SIGGRAPH 2002 Course 39

Acquiring Material Models Using Inverse Rendering

Organizers

Ravi Ramamoorthi
Steve Marschner
Stanford University

Lecturers

Samuel Boivin
University of Toronto

George Drettakis
INRIA Sophia-Antipolis

Hendrik P. A. Lensch
Max-Planck-Institut für Informatik

Yizhou Yu
University of Illinois at Urbana-Champaign

Description

Accurate, detailed models of the materials in a scene are important to achieve the full potential for realism in today's rendering systems. In order to fill this need, several recent research efforts have looked into capturing the appearance of materials from sets of photographs, resulting in models that have the required fine detail and correct behavior. This advanced course presents this recent work, with most of the presentations given by the researchers themselves. Both relevant background information and specific practical methods will be presented.

Prerequisites

Working knowledge of how materials are described for realistic rendering, including texture maps and the BRDF. Familiarity with shading and reflectance models and their use in rendering.

Syllabus

Session 1: Inverse methods

Introduction and preliminaries [Marschner]
Determining reflectance for interactive relighting [Drettakis]
Fitting complex material properties from a single image [Boivin]
Inverse rendering under complex illumination [Ramamoorthi]

— break —

Session 2: Complex material properties

Measuring spatial variation with complex BRDFs [Lensch]
Modeling and synthesizing 3D textures [Yu]
Image-based measurement of translucent materials [Marschner]
Q&A [all]

Introduction

Creating photorealistic computer-generated images requires accurate physically-based rendering algorithms, as well as accurate and detailed models of the scene geometry, illumination, and reflective properties of the objects. Simulating light transport has been a long-standing problem in computer graphics on which much progress has been made in the last two decades. Therefore, realism is today often limited by the quality of input models for geometry, illumination, and reflectance. There has been a signficant body of work in computer vision and more recently in graphics, including recent Siggraph courses, on active and passive *3D photography* to acquire object geometry from real objects. Natural illumination can often be measured simply using a lightprobe, such as a photograph of a mirror sphere. In a similar fashion, a promising way of obtaining realistic reflective properties involves acquiring and modeling material properties from photographs. This course brings together much of the recent work on modeling material properties from sequences of photographs. The potential applications range over all areas where realistic rendering is important, including games, visualization, computer-aided modeling and design, film visual effects, virtual reality, and image processing.

Our problem differs from much of computer vision in that we do not seek to estimate shape, and our methods will usually assume geometry is known using 3D photography methods. Nevertheless, we seek to solve an inverse problem. Our inputs are the normal outputs of the forward rendering process, i.e. real images, and our outputs are the inputs to a rendering algorithm—illumination and reflective properties. For this reason, we refer to the class of algorithms as *Inverse Rendering* methods.

In some ways, our efforts are related to recent research on *image-based rendering.* In its simplest form, image-based rendering uses view interpolation to construct new images from acquired images without constructing a conventional scene model. We go further in actually solving an inverse problem to derive scene attributes, primarily the reflective properties of the surfaces. This allows us to improve the quality of view interpolation using image-based rendering, or to use traditional rendering algorithms with realistic input parameters. Since we have measured the scene attributes, there is structure introduced into the raw imagery, allowing us to manipulate the scene in ways not usually possible with standard image-based techniques. For instance, an artist could change the illumination, producing realistic rerenderings under new lighting conditons.

While there has been a significant amount of theoretical and practical work in the last two decades on forward rendering, we are just beginning to understand the mathematical foundations and practical algorithms for inverse rendering. This course presents the state of the art in the area.

The speakers in the first session discuss the newest methods for acquiring and modeling reflective properties. George Drettakis shows how reflectance properties can be determined and used for interactive relighting. Following this, Samuel Boivin discusses automatic methods for estimating complex material properties from a single image. When solving inverse problems, especially under complex illumination, a number of ambiguities may arise which do not arise for forward rendering. Ravi Ramamoorthi ends the session by presenting a signal-processing framework for inverse rendering, identifying the nature of these ambiguities, and discussing how this analysis can lead to improved and more robust inverse rendering algorithms.

While traditional computer graphics algorithms have usually considered material properties specified by a single BRDF, possibly modulated by a texture map, real materials often require more complex descriptions. The second session of this course delves into these more advanced models. First, Hendrik Lensch shows how to estimate spatially varying BRDF properties on a surface. Yizhou Yu discusses estimation and synthesis of Bi-Directional Texture Functions (BTF), a concept that extends the 4D BRDF to 6D, and allows for textures that vary depending on the viewing and illumination directions. Finally, Steve Marschner discusses measurement and modeling of subsurface scattering effects, as expressed by the Bi-directional Surface Scattering Distribution Function (BSSRDF).

Our objective in this course is to collect together much of the recent work on inverse rendering methods to obtain reflective properties, giving an overview of the fundamentals as well as discussing specific practical algorithms. Bear in mind that this is a topic of current research, so this course is necessarily a snapshot of the current state of the field. We are sure there will be many exciting developments in the years to come.

Ravi Ramamoorthi and Steve Marschner
Stanford University
April 2002

Chapter 1

Determining Reflectance for Interactive Relighting

George Drettakis, **REVES/INRIA Sophia-Antipolis,**
http://www-sop.inria.fr/reves

1.1 Introduction

Our goal is to allow relighting at *interactive* rates, i.e., to permit a user to modify the lighting conditions of a real scene, which has been captured in digital form. In this part of the course, we present two methods in which a complete system was developed, from initial capture of the scene's geometry, lighting conditions, reflectance properties and geometry, and the special structures used to allow interactive modifications and viewing of real world lighting. The two methods are the papers [LFD$^+$99] and [LDR00]; the reader is also encouraged to look at [LD00], which discusses some practical issues related to low-cost solutions for high-dynamic range images. The URL's for these papers are given in the reference list.

In both methods, we have attempted to have a relatively low-cost and simple-as-possible capture process. We will be discussing some additional issues which are not in the presentation slides.

Both approaches are based on the use of hierarchical *radiosity* to estimate illumination conditions in the scene. The idea of using radiosity to simulate real-world lighting was first introduced by Fournier et al. [FGR93]. In the slides of the presentation, we show how to use the interactive extension to radiosity [DS97], to achieve interactive update of the display. In Section 1.3 we introduce some simple notions required to understand hierarchical radiosity.

Using a radiosity system for relighting requires an initialisation step which requires the extraction of *reflectance* for all the objects in the scene. This is a complex task, and we choose to use an *approximation*. A first approximation is to simply remove real shadows from the scene, and initialise the system in a way which allows a radiosity computation to take place [LDR00]. Extracting an approximate reflection requires more information during capture. These issues are discussed briefly in Section 1.4.

In Section 1.5, we discuss a few more practical issues which were encountered when implementing the interactive display for these methods.

Finally in 1.6 we discuss some issues of future work and an overview of where we think the area is going.

1.2 Capture

The two methods presented in the course talk tried to focus on low-cost solutions. As a result, we used a consumer level digital camera (a Kodak DC260) on a tripod. In retrospect, the resolution and the sensitivity of this camera is simply insufficient.

In work we have done since, we use a high-end digital camera (the Canon D30 is what we currently use). The main required feature is manual exposure control, which typically is only available on SLR digital cameras (”pro” cameras). The advantage is that these cameras are now relatively cheap, notably about half the price of what they were 3 years ago. We also always create true HDR images, using the free HDRshop software provided by Paul Debevec’s group (http://www.debevec.org). What we noticed is that it is worth investing in a heavy tripod; a 5 kg tripod will keep your camera stable enough even for manual exposure control.

With the appearance of cheap, high-resolution digital cameras, some of these issues will no longer be important.

However, the capture process is always quite painful, and always longer than what you imagine. We currently use RealViz products for camera calibration and geometry reconstruction. One possible technique is to use the new version of MatchMover to automatically calibrate the cameras from multiple views, and then use ImageModeler to actually generate the model. The newer versions of ImageModeler are slightly more user-friendly, but geometry reconstruction from images is always an involved process.

In terms of lighting, for the method described in [LFD$^+$99], we used a garden lamp with rice paper as a diffuser. This proved sufficient for the level of quality we required, and allowed us to move the lamp around simply.

3D scanning technologies, especially structured light approaches may prove to be much more appropriate in the relatively near future.

1.3 Radiosity

As mentioned in the introduction above, the two methods presented in this talk, use hierarchical radiosity to compute global illumination. These algorithms were initially developed for entirely synthetic scenes, in which the light sources and emission properties, entire geometry and reflectance properties are defined by the modelling program. In our context, we do not have any of this information available. As a consequence, we need to estimate or arbitrarily fix some of the parameters by hand to perform a calculation that can be subsequently used by the radiosity simulation. How exactly this is done, is explained in the initial paper by Fournier et al. [FGR93] and later in Drettakis et al. [DRB97] as well as the two references provided. In what follows we provide a rapid overview of the hierarchical radiosity methods used.

Radiosity methods (e.g., [GTGB84, CCWG88]) are able to simulate global illumination for diffuse-only environments, i.e., scenes in which objects materials are all lambertian. In particular hierarchical radiosity with clustering (HRC) [SAG94, Sil95] can simulate radiosity for large environments rapidly. One advantage of radiosity is that the result of the simulation can be interactively visualized for walkthroughs on graphics hardware, and in some cases interactive updates can be made [DS97]. The important steps and structures of HRC are reviewed in Fig. 1.1.

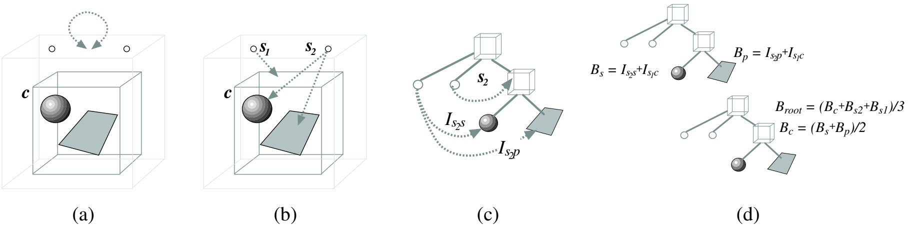

c
c
s_1
s_2
s_2
I_{s_2s}
I_{s_2p}
$B_s = I_{s_2s}+I_{s_1c}$
$B_p = I_{s_2p}+I_{s_1c}$
$B_{root} = (B_c+B_{s2}+B_{s1})/3$
$B_c = (B_s+B_p)/2$
(a)
(b)
(c)
(d)

Figure 1.1: Basic steps of the hierarchical radiosity algorithm: *Initialization:* the scene, consisting of a sphere and a polygon, contained in a cluster c, and two light sources. Initially there is a *self link* to the root cluster. For each iteration we perform three steps.(a) *Refinement*: a link is created between source s_1 and c, and s_2 is linked to the sphere s and the polygon p. (c) *Gather*: irradiance is gathered over the links, resulting in the values I_{s_2s} (s_2 to s) and I_{s_2p} (s_2 to p) shown in the graph representation of the hierarchy. (d) *Push-pull*: First the irradiance values are pushed to the leaves, where radiosity is set by reflecting the irradiance using reflectance ρ (top). The radiosity values are then pulled up the hierarchy by area weighted averages to maintain consistency (bottom).

1.4 Shadow Removal vs. Reflectance Estimation

In the two methods we present here, as well as in other previous papers [FGR93, DRB97], we use the radiosity simulation as a way to estimate original real-world lighting conditions, and then to modify them, typically by computing a difference or a ratio. It is thus important to note that we do not compute *correct* real-world illumination values, but a coarse approximation.

This is why using a diffuse-only global illumination calculation, which has many advantages in terms of speed in interactive settings, is acceptable. To achieve a more precise computation, such as that performed in [YDMH99], a much more involved capture process needs to be performed (calibrated lights, more pictures, specific geometric configurations etc.). In that paper, to a certain extent, one can expect that the inverse lighting computation is sufficiently accurate to be considered as a true simulation of real light.

For our purposes, that is simple capture and interactive updates, we typically cannot afford these types of constraints. The first method we present [LFD$^+$99] simply removes shadows from the real scene, using the approximate geometric model and a radiosity computation, as well as some image-processing techniques to clean up the result. As a consequence, we end up with a texture which represents original real lighting, ”without shadows”. Thus the texture *does* include the lighting at the time of capture. We can then modulate the texture with the radiosity mesh to add, turn on or turn off lights.

For the second method [LDR00], we adopt a different approach, in which we use multiple images to extract an approximate reflectance, so we try to *remove* the effects of current real lighting. This is not completely sucessful, but the quality of the result can suffice in certain cases.

