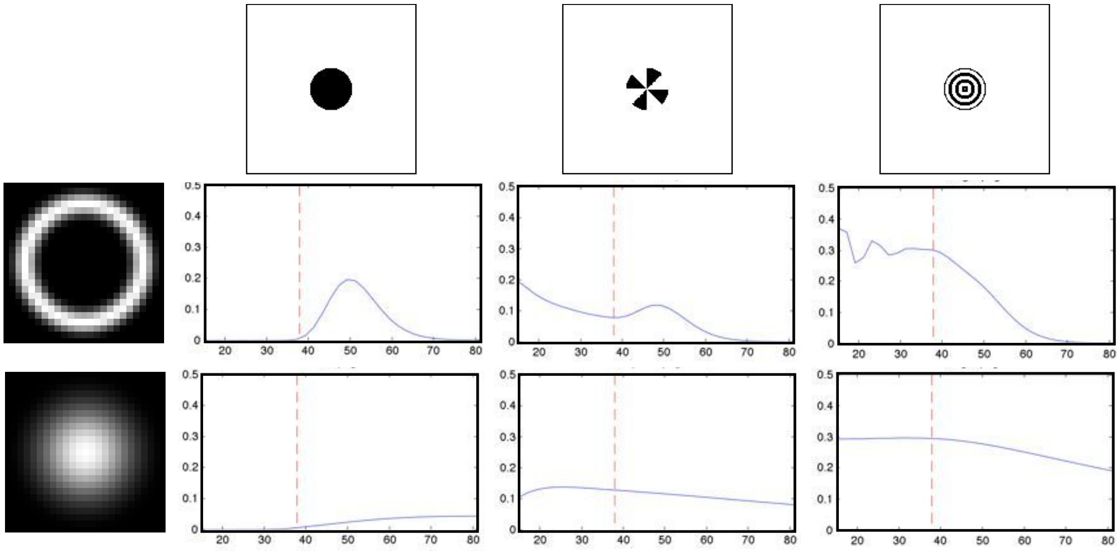


Figure 7.3: Response of the Laplacian filter as a function of scale for the central pixel of three synthetic images (top row), using SYM-IR. Two weight masks w_σ are shown: ring (middle row) and Gaussian (bottom row). The horizontal axis corresponds to kernel size and the vertical axis to fraction of maximum response of the LoG filter. The ring mask gives a much more localized response.

7.4 Local Symmetry Features

We now describe how we use our symmetry scores to define local features; we consider both feature detection, by finding local maxima of the score, and feature description, by building a feature vector from local symmetry scores.

7.4.1 Feature Detector

As mentioned above, the maxima of the SYM-IR (with ring weighting) and SYM-GR functions are good candidates for feature detection, due to their good localization. In contrast, the bilateral symmetries (the SYM-H and SYM-V score

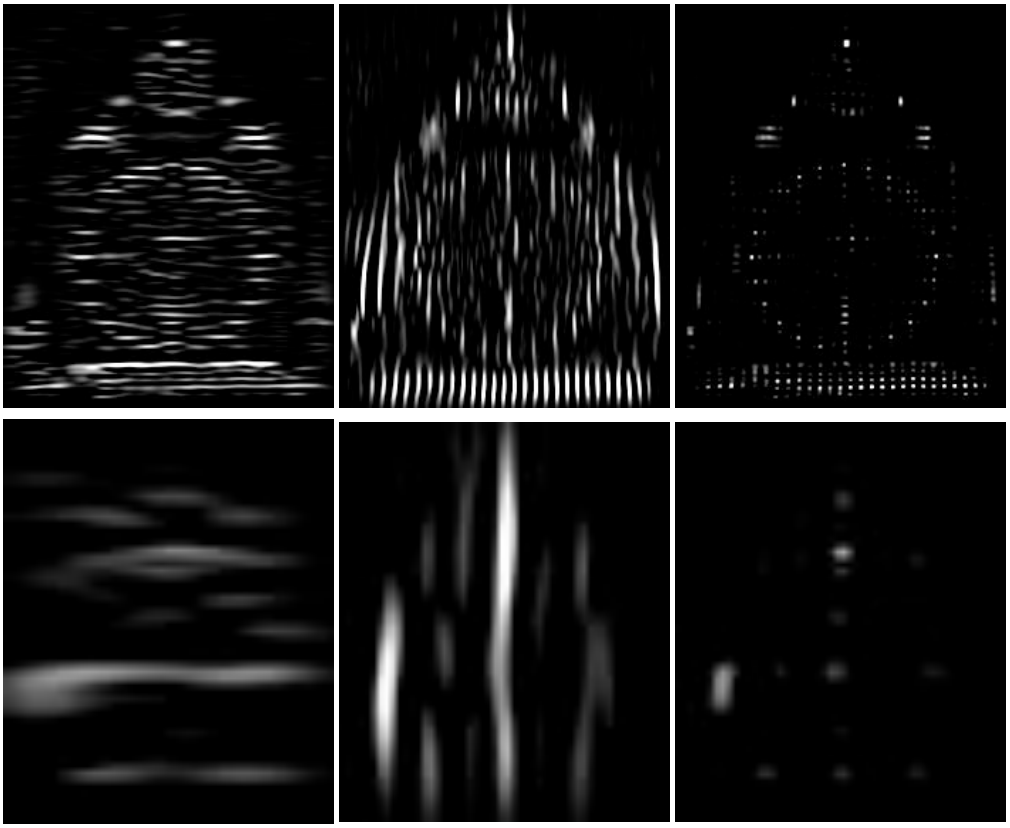


Figure 7.4: Two distinct levels of the scale space pyramid for SYM-G on the image in Figure 7.5. The top row shows the symmetry score computed at a fine scale, the bottom row a coarse scale. Columns correspond to horizontal, vertical, and the product of the two symmetries.