1.5 Interactive Display

The first method [LDR00] is based entirely on the display of textured polygons resulting from the subdivided radiosity mesh. Given that the geometric complexity of these scenes is overall very small, and given the polygon rendering capacities of todays hardware, we can achieve interactive updates and display. However, as can be seen in the images, and as is always the case for radiosity meshes, other than those using discontinuity lines, shadows boundaries are not well captured, and thus the quality suffers. Our implementations were on the original Onyx IR’s, but modern PC-based cards should be able to achieve much faster display rates, and one should be able to use the multi-texturing hardware and pixel shaders effectively.

It is for this reason that we decided to move to a pixel-based model for the method based on multiple lighting conditions in [LFD$^+$99]. We use a “deep framebuffer” approach, which allows us to store reflectance and shadow information per-pixel, and which allows relatively rapid display. Evidently, this type of approach is resolution-limited, even though with todays processor speed this should be less of a problem. Memory requirements for this kind of approach can also be problematic.

Gibson and Murta [GM00], presented a very interesting approach using hardware acceleration for shadow generation for real-world lighting.

1.6 Discussion and Future Work

It is currently safe to say that a general-purpose interactive relighting algorithm for real scenes is still out of reach for practical applications. All algorithms presented to date suffer from limitations and difficulties, either in geometric or photometric capture, calibration of light sources or capture devices, inverse lighting computations, approximations used for lighting simulation or reflectance extraction, and difficulties for achieving *interactive* display.

Gibson et al. [GHH01], are investigating different kinds of solutions, in particular trying to resolve the inherenet problem of lack of geometric detail for most of the scene.

This, and other approximate approaches rely on minimization techniques, which seem particularly promising.

Our current focus is on outdoor scenes, which pose particular problems in terms of the geometry of natural objects which is very complex, but, in some cases, is easier in terms of lighting conditions since sun and skylight usually dominate, and indirect lighting is often less important.

For such applications, we believe that user-controlled image-based editing techniques should be combined with automatic or semi-automatic methods. In the same manner that user intervention has allowed vision algorithms to be used in practical products, we believe that some level of user intervention can help to render relighting algorithms usable.

Bibliography

[CCWG88] M. F. Cohen, S. E. Chen, J. R. Wallace, and D. P. Greenberg. A progressive refinement approach to fast radiosity image generation. In *Computer Graphics (SIGGRAPH 88)*, volume 22(4), pages 75–84, August 1988.

[DRB97] George Drettakis, Luc Robert, and Sylvain Bougnoux. Interactive common illumination for computer augmented reality. In Julie Dorsey and Phillipp Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 45–56, New York, NY, 1997. Springer Wien.

[DS97] G. Drettakis and F. Sillion. Interactive update of global illumination using a line-space hierarchy. In *Proc. SIGGRAPH'97*, Annual Conference Series, pages 57–64, August 1997.

[FGR93] A. Fournier, A.S. Gunawan, and C. Romanzin. Common illumination between real and computer generated scenes. In *Proc. of Graphics Interface '93*, pages 254–262, May 1993.

[GHH01] Simon Gibson, Toby Howard, and Roger Hubbold. Flexible image-based photometric reconstruction using virtual light sources. *Computer Graphics Forum*, 20(3), 2001. ISSN 1067-7055.

[GM00] Simon Gibson and Alan Murta. Interactive rendering with real world illumination. In B. Peroche and H. Rushmeier, editors, *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, New York, NY, 2000. Springer Wien. 365–376.

[GTGB84] C. M. Goral, K. K. Torrance, D. P. Greenberg, and B. Battaile. Modelling the interaction of light between diffuse surfaces. In *Computer Graphics (SIGGRAPH'84)*, volume 18(3), pages 213–222, July 1984.

[LD00] Cline Loscos and George Drettakis. Low-cost photometric calibration for interactive relighting. In *Proceedings of the First French-British International Workshop on Virtual Reality*, 2000. http://www-sop.inria.fr/reves/publications/data/2000/LD00/.

[LDR00] C. Loscos, G. Drettakis, and L. Robert. Interactive virtual relighting of real scenes. *IEEE Transactions on Visualization* *and Computer Graphics*, 6(3), july-september 2000. http://www-sop.inria.fr/reves/publications/data/2000/LDR00/.

[LFD$^+$99] C. Loscos, M.-C. Frasson, G. Drettakis, B. Walter, X. Granier, and P. Poulin. Interactive virtual relighting and remodeling of real scenes. In *Rendering Techniques '99 (10th Eurographics Workshop on Rendering)*, pages 329–340. Springer-Verlag, June 1999. http://www-sop.inria.fr/reves/publications/data/1999/LFDWGP99/.

[SAG94] B. Smits, J. Arvo, and D. P. Greenberg. A clustering algorithm for radiosity in complex environments. In *Proc. SIGGRAPH'94*, Annual Conference Series, pages 435–442, July 1994.

[Sil95] F. X. Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Trans. on Visualization and Computer Graphics*, 1(3):240–254, September 1995.

[YDMH99] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs from. In Alyn Rockwood, editor, *Siggraph99, Annual Conference Series*, pages 215–224, Los Angeles, 1999. Addison Wesley Longman.

Determining reflectance for interactive relighting

George Drettakis

REVES/INRIA Sophia-Antipolis

http://www-sop.inria.fr/reves

Motivation

- Applications
  - Augmented reality, with consistent and modifiable real-world lighting
  - Real-world lighting design
  - Film postproduction

Motivation

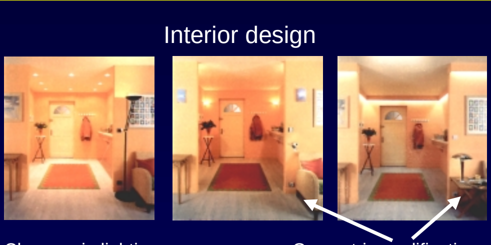
- Re-light a digitized real-world scene
  - including changes in real-world lighting
- Interactive display
  - allows changes to be tested and evaluated
- Limited re-modelling may be desirable

Motivation

- Goal: interactive system
  - simple capture process
  - interactive (at least 1 Hz)
  - modification of real world lighting
  - addition of virtual lights

Motivation

Interior design

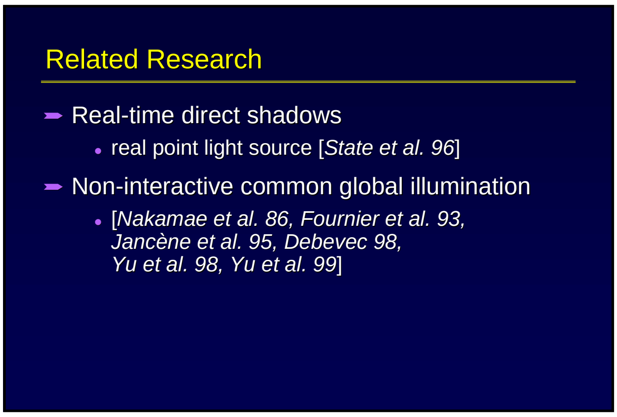


Changes in lighting

Geometric modification

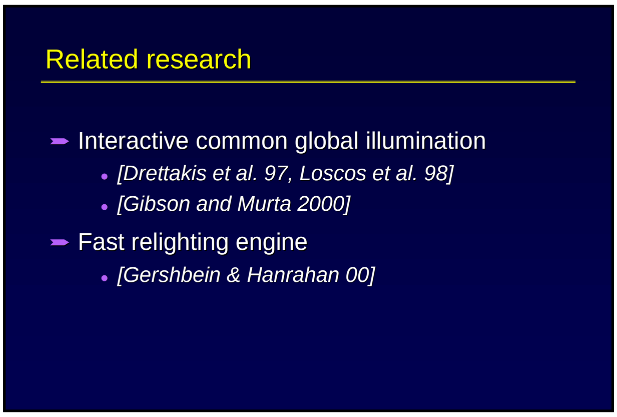
Related research

- Geometric reconstruction
  - vision methods [*Faugeras et al. 97, ...*] (*Realise/RealViz*)
  - constraint-based systems [*Debevec et al. 96*, *Poulin et al. 98*]
  - software: Photomodeler, etc.
- Reflectance recovery
  - e.g., [*Sato et al. 97, Ward92, Debevec98, Yu et al. 98, etc*].



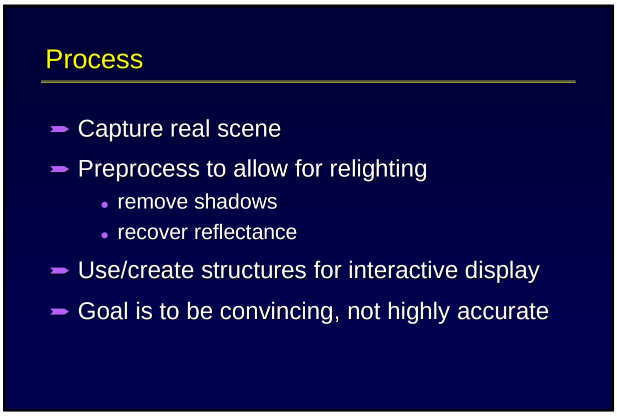
Related Research

- Real-time direct shadows
  - real point light source [*State et al. 96*]
- Non-interactive common global illumination
  - [*Nakamae et al. 86, Fournier et al. 93, Jancène et al. 95, Debevec 98, Yu et al. 98, Yu et al. 99*]



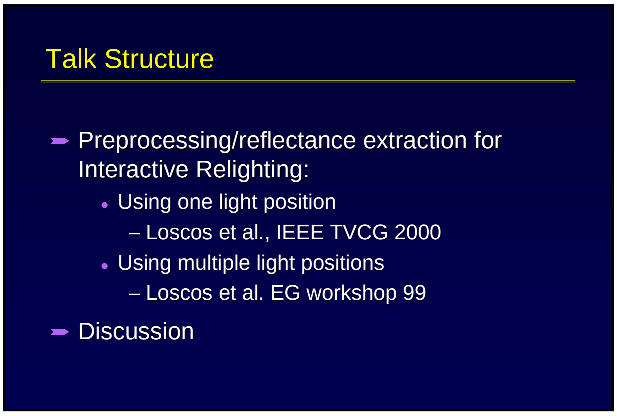
Related research

- Interactive common global illumination
  - *[Drettakis et al. 97, Loscos et al. 98]*
  - *[Gibson and Murta 2000]*
- Fast relighting engine
  - *[Gershbein & Hanrahan 00]*



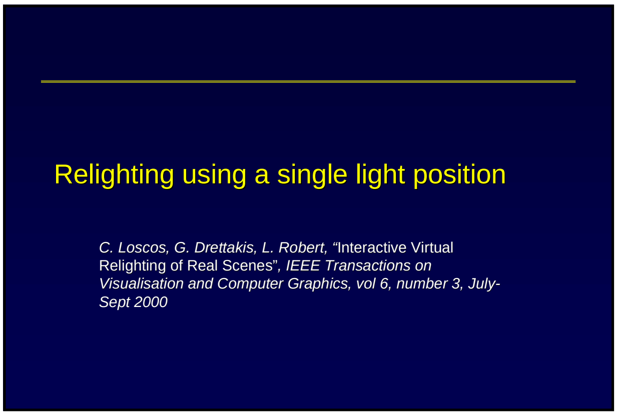
Process

- Capture real scene
- Preprocess to allow for relighting
  - remove shadows
  - recover reflectance
- Use/create structures for interactive display
- Goal is to be convincing, not highly accurate



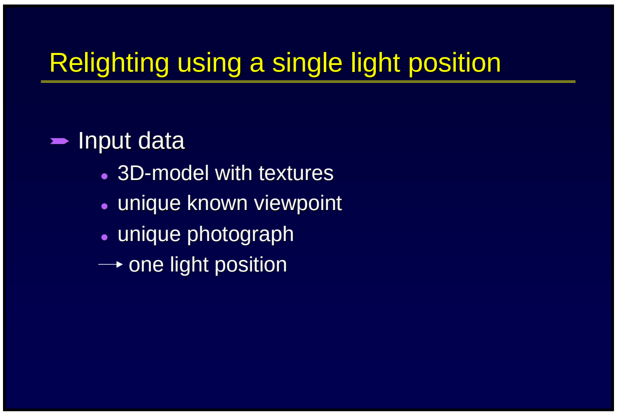
Talk Structure

- Preprocessing/reflectance extraction for Interactive Relighting:
  - Using one light position
    - – Loscos et al., IEEE TVCG 2000
  - Using multiple light positions
    - – Loscos et al. EG workshop 99
- Discussion



Relighting using a single light position

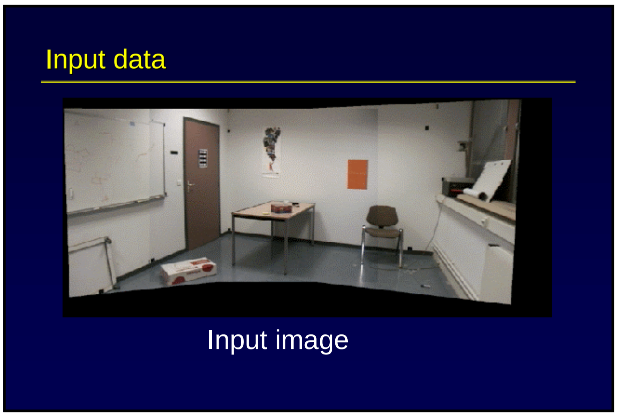
C. Loscos, G. Drettakis, L. Robert, “Interactive Virtual Relighting of Real Scenes”, *IEEE Transactions on Visualisation and Computer Graphics, vol 6, number 3, July-Sept 2000*



Relighting using a single light position

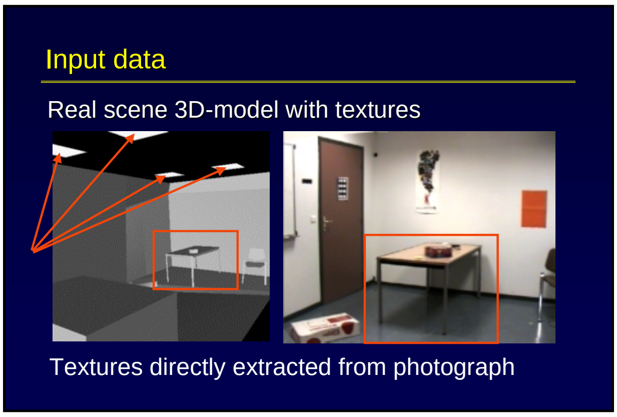
- Input data
  - 3D-model with textures
  - unique known viewpoint
  - unique photograph

  → one light position



Input data

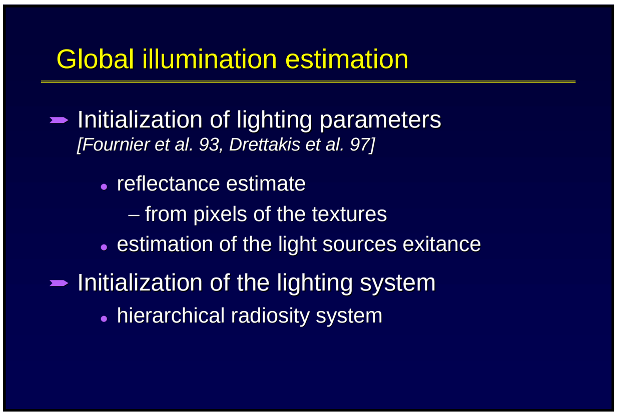
Input image



Input data

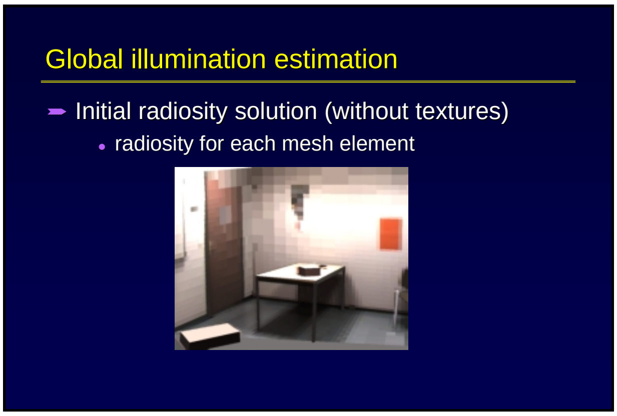
Real scene 3D-model with textures

Textures directly extracted from photograph



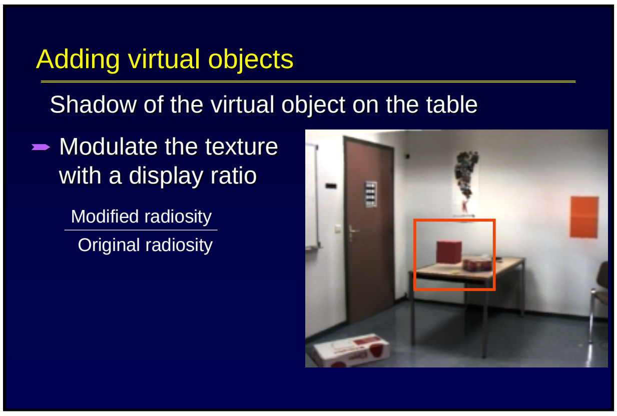
Global illumination estimation

- Initialization of lighting parameters
  [Fournier et al. 93, Drettakis et al. 97]
  - reflectance estimate
    - from pixels of the textures
  - estimation of the light sources exitance
- Initialization of the lighting system
  - hierarchical radiosity system



Global illumination estimation

- Initial radiosity solution (without textures)
  - radiosity for each mesh element

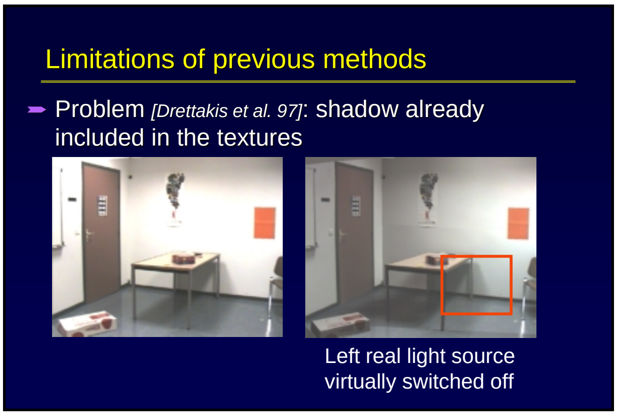


Adding virtual objects

Shadow of the virtual object on the table

- Modulate the texture with a display ratio

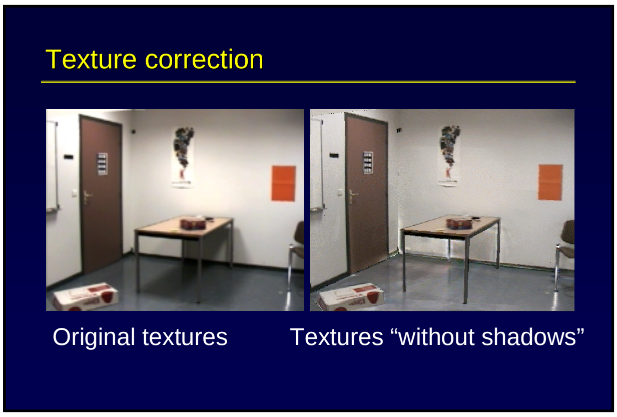
$$\frac{\text{Modified radiosity}}{\text{Original radiosity}}$$



Limitations of previous methods

- Problem *[Drettakis et al. 97]*: shadow already included in the textures

Left real light source virtually switched off



Texture correction

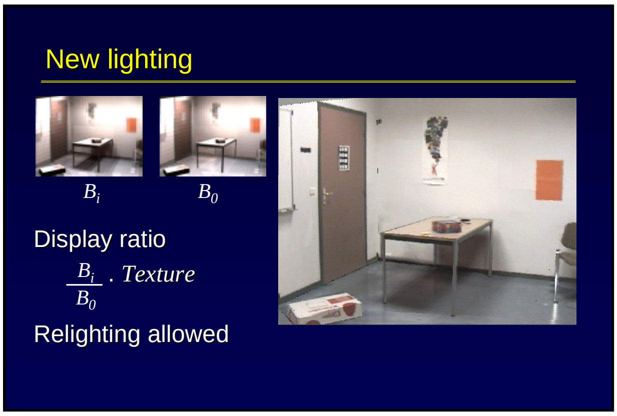
Original textures

Textures “without shadows”

New radiosity solution

Radiosity B_i

Radiosity B_0 ”without shadows”



New lighting

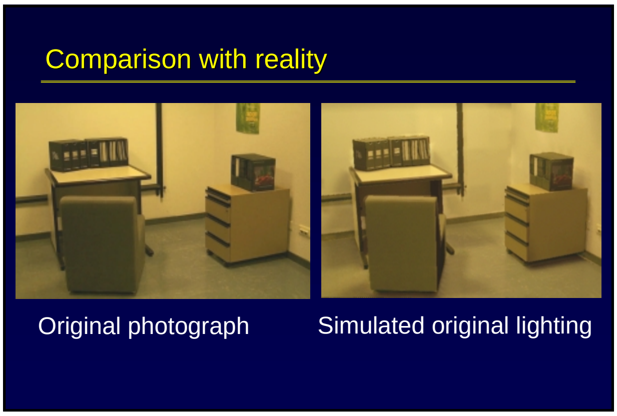
B_i

B_0

Display ratio

$$\frac{B_i}{B_0} \cdot \mathit{Texture}$$

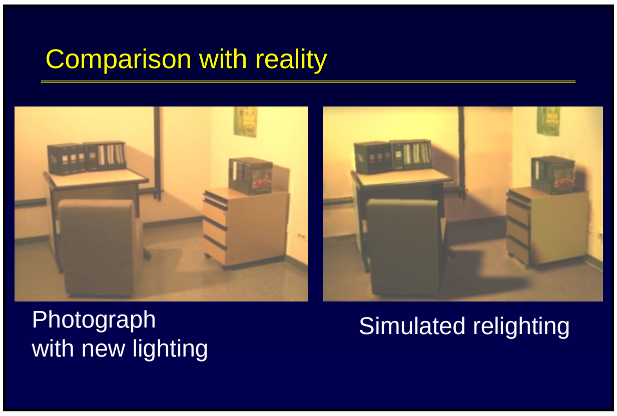
Relighting allowed



Comparison with reality

Original photograph

Simulated original lighting

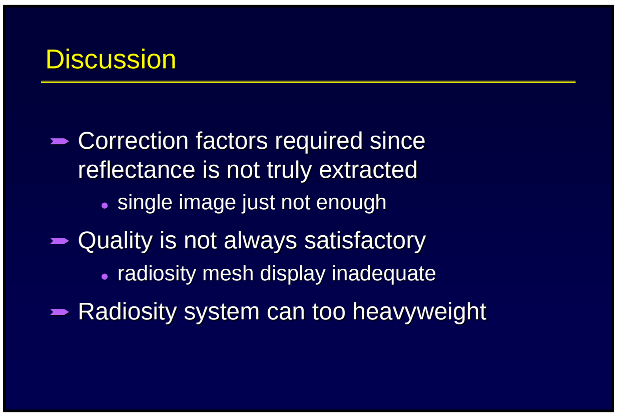


Comparison with reality

Photograph
with new lighting

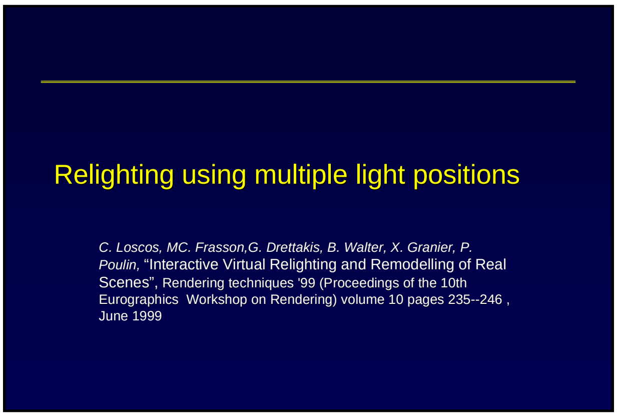
Simulated relighting

Video



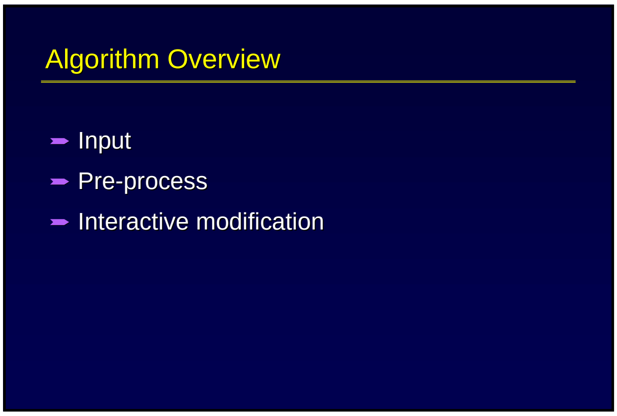
Discussion

- Correction factors required since reflectance is not truly extracted
  - single image just not enough
- Quality is not always satisfactory
  - radiosity mesh display inadequate
- Radiosity system can too heavyweight