functions) tend to not be well localized, as their responses tend to look edge-like—an entire line of pixels along some axis of symmetry will tend to be strong all at once—rather than having large values at isolated points. However, we found that another stable detector can be constructed by finding maxima of the *product* of SYM-H and SYM-V, as this will be large only at locations that exhibit both horizontal and vertical symmetries (e.g., the centers of windows or other such architectural features). In our evaluation, we consider using the SYM-IR function as a detector, as well as a SYM-GH × SYM-GV detector built from the product of SYM-GH and SYM-GV.

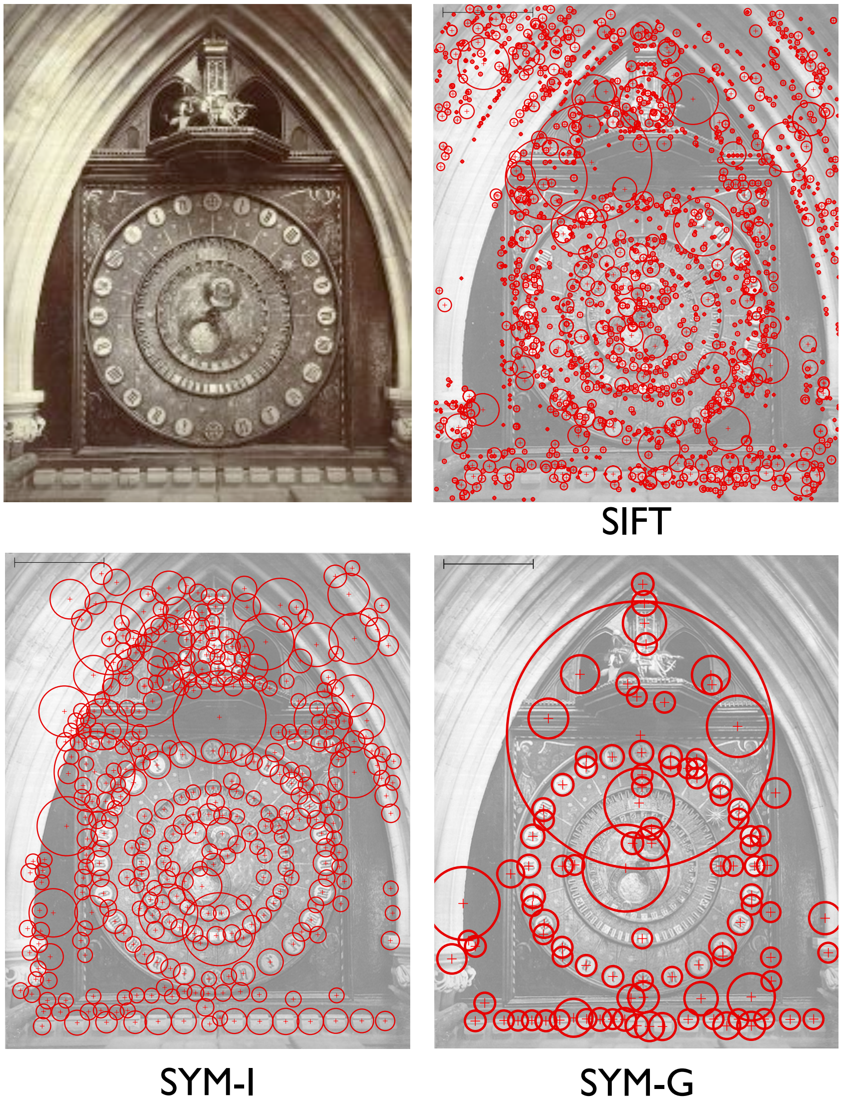


Figure 7.5: Detected features for SIFT, Sym-I and Sym-G for an example image. Each circle shows a feature with scale. To ease vizualization the non-maxima overlap threshold was set to a stricter 0.1 for Sym-I. Note how the symmetry-based detectors more reliably fire on features such as the ring of circles around the clock's circumference. Our symmetry features also tend to fire in larger regions on average than SIFT.

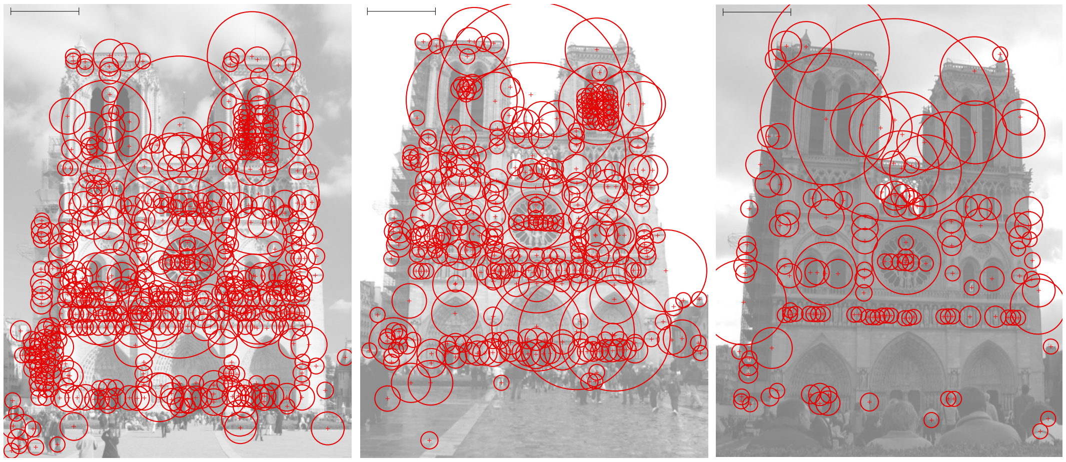


Figure 7.6: Detected features for SYM-G for images of the Notre Dame Cathedral. Note how some symmetric features are repeatably detected across each image, including the central rose window.

Non-maxima suppression. To detect features given a score function computed on a Gaussian pyramid, we first find maxima in each scale independently (thresholding small score values), resulting in a set of initial detections at locations (x, y, s), where x and y represent positions in the full resolution image, and s is the detected scale of the feature. We represent the support of each feature as a circle of radius s centered at (x, y) in the original image. We then select a subset of features that are locally strong across scales. In particular, for each detected feature F, we keep F if and only if it has the highest symmetry score of any feature whose support overlaps that of F by more than a threshold α (in our work, we use $\alpha = 0.4$ for SYM-G and $\alpha = 0.2$ for SYM-I; SYM-I tends to be noisier, demanding more stable detections). This results in a final set of partially overlapping features; we note that symmetric regions often overlap, due to repeated patterns and hierarchically nested symmetric regions. This approach to non-maxima suppression is similar to that commonly used in object detection [25]. Example detections using SYM-I and SYM-G are shown in Figure 7.5, and example detections for a set of images of Notre Dame are shown in Figure 7.6.

7.4.2 Feature Descriptor

We also devise two feature descriptors based on our local symmetry score, which we refer to as LPG (Log-Polar Grid of Local Symmetries) and HOLS (Histogram of Oriented Local Symmetries).

LPG. Encodes the distributions of the three SYM-I scores (SYM-IH, SYM-IV, and SYM-IR) around a feature location and at the detected scale; in essence, this describes patterns of local symmetry around the detected feature point (see a visualization of the descriptor in Figure 7.7). We use the Gaussian weighting for SYM-I here, as we are not concerned with detection. To compute the descriptor for a keypoint (x, y, s), we use a strategy similar to that of the shape context descriptor [14]: we impose a log-polar grid on the image, centered at (x, y), at scale s of the Gaussian pyramid, with diameter four times the scale s. For each symmetry type and each cell we store the maximum value of the corresponding symmetry score within that cell; the max gives some robustness to small deformations. We concatenate values for the cells for the three different symmetry types and normalize the resulting vector to have unit norm. For our experiments we use a log polar grid with 20 angular cells and 4 radial ones for each score, resulting in a 240-dimensional descriptor.

HOLS. Encodes the orientations of bilateral symmetries as localized histograms, very similar to SIFT [60]. For each possible scale s we compute the bilateral symetry score SYM-B-θ_i for a small number of orientations $i \in \{1, \ldots, n_\theta\}$

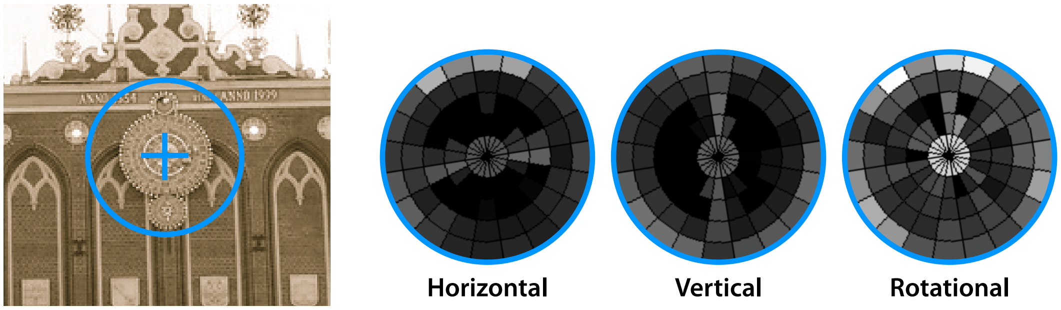


Figure 7.7: Visualization of the LPG descriptor. Left: input image with marked feature and descriptor support. Right: the three symmetry score maps.

(in the experiments we used $n_\theta = 8$). Next we blur each Sym-B-θ_i using a Gaussian kernel with standard deviation σ_b, which gives the descriptor certain robustness to deformations. The parameter σ_b is chosen so that neighboring samples (discussed next) do not contain the same information. Now that we have a set of blurred Sym-B-θ_i's the next step is to extract a descriptor at location (x, y). To do this we sample the Sym-B-θ_i's at a set of discrete locations on a square grid (for the experiments we used a 4×4 grid), where each location corresponds to a different histogram. We further multiply each histogram by a Gaussian weight using the distance of the sample to the feature coordinate (x, y). The width of the Gaussian is chosen so that twice its standard deviation covers the sampling grid's support; the purpose of this weighting is to give higher importance to the histograms closer to the center. As a last step there are two normalizations. First, we concatenate all histograms for a given feature and normalize the descriptor to have unit norm, then we threshold any value below a threshold τ ($\tau = 0.2$ for the experiments), and finally we renormalize the descriptor to unit length. For a 4×4 spatial grid and 8 orientations we end up with a $8 \times 16 = 128$ dimensional descriptor, the same dimensionality as SIFT. We report results using both Sym-I and Sym-G as our local symmetry measures.