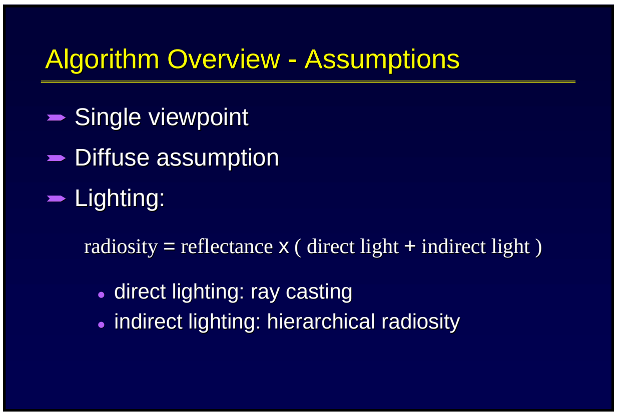
Relighting using multiple light positions

C. Loscos, MC. Frasson,G. Drettakis, B. Walter, X. Granier, P. Poulin, "Interactive Virtual Relighting and Remodelling of Real Scenes", Rendering techniques '99 (Proceedings of the 10th Eurographics Workshop on Rendering) volume 10 pages 235--246 , June 1999



Algorithm Overview

- Input
- Pre-process
- Interactive modification

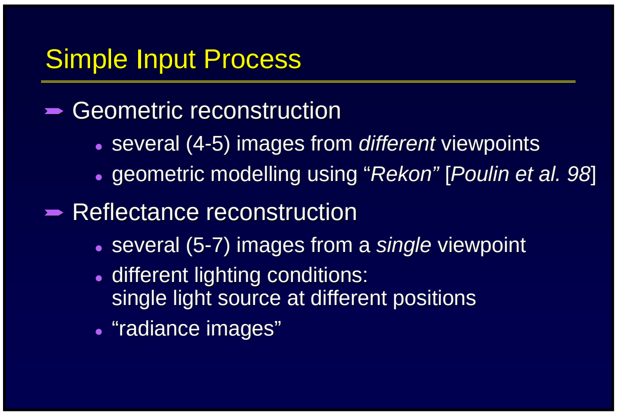


Algorithm Overview - Assumptions

- Single viewpoint
- Diffuse assumption
- Lighting:

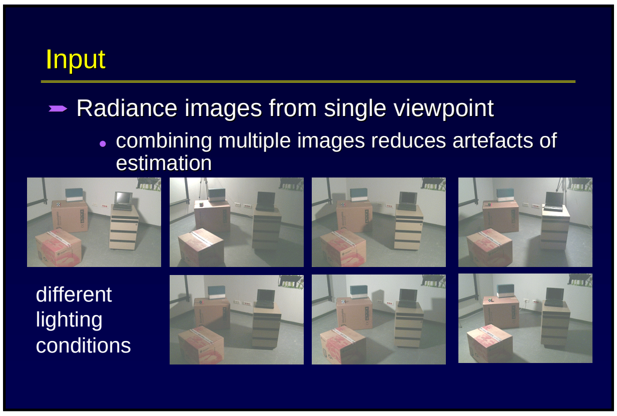
radiosity = reflectance x (direct light + indirect light)

- direct lighting: ray casting
- indirect lighting: hierarchical radiosity



Simple Input Process

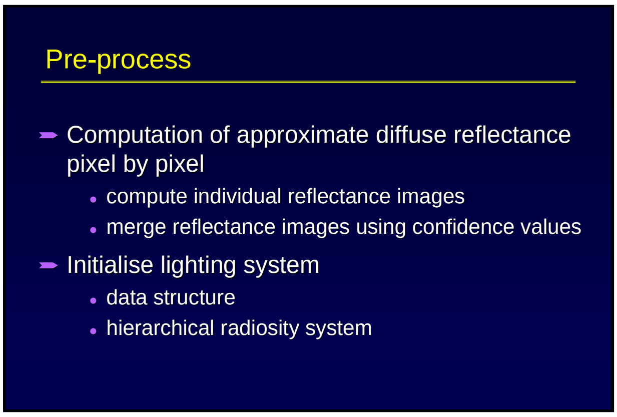
- Geometric reconstruction
  - several (4-5) images from *different* viewpoints
  - geometric modelling using *"Rekon"* [*Poulin et al. 98*]
- Reflectance reconstruction
  - several (5-7) images from a *single* viewpoint
  - different lighting conditions:
    single light source at different positions
  - "radiance images"



Input

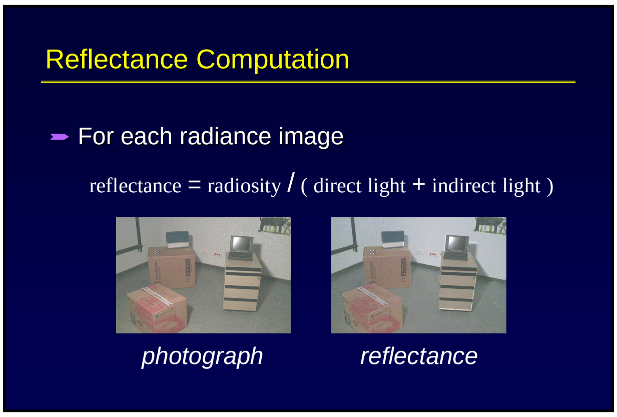
- Radiance images from single viewpoint
  - combining multiple images reduces artefacts of estimation

different lighting conditions



Pre-process

- Computation of approximate diffuse reflectance pixel by pixel
  - compute individual reflectance images
  - merge reflectance images using confidence values
- Initialise lighting system
  - data structure
  - hierarchical radiosity system

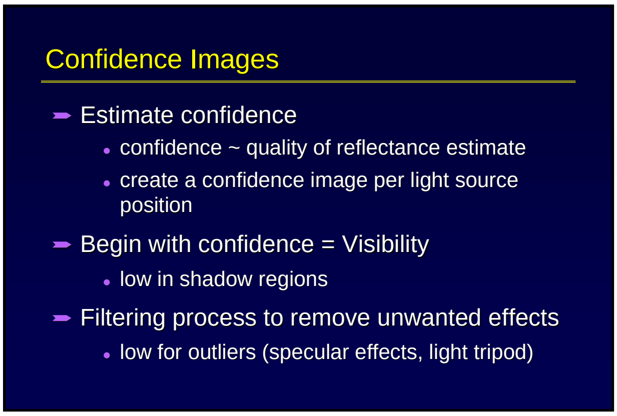


Reflectance Computation

- For each radiance image

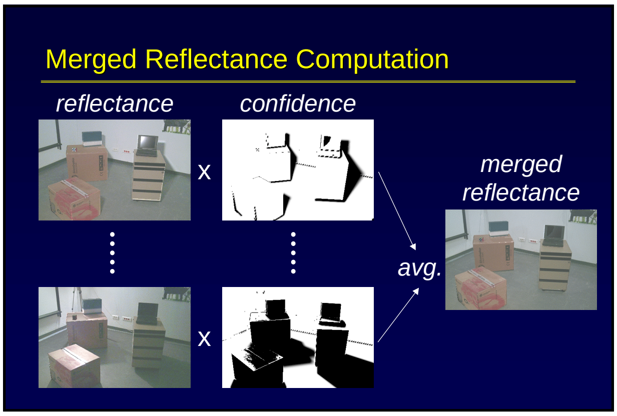
reflectance = radiosity / (direct light + indirect light)

photograph *reflectance*



Confidence Images

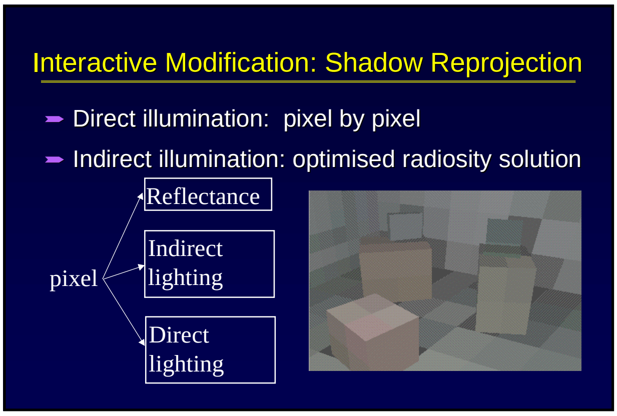
- Estimate confidence
  - confidence ~ quality of reflectance estimate
  - create a confidence image per light source position
- Begin with confidence = Visibility
  - low in shadow regions
- Filtering process to remove unwanted effects
  - low for outliers (specular effects, light tripod)



Merged Reflectance Computation

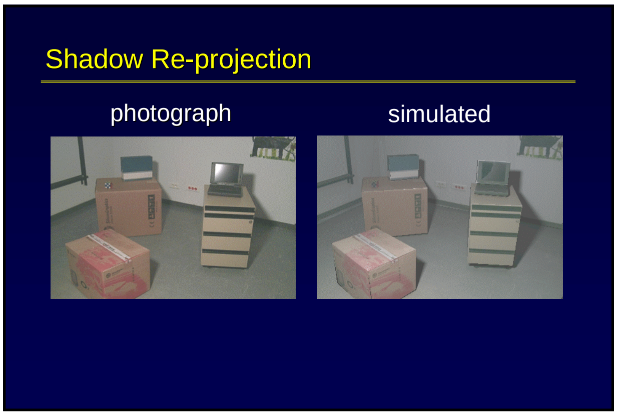
reflectance *confidence*

merged reflectance



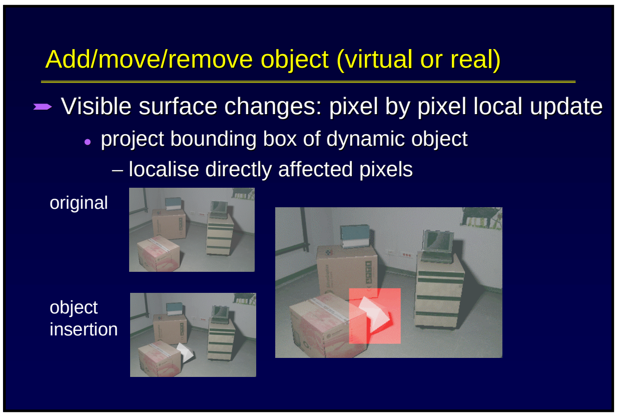
Interactive Modification: Shadow Reprojection

- Direct illumination: pixel by pixel
- Indirect illumination: optimised radiosity solution



Shadow Re-projection

photograph simulated

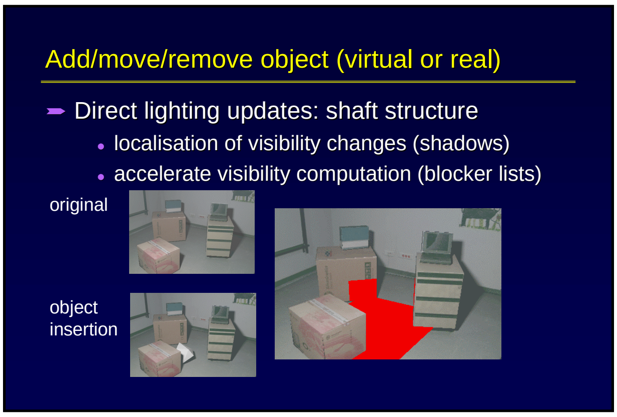


Add/move/remove object (virtual or real)

- Visible surface changes: pixel by pixel local update
  - project bounding box of dynamic object
    - localise directly affected pixels

original

object insertion

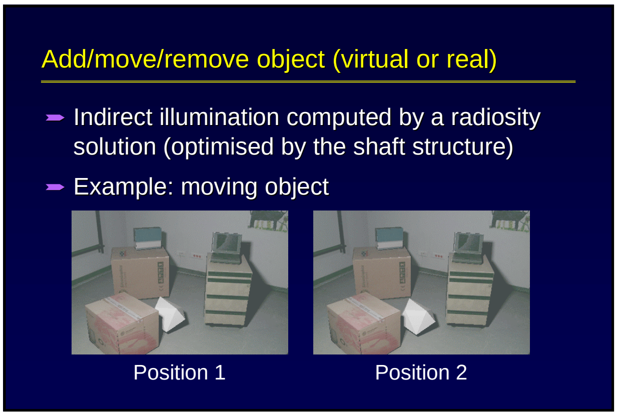


Add/move/remove object (virtual or real)