7.5 Experimental Results

We conducted a set of experiments to test the performance of our detector and descriptor; we first evaluate the detector, comparing its repeatability against the common DoG detector, then show how different combinations of detector and descriptor perform on a feature matching test.

For evaluation we collected a set of pairs of images, composed mostly of architectural scenes, that we believe challenge modern local feature matching methods. The image pairs exhibit an array of dramatic appearance changes, due to illumination, age, and rendering style (paintings, drawings, etc.). For this set, we wanted to factor out geometry, and focus on appearance; to that end, we pre-aligned most images using a homography. A few examples of the image pairs in the dataset are shown in Figure 7.1.

In total there are 48 image pairs, 12 of which come from the benchmark produced by [98] (the “Notre Dame” and “Painted Ladies” sets). Two pairs come from the dataset of Mikolajczyk et al. [63], one from “Graffiti” and one from “Cars.” The first exhibits large change in viewpoint and the second in illumination.

7.5.1 Evaluating Detections

We evaluate the repeatability of each feature detector given two images I_1 and I_2, the detected sets of keypoints K_1 and K_2, and a homography H_{12} that maps points in I_1 to points in I_2 (most of our homographies are the identity, due to pre-alignment). A common repeatibility metric is to compute the fraction of keys in K_1 that have a similar detection in K_2, when K_1 is warped by H_{12}. However, using this measure directly gives bias towards detectors that produce large numbers of keypoints, so instead, we take the top n keypoints from each set according to some ordering (size or detection score) and compute the fraction m_n of similar detections given these subsets; this is the *repeatability score* (higher is better, 1.0 is best). By varying n we can generate a curve that shows repeatability for all sets of top n features. By ordering keypoints according to the detector response, we measure how invariant the detector is to the changes observed in the pair of images. By ordering according to scale in decreasing order we measure the repeatability of larger features; this is an interesting measure, as we observe for these difficult images that larger features are often more stable.

To determine if two keypoints $k_1 \in K_1$ and $k_2 \in K_2$ are "similar enough" detections, we use the H_{12} to map k_1 into I_2 (both its location and support region), then measure the relative overlap of the support regions [63]. If the relative overlap is larger than a given threshold (we used 0.6) we declare a match. Since there are small errors in our alignments (on the order of a few pixels) we boost the support region of smaller keypoints to be at least 10 pixels in diameter, so as not to penalize small features because of misalignment. We also calibrated feature sizes by comparing the reported scale for a simple image consisting of a black disk on a white background, making sure the detectors report compatible scales. For this experiment, we compare the DoG detector (as implemented in SIFT) with our SYM-I and SYM-G detectors. Example repeatability curves for top subsets of features ordered by score and scale are shown in Figure 7.5.1. Table 7.5.1 shows the average repeatability score on all images of the dataset for $n = 100$ and $n = 200$, for both score and scale. On average, repeatability scores are more reliable for symmetry features as shown by the repeatability

	Scale		Score	
	100	200	100	200
SIFT	0.124	0.105	0.035	0.053
Sym-I	0.121	0.113	0.098	0.103
Sym-G	0.162	0.181	0.175	0.207

Table 7.1: Average repeatability score for the top $n = 100, 200$ detections according to scale and score. In cases where the detector produced less than n keypoints we report the repeatability score for the full feature set.

score for the score ordering. This suggests that our symmetry scores, both intensity based (Sym-I) and gradient based (Sym-G) are better preserved under the types of variation exhibited in the dataset. For Sym-I we observe that the largest features are not always the most reliable as its average repeatability becomes lower than SIFT. Sym-G does better overall, though we have observed that for image pairs with few symmetries (such as the standard Graffiti pairs), SIFT outperforms the symmetry detectors.

7.5.2 Evaluating Descriptors

We now evaluate our local symmetry descriptor. For each pair of images we extract keypoints and descriptors and match descriptors using the standard ratio test [60] on the top two nearest neighbor distances. By varying the threshold on the ratio score, and comparing the matched set of keypoints to ground truth (known in advance from the homography) we can obtain a precision-recall (PR) curve that summarizes the quality of the match scores.

We measure the impact of the descriptor and the detector separately as follows. First, we generate two sets of perfectly matched synthetic detections by first creating a set of keypoints K_1 on a grid in I_1 (in our experiments the spac-