- Direct lighting updates: shaft structure
  - localisation of visibility changes (shadows)
  - accelerate visibility computation (blocker lists)

original

object insertion

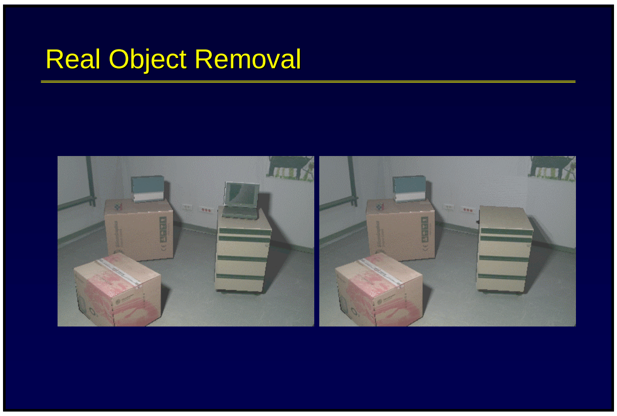


Add/move/remove object (virtual or real)

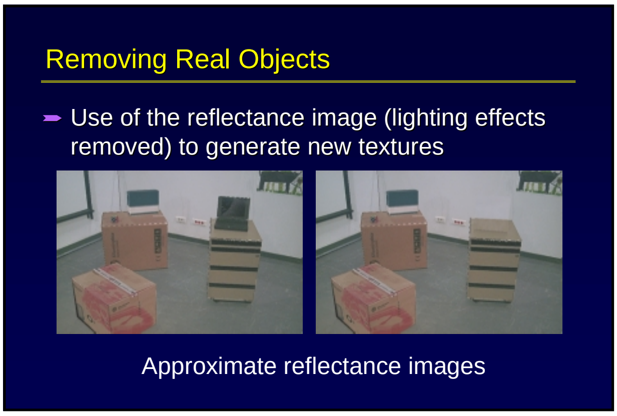
- Indirect illumination computed by a radiosity solution (optimised by the shaft structure)
- Example: moving object

Position 1

Position 2



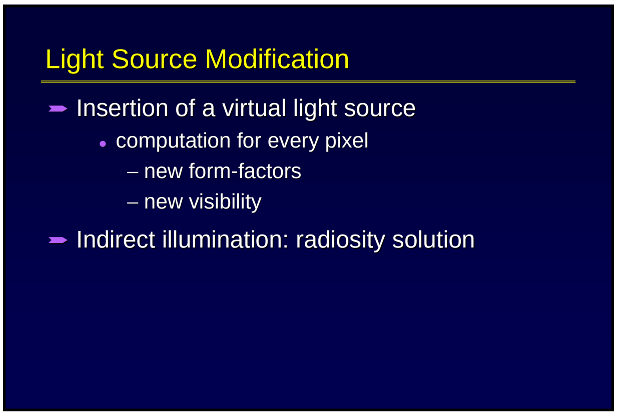
Real Object Removal



Removing Real Objects

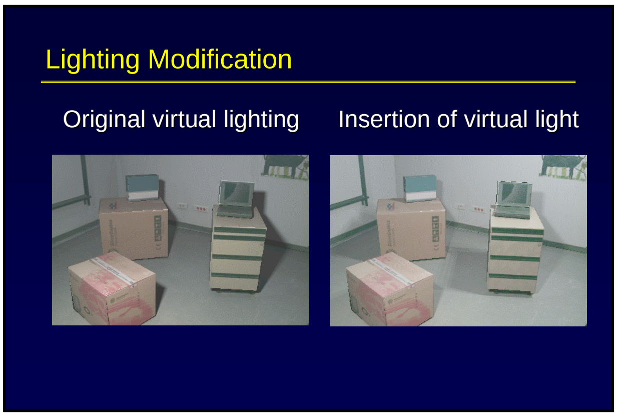
- Use of the reflectance image (lighting effects removed) to generate new textures

Approximate reflectance images



Light Source Modification

- Insertion of a virtual light source
  - computation for every pixel
    - new form-factors
    - new visibility
- Indirect illumination: radiosity solution



Lighting Modification

Original virtual lighting

Insertion of virtual light

Video

Discussion

- Capture process is still complicated and too long
  - need for many pictures, different lighting conditions etc.
- Some level of reflectance estimation is required even for interactive display
- Interactivity gives some leeway for lower quality

Shadow removal vs. Reflectance estimation ?

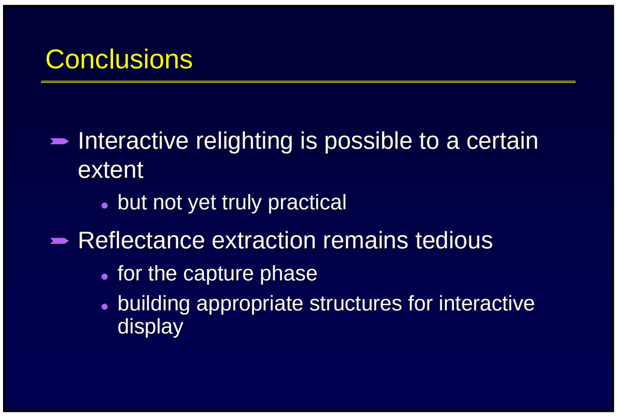
- Removing shadows may be enough
  - radiosity-based approaches cumbersome
- Reflectance estimation more stable with multiple light positions
  - can extract reasonable reflectance

Interactive display

- Choice between a mesh and a pixel (“deep framebuffer”) based approach
  - mesh is faster for higher resolution but with lower quality shadows
  - pixel-based approach is more accurate, allows extraction of reflectance

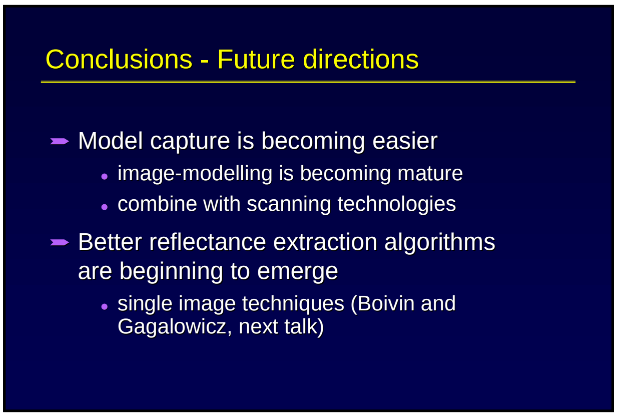
Interactive display

- Alternative: high resolution reflectance textures
  - *[Yu et al. ‘99]*
  - Capture process can be complex and cumbersome
- Hardware assisted shadows
  - *[Gibson and Murta ‘00]*



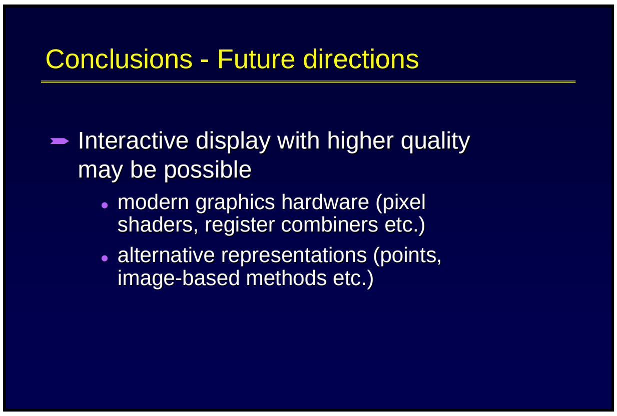
Conclusions

- Interactive relighting is possible to a certain extent
  - but not yet truly practical
- Reflectance extraction remains tedious
  - for the capture phase
  - building appropriate structures for interactive display



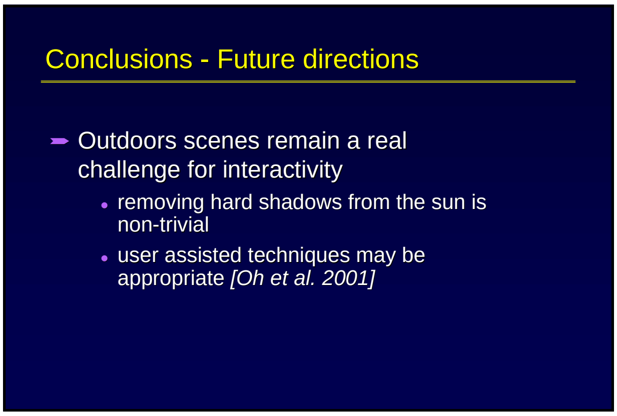
Conclusions - Future directions

- Model capture is becoming easier
  - image-modelling is becoming mature
  - combine with scanning technologies
- Better reflectance extraction algorithms are beginning to emerge
  - single image techniques (Boivin and Gagalowicz, next talk)



Conclusions - Future directions

- Interactive display with higher quality may be possible
  - modern graphics hardware (pixel shaders, register combiners etc.)
  - alternative representations (points, image-based methods etc.)



Conclusions - Future directions

- Outdoors scenes remain a real challenge for interactivity
  - removing hard shadows from the sun is non-trivial
  - user assisted techniques may be appropriate *[Oh et al. 2001]*

Fitting Complex Materials From A Single Image

Samuel Boivin[*]

University of Toronto
Department of Computer Science
Dynamic Graphics Project (DGP)

1 Introduction

1.1 Role of these course notes

These notes explain in detail material that do not appear in the original paper [3]. Therefore, even if the reader will find some overlap with the paper, it is fundamental to read the original paper [3] to understand the following notes.

1.2 Overview of the problem

Since its origin, Computer Graphics has aimed to simulate an illusion of reality. Rendering algorithms have been developed specifically to generate near-perfect images under realistic illumination conditions. It is often difficult to say if such images are realistic or not because there is no real reference such as a photograph. Moreover, the application may need to create novel viewpoints and/or novel illumination conditions from a sparse set of photographs. This is difficult to achieve without using image-based modeling and rendering algorithms. For example, suppose we want to insert a new synthetic object on top of a real anisotropic mirror inside a real scene. This operation clearly needs to take into account the interaction between the new object and its environment (especially this mirror). This is impossible to do, if we do not have an approximation of the reflectance properties of the real surfaces in the image. Therefore specific algorithms are necessary to recover these reflectance properties from the real images.

Many authors have contributed to the resolution of this problem [13, 16, 24, 23, 25, 17, 18, 28, 7, 29, 14, 15, 22, 21, 10, 9, 20]. The algorithms that they have produced vary greatly and not all can be re-used for our applications. Considerable work has been done for the reflectance estimation of an isolated object in particular illumination conditions [13, 16, 24, 23, 25, 17, 18] . Although these techniques often bring very detailed reflectance information (i.e. a full BRDF in some cases), their goal is more to replace the use of an expensive gonioreflectometer rather than to be able to change the viewpoint and/or the illumination. Recently, several methods have been developed to take into account the interaction between objects inside a real scene, from a sparse set of photographs [7, 29, 14, 15]. Fournier [10] proposed a different approach but with the use of a single image. However, his technique was limited to perfect diffuse environments and was not able to take into account specular surfaces. Our method has the similar ambition to recover an approximation of the BRDF of the surfaces from a single image, including the processing of specular, isotropic or anisotropic surfaces. This is extremely difficult to achieve because it is not possible to compute a full BRDF correctly without having several images, except for trivial cases

We propose a hierarchical and iterative technique that computes the best possible approximation of a real image, using the error computed between the rerendered image and the real one. Each of the new images is generated by making more and more complex assumptions about the reflectance properties of the real surfaces. It is rendered by a global illumination software that takes into account these reflectance changes. The main advantages of our approach are: it does not need any special device to capture the real image (a classical camera is enough), and it estimates the reflectances of all types of surfaces (including anisotropic mirrors) from a single image without any particular constraint

[*] email: boivin@dgp.toronto.edu

for the viewpoint position, the light sources[1] or the orientation of the objects . The goal of our method is to recover an approximation of the BRDF of the surfaces, and to compute the best synthetic image preserving the real properties of the scene (a real mirror has to be simulated as a specular surface and not as a textured surface for example).

2 Background and Related Work

All the techniques and ideas in this paper have been made possible by works about photorealistic rendering including global illumination and ray tracing, image-based modeling and BRDF modeling. However, the most relevant domains deal with *inverse rendering*, *image-based rendering* and *reflectance recovery*. We can split the *reflectance recovery* algorithms into three groups: direct measure of reflectances on the object using a specific device [26, 12, 1, 6], the extraction of reflectances from a set of images [13, 16, 24, 23, 25, 17, 18, 28, 7, 29, 14, 15], and the extraction of reflectances from a single image [22, 21, 10, 9, 20]. The last two parts may be subdivided into two categories, depending on whether the method takes into account energetic interreflections (using a global illumination algorithm for example) or not. A complete overview of these techniques can be found in our paper [3]. We only include here related techniques that use a single image and global illumination for reflectance recovery.

A pioneering work in reflectance recovery from a single image and using global illumination was completed by Fournier et al. [10] in 1993. He proposed to rerender an original image using a 3D representation of the scene (including the positions of the light source and the camera parameters) and a single image of this scene. All the surfaces are considered as perfect diffuse, and they used their reprojection on the real image to estimate their reflectances. A radiosity-based algorithm then computes an image applying these reflectances to a progressive radiosity technique [5] to obtain a new synthetic image.

An extension of the previous method was developed by Drettakis et al. [9]. They proposed an interactive version of the initial paper and added a vision algorithm for the camera calibration and the automatic positioning of the 3D geometrical model. They described a slightly different technique for the estimation of the reflectances of the surfaces and they used a hierarchical radiosity algorithm [11] to compute a new synthetic image close to the real one.

An approach similar to Fournier et al.'s was chosen by Gagalowicz [20]. It included a feedback that compares the real image to the synthetic one. He described a technique to generate a new synthetic image from a single one (except the 3D geometrical model built from two stereo images) using an iterative method that minimizes the error between the real image and the synthetic one. However, this technique is limited to a pure Lambertian approximation of the surface reflectances.

3 Elements of Reflectance Recovery

3.1 The Notion of Group

In our paper [3], we describe a new concept that we called *groups* to solve the problem of not directly seen objects. When we only have one single image, it is very common that several objects are not directly visible in the image from the original viewpoint, or their projection area is too small to be usable for reflectance analysis. It is then almost impossible to compute their reflectance because no information is available in the image. Therefore, every object in the scene is described as a part of a *group* of objects. In this group, at least one object is directly visible in the image and it is assumed to exactly have the same reflectance as the other objects of its group. So, when the reflectance for this object (and/or all other directly seen objects) has been computed, it is propagated to all other objects of the same group.

3.2 Reflectance Model and Data Description

In this technique, we used the Ward's BRDF model [26] for several reasons. First of all, it is a very simple model, and it gives you the opportunity to generate anisotropic surfaces. Secondly, is has been validated on real materials using an experimental device to compute the parameters used in this model. However, it is possible to use any other BRDF model if you can determine a hierarchy of BRDFs for the implementation: the hierarchy should be built following the

[1] In fact, the emittances of the light sources are supposed to be known. However, if it is not the case then there is a method that can recover them automatically as proposed by Fournier et al. [10].

number and the complexity of the parameters used in the model. For example, if you look at the Ward's BRDF model and then at our hierarchy, you see that the hierarchy has been built following theses rules (perfectly diffuse: *natural* case with one parameter, perfectly specular: simplest case with one parameter, non-perfectly specular: enhanced specular case with one iteratively computed parameter, ..., isotropic: three parameters, ...).

A 3D geometrical model is **indispensable** to our technique. It does not matter how this 3D model has been obtained, but its projection on the image must match the scene, and the data must be usable by your rendering algorithm[2]. Some proposals to obtain such models are given in [8] for example. The technique that we used is described in the paper.

3.3 Accuracy of the geometrical model and extraction of the pixel intensities

Since we have the geometrical model, the camera parameters and the original image, we are able to compute *index buffers* using a famous technique called *offscreen rendering* [19]. The index buffers produce a new image that gives the number of the group for every pixel it belongs to. So, when we compute the reflectance we just traverse the index buffer and the original image to sum these pixel intensities [3] The precision required by the inverse algorithm for the positioning of the geometrical model tolerates several pixels of difference between the projection of the model and the real objects in the image. The acceptable number of misclassified pixels depends on the size of the projected object in the original image. For example, if the projection of all objects belonging to the same group has a total number of ten visible pixels, then the inverse algorithm will compute the wrong BRDF when at least about three or four of the ten pixels do not belong to the currently analyzed objects. We use very classical filtering methods, such as edge detectors, edge removal filters and a planar approximation, to reduce inconsistencies with the geometrical model by minimizing the number of pixels assigned to a wrong object.

4 Inverse Rendering from a Single Image

4.1 Overview of the Algorithm

Here, we want to remind you of the main characteristic of our inverse rendering algorithm. The following paragraph is a part of our SIGGRAPH 2001 paper.

The core of our technique is incremental and hierarchical (see figure 1). It is incremental because the surface reflectances are going to evolve to their optimum value. It is hierarchical because the general algorithm forces the surfaces BRDF to be more and more complex if the error between the real and the synthetic image does not decrease for these surfaces. This algorithm is iterative and will proceed to successive corrections of the surface reflectances by minimizing the error between the real and the synthetic image. Indeed, each computed error for a group of objects having the same photometric properties drives the correction of their reflectance. Our technique successively applies the selected assumption on the group reflectances until the error becomes smaller than a user-defined threshold. The notion of threshold and how to fix its value is discussed in [3, 2]

We start the algorithm with the *perfect diffuse* case without considering texture (the diffuse reflectance of a group is computed by averaging the radiances covered by its projection in the real image). All the surfaces are then considered as perfectly Lambertian, and the rendering software (*Phoenix* in this case[4]) computes a new approximation of the image. If the difference between the real and the synthetic image for a group is greater than a fixed threshold on all the group projection, then the reflectance of this group is considered as *perfectly specular* for the next rerendering iteration. If, after *Phoenix* has recomputed a new image using the new assumption, the error for this group remains big, then its reflectance is simulated as *non-perfectly specular.* We apply the same principle again to change the group reflectance to a *diffuse and specular* one. Until then, all the surfaces were considered with no roughness term (only a ρ_d and a ρ_s were estimated). In the next assumption, if the difference between the two images still produces big

[2] Remember that you must have a rendering software to produce a photorealistic image. This image is supposed to be compared to the original one. So, the choice of the technique is *critical* to our inverse rendering algorithm because on one hand it must provide a **photorealistic** image, and on the other hand this image must be computed on a **reasonably short time** (less than a few minutes).

[3] In fact, the pixel intensities are first converted into radiances. See [3, 2] to understand why and how.

[4] It is possible to use any other global illumination rendering software, such as *Radiance* for example.

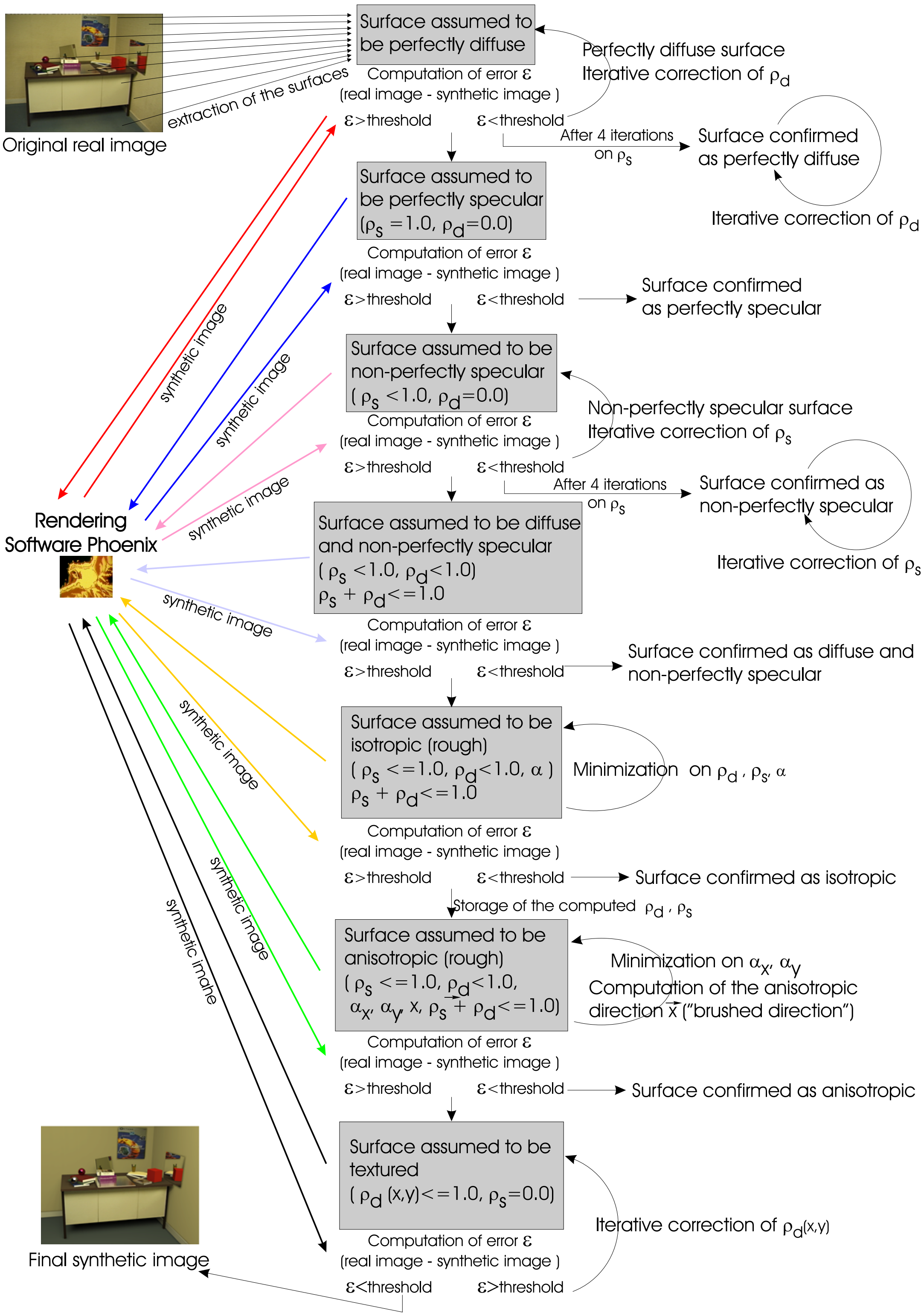

Original real image
extraction of the surfaces
Surface assumed to be perfectly diffuse
Computation of error ε
(real image - synthetic image)
ε>threshold
ε<threshold
Perfectly diffuse surface
Iterative correction of ρ_d
After 4 iterations on ρ_s
Surface confirmed as perfectly diffuse
Iterative correction of ρ_d
Surface assumed to be perfectly specular (ρ_s =1.0, ρ_d=0.0)
Computation of error ε
(real image - synthetic image)
ε>threshold
ε<threshold
Surface confirmed as perfectly specular
Surface assumed to be non-perfectly specular (ρ_s <1.0, ρ_d=0.0)
Computation of error ε
(real image - synthetic image)
ε>threshold
ε<threshold
Non-perfectly specular surface
Iterative correction of ρ_s
After 4 iterations on ρ_s
Surface confirmed as non-perfectly specular
Iterative correction of ρ_s
Rendering Software Phoenix
synthetic image
synthetic image
synthetic image
Surface assumed to be diffuse and non-perfectly specular (ρ_s <1.0, ρ_d<1.0) ρ_s + ρ_d<=1.0
synthetic image
Computation of error ε
(real image - synthetic image)
ε>threshold
ε<threshold
Surface confirmed as diffuse and non-perfectly specular
synthetic image
Surface assumed to be isotropic (rough) (ρ_s <=1.0, ρ_d<1.0, α) ρ_s + ρ_d<=1.0
Minimization on ρ_d , ρ_s, α
Computation of error ε
(real image - synthetic image)
ε>threshold
ε<threshold
Surface confirmed as isotropic
Storage of the computed ρ_d , ρ_s
synthetic image
synthetic imahe
Surface assumed to be anisotropic (rough) (ρ_s <=1.0, ρ_d<1.0, α_x, α_y, $\vec{x}$, ρ_s + ρ_d<=1.0)
Minimization on α_x, α_y
Computation of the anisotropic direction $\vec{x}$ ("brushed direction")
Computation of error ε
(real image - synthetic image)
ε>threshold
ε<threshold
Surface confirmed as anisotropic
Surface assumed to be textured (ρ_d (x,y)<=1.0, ρ_s=0.0)
Iterative correction of $\rho_{d(x,y)}$
Computation of error ε
(real image - synthetic image)
ε<threshold
ε>threshold
Final synthetic image

Figure 1: General iterative and hierarchical algorithm for reflectance recovery. Each surface of the scene is analyzed separately, depending on the assumption about its reflectance (perfectly diffuse, perfectly specular, etc.). If the assumption is false (the error between the real and the synthetic image is large), then the surface reflectance is assumed to be more complex (hierarchical principle). If the assumption is correct then the surface reflectance is modified accordingly in order to minimize the error between the two images (iterative principle). During each global rerendering iteration, the reflectances of all surfaces are then continuously updated, to take into account the incident energy coming from any surface for which the BRDF has changed (a diffuse surface which became *perfectly specular* for example).

errors, they are considered as isotropic and a roughness factor (α) has to be evaluated. This assumption is extended to anisotropic properties if the user-defined threshold for the error has not been reached. If all assumptions have failed, the group is supposed to be highly textured. Since only a single image is available, it is extremely difficult and sometimes impossible to create a combination between this texture and other reflectance properties (a glossy textured surface for example). This situation is discussed in paragraph 5.2.