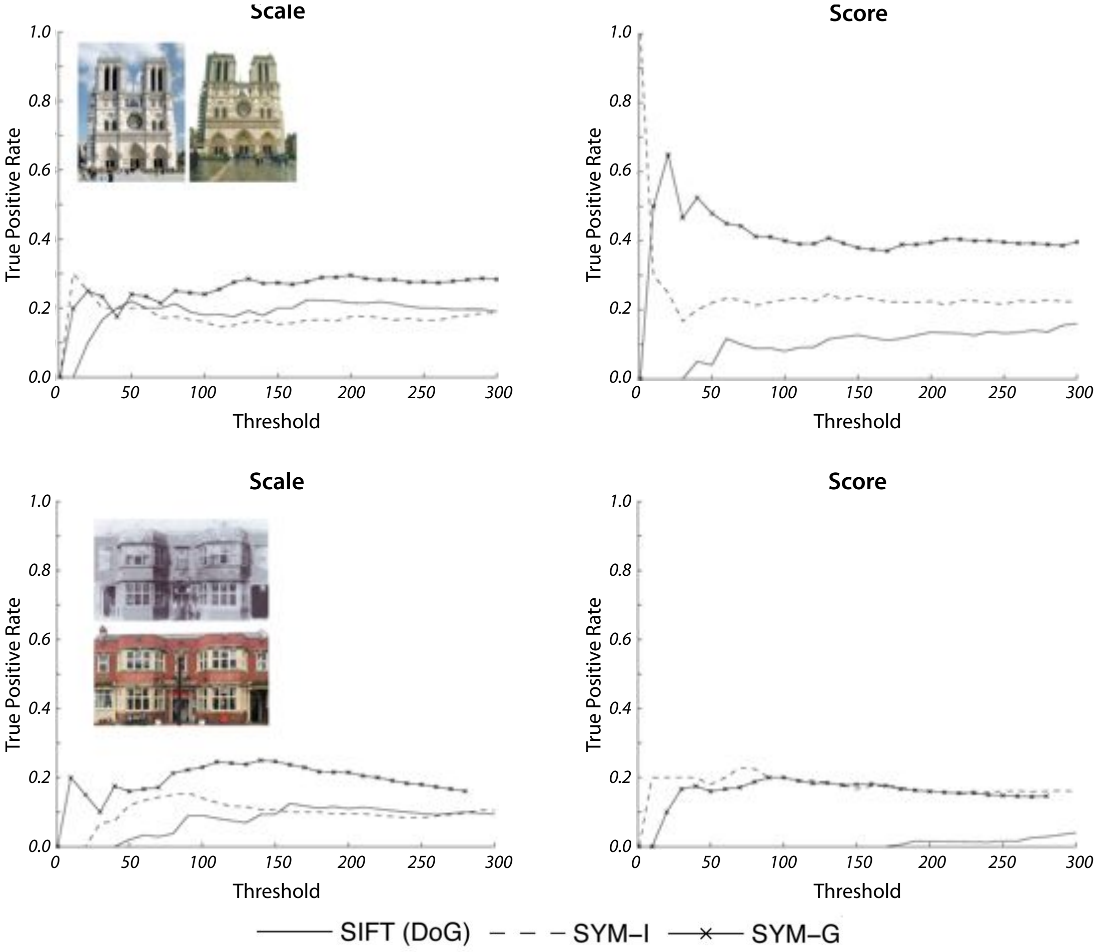


Figure 7.8: Example results for the detector repeatability experiment for two pairs of images, for each of three detectors, SIFT, SYM-I, and SYM-G. The first plot shows the repeatability score as a function of subsets of features with largest scale, and the second plot shows the repeatability score according to subsets of features with highest detector score. In these plots, higher (closer to 1) represents better repeatability.

ing between points is 25 pixels and the scale of each keypoint is set to 6.25). We then map these keypoints to I_2 using H_{12}, creating a matched set of keys K_2. We discard keypoints whose support regions are not fully within the image. In addition, we extract SIFT, SYM-I, and SYM-G detections from each image, and describe all four types of detections (GRID, SIFT, SYM-I, SYM-G) with three combinations of feature descriptors: SIFT, our symmetry descriptors (LPG-I, HOLS-I, HOLS-G), and combinations formed by concatenating the SIFT descriptor and one of our symmetry based descriptors (after normalizing each to have unit norm). This combined descriptor gives a measure of the complementarity of the sources of information provided by SIFT (a gradient-based descriptor) and one of our local symmetry-based descriptor.

The PR curves for a few image pairs are shown in Figure 7.9, and Table 7.2 reports the mean average precision for each combination of detector and descriptor over all pairs from our dataset. For the simple descriptors (as opposed to the combinations) HOLS-G is the top performer, followed closely by HOLS-I. For the combined descriptors combinations of SIFT and an intensity based symmetry based descriptor give better results. It is interesting to see in Figure 7.9, GRID detector, middle row how symmetry based descriptors outperform both SIFT and combinations of SIFT and symmetry based descriptors. Furthermore, in many cases the curves for LPG-I and HOLS-I are virtually identical, suggesting that the encoding of the descriptor is not as important as the underlying symmetry measure.