5 Implementation details

We present here details regarding particular cases and implementation issues.

5.1 The problem of multiple specular reflections

This is a very common problem in our technique and no details have been given in our paper on the method to solve it. Let us consider the following case: given a real image containing two non-perfect mirrors (A and B) facing each other. So, in the view of A in the original image, B is seen through the specular reflection and vice-versa. When our algorithm attempts to compute the BRDF of A, it is biased by the BRDF of B and vice-versa. You could think that this is not a very common case, but it is. Due to the hierarchical principle of our technique, this case is very common. Indeed, every time that the algorithm needs to go further than the perfectly specular case, it tries the non-perfectly specular case. So when you have two textured books facing each other, they will be considered as non-perfectly specular at the same time in the hierarchy. Another problem is the case where a textured book (A) has a part reflecting in a specular object (B). In this case, when the hierarchical algorithm decide that both of them are being assumed as specular surfaces, the problem is that the part of A reflecting in B prevents our algorithm of finding the properties of B: all the pixels of the projection of A reflecting in B bias the reflectance computation of B.

Of course, our algorithm takes into account all these cases by doing a simple processing. When the algorithm detects a surface as specular (let us call it now A), it first analyzes the content of the indirectly seen object (let say for simplicity purposes that there is only one object seen through the mirror, and let us call it B). We compute an index buffer of the objects seen through the mirror. For each of these objects, we look if their reflectance is being analyzed or if it has already been *confirmed* (see figure 1). If it has already been confirmed, there is no problem and its reflectance will not bias the specular surface. If it is being analyzed then our algorithm proceeds as follows: the object (B) seen through the specular surface is considered as textured. We directly extract its texture from the image regardless of its properties, and no illumination is taken into account for the next rendering step on this *pseudo-textured* object. Now, when the algorithm looks at the properties of the mirror (A), the part of the object (B) seen through the mirror can not bias the reflectance of the mirror because it is considered as a texture. The inverse rendering algorithm then computes the properties of the specular surface (A). The error between the real and the synthetic image determine if the specular assumption is true or not and this error is stored for later use. Then, the algorithm tries the opposite assumption: A is considered as textured and B is a specular object. A new error between the real and the synthetic image is computed and it is stored for later use. Finally, using the **previously stored specular properties** of A and B, the rendering algorithm computes a new synthetic image assuming that both A and B are specular objects. A new error is computed between the real and the synthetic image. The inverse rendering algorithm then compares the three computed errors and chooses the hypothesis which has the smallest one. However, if this error concerns a case where an object has been assumed textured, then **only the specular object** has its properties confirmed. The object assumed textured is still being furthermore investigated to compute its real properties (it could be an isotropic one for example).

5.2 Textured surfaces

When the simulation of a surface as anisotropic still produces big errors in the difference image, we proceed to texture extraction.

Extracting the texture from the real image is an easy task that can be realized using the technique proposed in [27] for example (see figure 2). However, we have to extract this texture while taking into account the fact that it already has received the energy from the light sources, and that the pixels covered by its projection in the real image contain this information. Otherwise, if we send the energy of the light sources to these textures again, they will be over-illuminated. Therefore, we introduce here a notion called *radiosity texture* that balances the extracted texture with an intermediate texture in order to minimize the error between the real and the synthetic image. As for the perfectly diffuse reflectance case, this intermediate texture is computed by an iterative method (see algorithms 1 and 2). All fully detailed algorithms can be found in [2].

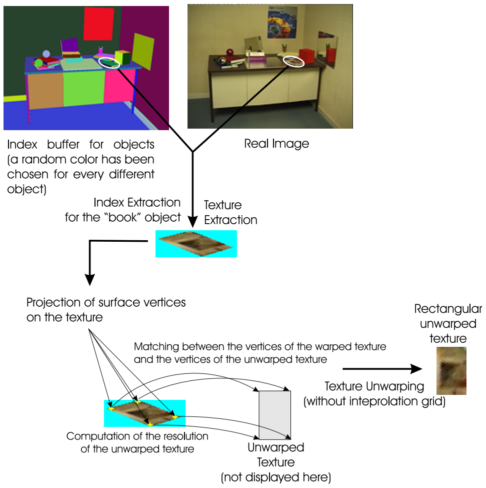

Index buffer for objects (a random color has been chosen for every different object)
Real Image
Index Extraction for the "book" object
Texture Extraction
Projection of surface vertices on the texture
Rectangular unwarped texture
Matching between the vertices of the warped texture and the vertices of the unwarped texture
Texture Unwarping (without inteprolation grid)
Computation of the resolution of the unwarped texture
Unwarped Texture (not displayed here)

Figure 2: Extracting and Unwarping Textures

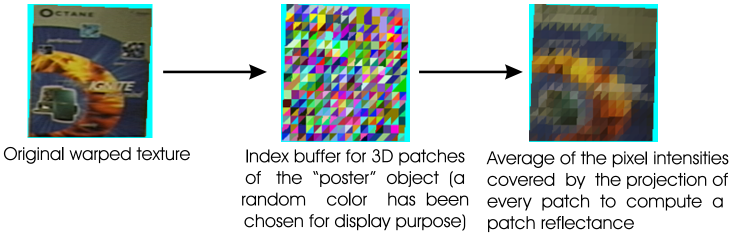

Original warped texture
Index buffer for 3D patches of the "poster" object (a random color has been chosen for display purpose)
Average of the pixel intensities covered by the projection of every patch to compute a patch reflectance

Figure 3: Computation of the diffuse reflectances for a texture in a real image. Here, the subdivision level has been voluntarily increased for clarity purposes.

At the first iteration, the texture used to rerender the image is the texture directly extracted from the real image. At the second iteration, the texture used to obtain the resulting synthetic image is multiplied by the ratio between the newly extracted texture of this synthetic image and the texture of the real image (see algorithm 2). This iterative process stops when the user-defined threshold for textured surfaces has been reached. The textures of the poster and the books in the rerendered images of section 6.2 have been obtained using this technique. The problem of this method is that it computes a texture including the shadows, the specular reflections and the highlights. Typically, suppose that we have a marbled floor on which a sphere is reflected. The texture of this floor in the real image then includes the marble characteristics, its reflectance properties and the sphere reflection including its own reflectance properties. How does one extract the marble characteristics only and independently of the rest of the scene ? This is an extremely hard problem, and according to Y.Sato et al. [25] no algorithm has been proposed yet to solve it using a single image.

6 Results

6.1 Experimental Validation on a synthetic scene

In this section, we present the values obtained for the recovered BRDF of a computer-generated scene (see left image of Figure 5). We compare them to the original known values used to render the original image with *Phoenix* [3, 2].

The first level of the hierarchy in the inverse rendering process computes all the parameters of the surfaces in a straightforward manner. However, the error remains large for the floor and the next levels are tested for this object.

Algorithm 1 Function replacing the anisotropic properties by the texture properties

```
Function Change_Anisotropic_Group_By_Texture(group)
// Is the error for the group bigger than a user-defined threshold ?
if group.$\varepsilon$ $\geq$ anisotropic threshold then
    group.type = texture;
    // The surface is now considered as a textured surface. We extract the object textures
    // and we unwarp them to make them usable by the following function.
    for all objects $j$ do
        Compute_Texture(group.object[$j$], real_img);
    end for
end if
```

Algorithm 2 Pseudo-Algorithm for the computation and the iterative correction of texture reflectances

```
Function Compute_Texture(object, real_img)
 The patch reflectances are computed using the warped original texture
 otherwise the unwarped texture could bias the computation of $\rho_d$
object−>texture = texture extracted from real image using index buffer;
for all patchs $k$ do
    Compute the average diffuse reflectance of patch k (see figure 3) using offscreen-rendering
end for
// Texture unwarping for final rendering
Compute vertex coordinates for all object facets projected on the image;
Unwarp object−>texture using previous coordinates to generate a rectangular image (see figure 2);

Function Modify_Texture(object, real_img, syn_img)
// Computation of the error image
tmp_texture = texture extracted from the the synthetic image;
texture_$\varepsilon$ = $\frac{\text{texture(object,real\_img)}}{\text{tmp\_texture}}$;
// Compute new texture
new_texture = object−> texture $\times$ texture_$\varepsilon$;
// Computation of patch reflectances for the radiosity equation (see [4] for more details)
for all patchs $k$ projecting on new_text do
    Compute $\rho_{d_k}$ as the average of the pixel intensities
    covered by the projection of $k$;
end for
// Unwarping of the new texture for the next rendering
Compute vertex coordinates for all object facets projected on the image;
Unwarp new_texture using previous coordinates to generate a rectangular image (see figure 2);
```

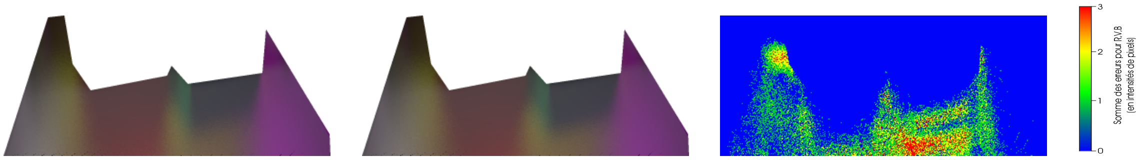


Figure 4: From left to right: original anisotropic floor, floor simulated as an isotropic object, and the error image between the original and the rerendered images.

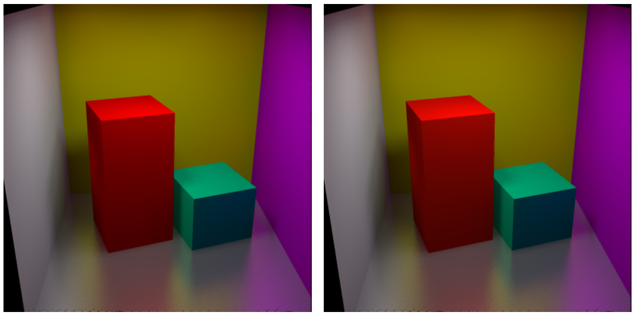


Figure 5: Left: the original computer-generated image. Right: the new synthetic image produced by our inverse rendering technique.

The specular assumptions (perfectly specular, non-perfectly specular, both diffuse and specular) produced large errors forcing the algorithm to choose the isotropy hypothesis. During the isotropy case, a global minimum has been found for ρ_d, ρ_s and α, and the synthetic image is visually very close to the original as shown by Figure 4. However, as we only set 1% for the maximum tolerated error to switch from the isotropy hypothesis to the anisotropy, our method tries to simulate the floor as an anisotropic object.

Surface	Parameter	Real value	Computed value	Comment
Left wall	ρ_d	(0.66, 0.66, 0.66)	(0.65916, 0.66075, 0.66037)	
	ρ_s	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)	
Right wall	ρ_d	(0.69, 0, 0.95)	(0.69002, 0.0, 0.95901)	
	ρ_s	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)	
Back wall	ρ_d	(0.65, 0.65, 0.0)	(0.64997, 0.65067, $4 \cdot 10^{-7}$)	
	ρ_s	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)	
Ceiling	ρ_d	(1.0, 1.0, 1.0)	(1.0, 1.0, 1.0)	See footnote [5].
	ρ_s	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)	
Big block	ρ_d	(0.77, 0.0, 0.0)	(0.77002, $2 \cdot 10^{-4}$, $3 \cdot 10^{-6}$)	
	ρ_s	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)	
Small block	ρ_d	(0.0, 0.76, 0.26)	(0.0, 0.75802, 0.25912)	
	ρ_s	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)	
Floor	ρ_d	(0.1, 0.1, 0.1)	(0.10013, 0.10045, 0.09981)	The total error for θ is 0.77%.
	ρ_s	(0.9, 0.9, 0.9)	(0.89909, 0.90102, 0.89903)	
	θ	0.0^o	2.8^o	
	α_x	0.07	0.06999	
	α_y	0.11	0.1101	

Figure 6: Comparison between the recovered reflectance parameters and their original values.