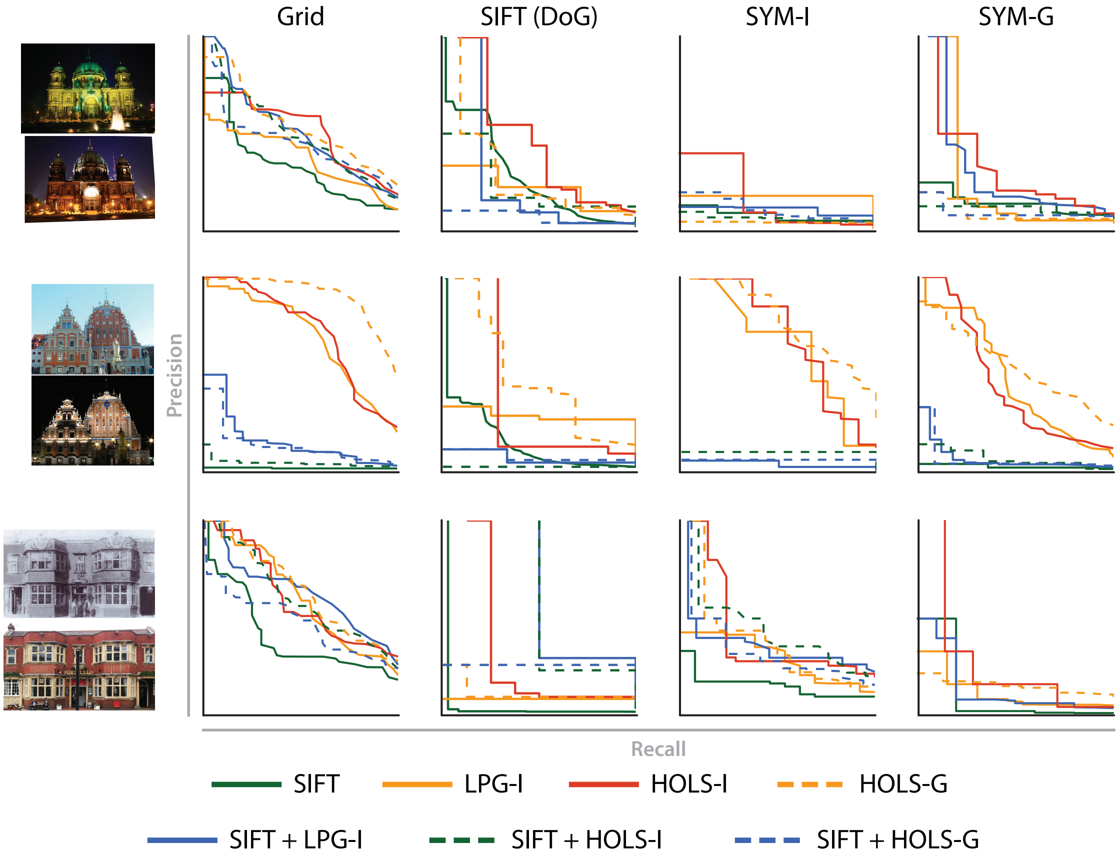


Figure 7.9: Precision-recall curves for selected image pairs from the dataset. Each column corresponds to a different detector and each row to a different pairs of images.

	GRID	SIFT	SYM-I	SYM-G
SIFT	0.50	0.20	0.24	0.25
LPG-I	0.41	0.18	0.21	0.26
HOLS-I	0.50	**0.25**	0.26	0.29
HOLS-G	**0.53**	**0.25**	**0.27**	**0.30**
SIFT-LPG-I	0.58	**0.26**	**0.30**	**0.34**
SIFT-HOLS-I	**0.59**	**0.26**	0.28	0.32
SIFT-HOLS-G	0.53	0.24	0.26	0.28

Table 7.2: Mean average precision for different combinations of detector and descriptor on our dataset. Rows represent the descriptors and columns the detectors. First four rows show performance for individual descriptors while last three rows show the performance for concatenated descriptors. Results for detectors SIFT, SYM-I, and SYM-G are over a subset of the images.

7.6 Conclusion and Future Work

In this chapter we demonstrated that symmetry is a powerful cue which is well preserved across very drastic image changes. We start by defining local symmetry and introduce two algorithms for computing it, one based on raw image intensities and another one that encodes information about image gradients as histograms of oriented gradients. Once we have a measure of local symmetry we show that it can be used to define a symmetry based scale space by computing the symmetry distance on an image pyramid. We then extract features by searching for local maxima in the symmetry based scale space. Finally, we introduced two different encodings of symmetry into descriptors: LPG and HOLS.

In the experimental section we presented a new dataset of image pairs, with dramatic appearance changes, and showed that our features are more repeatable than DoG features used by SIFT. We also show that in this dataset our descriptors perform better than SIFT. The best performance though is achieved when combining symmetry based descriptors with the gradient based SIFT descriptor, demonstrating that the two descriptors complement each other.

In the future we would like to explore more comprehensive descriptor encodings. There are two avenues of future research that could be pursued. First, by encoding symmetries across scales into one descriptor we could encode fine details of the image pattern. Second, it would be also interesting to investigate what is the impact of encoding a broader range of types of symmetry.

BIBLIOGRAPHY

[1] Photometric Ambient Occlusion webpage. `http://www.cs.cornell.edu/projects/photoao`.

[2] Austin Abrams, Christopher Hawley, and Robert Pless. Heliometric stereo: Shape from sun position. In *ECCV*, 2012.

[3] J. Ackermann, F. Langguth, S. Fuhrmann, and M. Goesele. Photometric stereo for outdoor webcams. In *CVPR*, 2012.

[4] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building Rome in a day. In *ICCV*, 2009.

[5] Oswald Aldrian and William AP Smith. Inverse rendering of faces on a cloudy day. In *ECCV*. 2012.

[6] Jonathan T Barron and Jitendra Malik. High-frequency shape and albedo from shading using natural image statistics. In *CVPR*, 2011.

[7] Jonathan T Barron and Jitendra Malik. Color constancy, intrinsic images, and shape estimation. In *ECCV*, 2012.

[8] Jonathan T. Barron and Jitendra Malik. Shape, albedo, and illumination from a single image of an unknown object. In *CVPR*, 2012.

[9] Jonathan T. Barron and Jitendra Malik. Intrinsic scene properties from a single rgb-d image. In *CVPR*, 2013.

[10] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. Technical Report UCB/EECS-2013-117, EECS, UC Berkeley, May 2013.

[11] R. Basri, D. Jacobs, and I. Kemelmacher. Photometric stereo with general, unknown lighting. *IJCV*, 2007.

[12] Ronen Basri and David W. Jacobs. Lambertian reflectance and linear subspaces. *PAMI*, 25(2), February 2003.

[13] Thabo Beeler, Derek Bradley, Henning Zimmer, and Markus Gross. Improved reconstruction of deforming surfaces by cancelling ambient occlusion. In *ECCV*, 2012.

[14] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 2002.

[15] Daniel Cabrini Hauagge and Noah Snavely. Image matching using local symmetry features. In *CVPR*, 2012.

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

[17] Manmohan Chandraker, Sameer Agarwal, and David Kriegman. Shadowcuts: Photometric stereo with shadows. In *CVPR*, 2007.

[18] Xiaowu Chen, Ke Wang, and Xin Jin. Single image based illumination estimation for lighting virtual object in real scene. In *International Conf. on CAD/Graphics*, 2011.

[19] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[20] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH*, 1998.

[21] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH*, 1997.

[22] V. Di Gesù, V. Di Gesu, and C. Valenti. The Discrete Symmetry Transform in Computer Vision. Technical report, Universita di Palermo, 1995.

[23] Mauricio Díaz and Peter Sturm. Radiometric calibration using photo collections. In *Int. Conf. on Computational Photography*, 2011.

[24] Mauricio Díaz and Peter F. Sturm. Estimating photometric properties from image collections. *J. Mathematical Imaging and Vision*, 47(1-2), 2013.

[25] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010.

[26] Yasutaka Furukawa. Clustering views for multi-view stereo (cmvs). `http://www.di.ens.fr/cmvs/`.

[27] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. In *CVPR*, 2007.

[28] Rahul Garg, Hao Du, Steven M. Seitz, and Noah Snavely. The dimensionality of scene appearance. In *ICCV*, 2009.

[29] Michael D Grossberg and Shree K Nayar. What is the space of camera response functions? In *CVPR*, 2003.

[30] R. Grosse, M.K. Johnson, E.H. Adelson, and W.T. Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*, 2009.

[31] 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.

[32] Roger Grosse, Micah K. Johnson, Edward H. Adelson, and William T. Freeman. MIT Intrinsic Images, 2009. `http://people.csail.mit.edu/rgrosse/intrinsic/`.

[33] Tom Haber, Christian Fuchs, Philippe Bekaer, H-P Seidel, Michael Goesele, and Hendrik PA Lensch. Relighting objects from image collections. In *CVPR*, 2009.

[34] Daniel Hauagge, Scott Wehrwein, Kavita Bala, and Noah Snavely. Photometric ambient occlusion. *CVPR*, 2013.

[35] Daniel Hauagge, Scott Wehrwein, Paul Upchurch, Kavita Bala, and Noah Snavely. Reasoning about photo collections using outdoor illumination models. In *Computer Vision–CVPR 2014. Workshops and Demonstrations*, 2014.

[36] Lukas Hosek and Alexander Wilkie. An analytic model for full spectral sky-dome radiance. *ACM Transactions on Graphics*, 2012.

[37] Nathan Jacobs, Nathaniel Roman, and Robert Pless. Consistent temporal variations in many outdoor scenes. In *CVPR*, 2007.

[38] Wenzel Jakob. Mitsuba renderer, 2010. `http://www.mitsuba-renderer.org`.

[39] Micah K. Johnson and Hany Farid. Exposing digital forgeries in complex lighting environments. *IEEE Trans. Information Forensics and Security*, 2(3), 2007.

[40] K. Karsch, K. Sunkavalli, S. Hadap, N. Carr, H. Jin, R. Fonte, M. Sittig, and D. Forsyth. Automatic scene inference for 3d object compositing. *ACM Transactions on Graphics*, 2014.

[41] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics*, 2011.

[42] E. Kee and H. Farid. Exposing digital forgeries from 3-d lighting environments. In *Workshop on Information Forensics and Security (WIFS)*, Dec 2010.

[43] Seon Joo Kim, Hai Ting Lin, Zheng Lu, Sabine Susstrunk, Steven Lin, and Michael S. BrownHai. A new in-camera imaging model for color computer vision and its application. 2012.

[44] J. Kontkanen and S. Laine. Ambient occlusion fields. In *Proc. Symp. on Interactive 3D Graphics and Games*. ACM, 2005.

[45] P. Kovesi. Symmetry and asymmetry from local phase. *Australian Joint Conf. on Artificial Intelligence*, 1997.

[46] P. Kovesi. Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1999.

[47] Sujit Kuthirummal, Aseem Agarwala, Dan B Goldman, and Shree K Nayar. Priors for large photo collections and what they reveal about cameras. In *ECCV*. Springer, 2008.

[48] Pierre-Yves Laffont, Adrien Bousseau, and George Drettakis. Rich intrinsic image decomposition of outdoor scenes from multiple views. *Trans. Visualization and Computer Graphics*, 2013.

[49] Pierre-Yves Laffont, Adrien Bousseau, Sylvain Paris, Frédo Durand, and George Drettakis. Coherent intrinsic images from photo collections. *SIGGRAPH Asia*, 2012. `http://www-sop.inria.fr/reves/Basilic/2012/LBPDD12`.

[50] 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.

[51] Jean-François Lalonde, Derek Hoiem, Alexei A. Efros, Carsten Rother, John Winn, and Antonio Criminisi. Photo clip art. *SIGGRAPH*, 2007.

[52] Jean-François Lalonde, Alexei A Efros, and Srinivasa G Narasimhan. Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. In *ACM Transactions on Graphics*, 2009.

[53] E.H. Land, J.J. McCann, et al. Lightness and retinex theory. *Journal of the Optical society of America*, 1971.

[54] H. Landis. Production-ready global illumination. *SIGGRAPH Course Notes*, 2002.

[55] M. S. Langer and S. W Zucker. Shape-from-shading on a cloudy day. *J. Optical Society of America A*, 1994.

[56] Kuang-Chih Lee, Jeffrey Ho, and David Kriegman. The Extended Yale Face Database B, 2005. `http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html`.

[57] Tony Lindeberg. *Scale-Space Theory in Computer Vision*. 1994.

[58] Y. Liu, Hagit Hel-Or, Craig Kaplan, and Luc Van Gool. Computational Symmetry in Computer Vision and Computer Graphics. *Foundations and Trends in Computer Graphics and Vision*, 2009.

[59] David G Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[60] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[61] G. Loy and A. Zelinsky. Fast radial symmetry for detecting points of interest. *PAMI*, 2003.

[62] Gareth Loy and Jan-Olof Eklundh. Detecting symmetry and symmetric constellations of features. In *ECCV*, 2006.

[63] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool. A comparison of affine region detectors. *IJCV*, 2005.

[64] Peter Nillius and Jan-Olof Eklundh. Automatic estimation of the projected light source direction. In *CVPR*, 2001.

[65] NOAA. Quality controlled local climatological data. `http://cdo.ncdc.noaa.gov/qclcd/QCLCD`. 01/2013.

[66] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001.

[67] J. Pantaleoni, L. Fascione, M. Hill, and T. Aila. PantaRay: Fast ray-traced occlusion caching of massive scenes. In *ACM Transactions on Graphics*, 2010.

[68] Matt Pharr and Simon Green. Ambient occlusion. *GPU Gems*, 2004.

[69] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser. A planar-reflective symmetry transform for 3D shapes. In *SIGGRAPH*, 2006.

[70] Emmanuel Prados, Nitin Jindal, and Stefano Soatto. A non-local approach to shape from ambient shading. In *Scale Space and Variational Methods in Computer Vision*. Springer, 2009.

[71] A. J. Preetham, Peter Shirley, and Brian Smits. A practical analytic model for daylight. In *SIGGRAPH*, 1999.

[72] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *SIGGRAPH*, 2001.

[73] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *SIGGRAPH*, 2001.

[74] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context-free attentional operators: the generalized symmetry transform. *IJCV*, 1995.

[75] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *IJCV*, 2000.

[76] Grant Schindler and Frank Dellaert. Probabilistic temporal inference on reconstructed 3d scenes. In *CVPR*, 2010.

[77] Grant Schindler, Panchapagesan Krishnamurthy Roberto Lublinerman, Yanxi Liu, and Frank Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *CVPR*, 2008.

[78] Qi Shan, Riley Adams, Brian Curless, Yasutaka Furukawa, and Steven M. Seitz. The visual Turing test for scene reconstruction. 2013.

[79] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007.

[80] J. Shen, X. Yang, X. Li, and Y. Jia. Intrinsic image decomposition using optimization and user scribbles. *Trans. Systems, Man, and Cybernetics*, 2012.

[81] Li Shen and Ping Tan. Photometric stereo and weather estimation using internet images. 2009.

[82] Li Shen and Chuohao Yeo. Intrinsic images decomposition using a local and global sparse representation of reflectance. In *CVPR*, 2011.

[83] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Data-driven visual similarity for cross-domain image matching. *SIGGRAPH Asia*, 2011.

[84] Noah Snavely. Bundler: Structure from motion (sfm) for unordered image collections. `http://phototour.cs.washington.edu/bundler/`.

[85] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring image collections in 3d. In *SIGGRAPH*, 2006.

[86] Kalyan Sunkavalli, Wojciech Matusik, Hanspeter Pfister, and Szymon Rusinkiewicz. Factored time-lapse video. In *SIGGRAPH*, 2007.

[87] Kalyan Sunkavalli, Todd Zickler, and Hanspeter Pfister. Visibility subspaces: Uncalibrated photometric stereo with shadows. In *ECCV*, 2010.

[88] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 1991.

[89] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV*, 2001.

[90] Daniel N. Wood, Daniel I. Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H. Salesin, and Werner Stuetzle. Surface light fields for 3D photography. In *SIGGRAPH*, 2000.

[91] R.J. Woodham. Analysing images of curved surfaces. *Artificial Intelligence*, 1981.

[92] C. Wu, B. Wilburn, Y. Matsushita, and C. Theobalt. High-quality shape from multi-view stereo and shading under general illumination. In *CVPR*, 2011.

[93] Changchang Wu, Jan-Michael Frahm, and Marc Pollefeys. Detecting large repetitive structures with salient boundaries. In *ECCV*, 2010.

[94] T.P. Wu and C.K. Tang. Photometric stereo via expectation maximization. *PAMI*, 2010.

[95] Guanyu Xing, Xuehong Zhou, Qunsheng Peng, Yanli Liu, and Xueying Qin. Lighting simulation of augmented outdoor scene based on a legacy photograph. *Comput. Graph. Forum*, 32(7), 2013.

[96] Lap-Fai Yu, Sai-Kit Yeung, Yu-Wing Tai, Demetri Terzopoulos, and Tony Chan. Towards Outdoor Photometric Stereo. 2013.

[97] C. Lawrence Zitnick. Binary coherent edge descriptors. In *ECCV*, 2010.

[98] C. Lawrence Zitnick and Krishnan Ramnath. Edge foci interest points. In *ICCV*, 2011.