[5] The ceiling is not directly visible in the original image. When this happens, the algorithm considered this object as a perfect diffuse white object. In practice, if such a case happens, the user should find an object whose photometric properties are close to the ceiling. The ceiling will then be declared in the same group as this object.

6.2 Inverse Rendering

All the following synthetic images have been generated using *Phoenix* as rendering and inverse rendering software. The first synthetic image at the top right of figure 8 has been generated in 37 minutes using the hierarchical algorithm, from the left real photograph. Two specular surfaces have been recovered and simulated as non-perfect mirrors. Neither isotropic nor anisotropic hypotheses have been done thanks to the optimization technique described in our paper [3], and 14 rerendering iterations were necessary to generate the final image.

The inverse algorithm took 4 hours and 40 minutes to produce the image at the bottom right of figure 8. Roughly 4 hours of this time were necessary to recover the anisotropic BRDF of the aluminum surface. The final rendering stage took 32 minutes to render the final image (100 bounced rays have been used for the anisotropic surface).

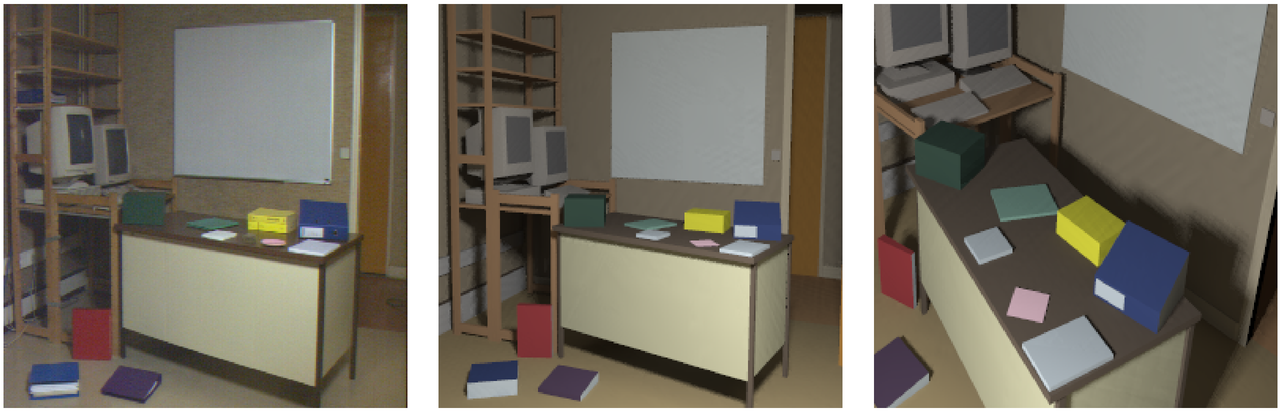


Figure 7: Example of a pure diffuse approximation of a whole 3D scene. From left to right: the original image captured with a camera, the synthetic image and a synthetic image generated under a new viewpoint. The perfectly diffuse assumption is realistic enough for many surfaces, except the computer monitor and the door. Moreover, even if the desk is a real anisotropic surface, a pure diffuse approximation produces a realistic enough result for this object. Note that a book on the left bookshelf has not been modeled. Due to the filtering step and the principle of the method, this does not disturb the inverse rendering case. However, this remains true only for small objects that do not interact much with the real environment. A “very“ large error in the modeling step would definitely produce wrong results.

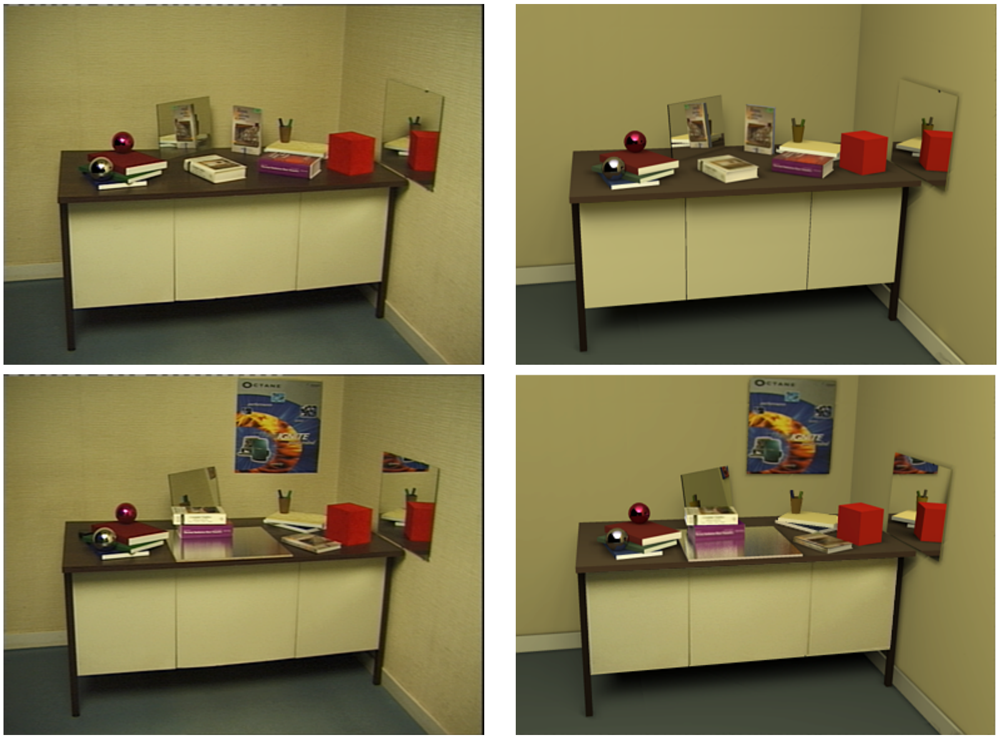


Figure 8: Two different examples of synthetic images (right) rerendered from a single real image (left). We remark that the perfectly diffuse assumption is realistic enough for many surfaces (including the walls, the floor, the desk, etc.).

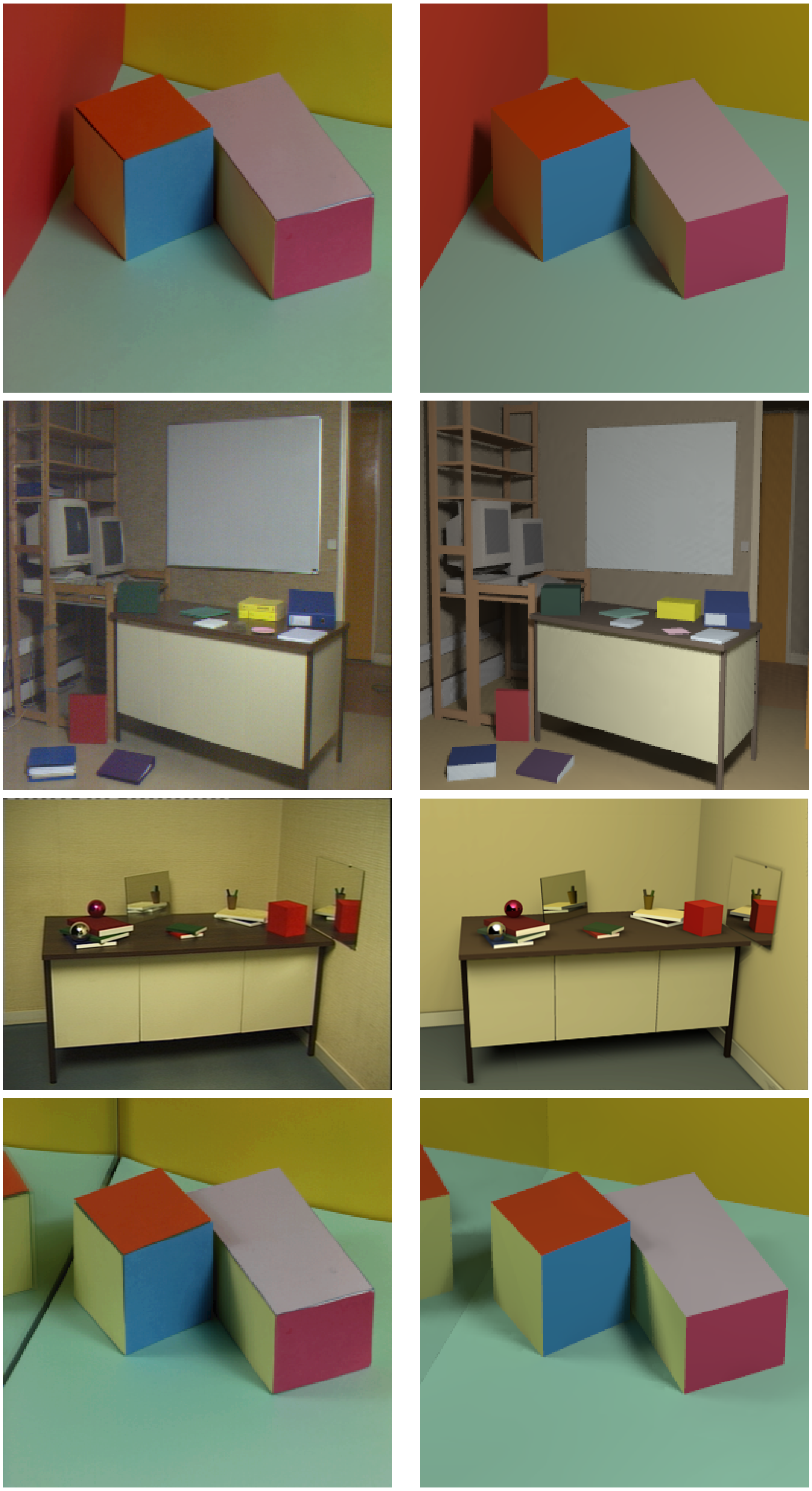


Figure 9: Some examples of rerendered scenes (right column) including perfectly diffuse and specular surfaces compared to the original image (left column)

7 Augmented Reality Applications

The images of figure 10 show examples of applications in augmented reality. Some synthetic objects have been added such as a small robot and a luxo-like desk lamp. It is also possible to modify the reflectances easily too. New viewpoints can be generated and new illumination conditions can be created as well.

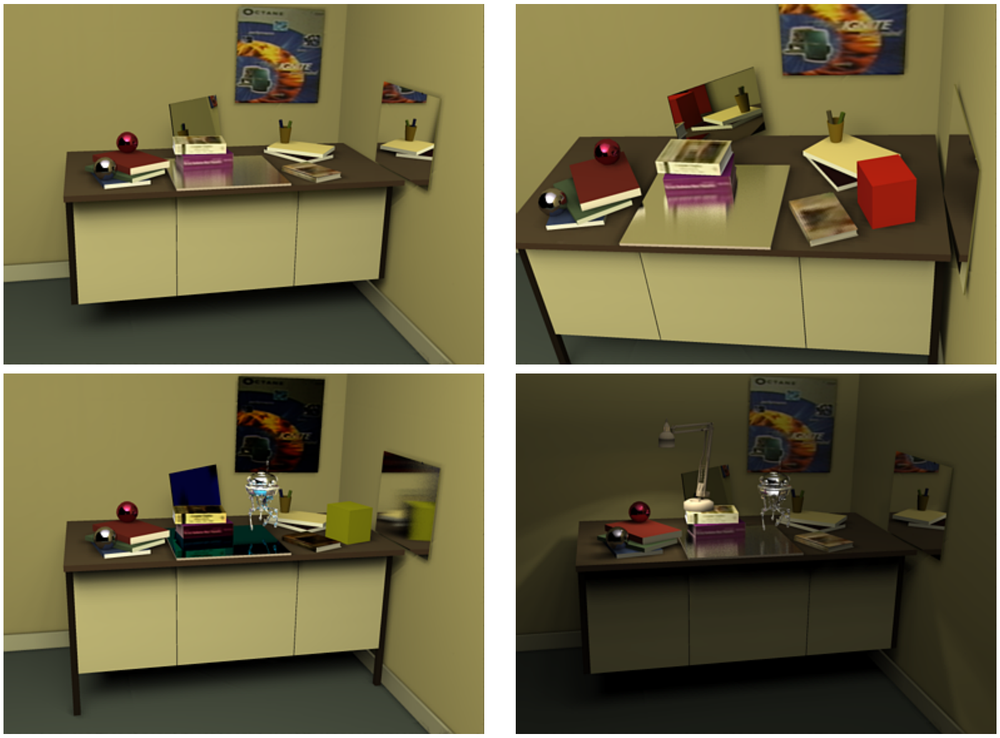


Figure 10: Examples of several augmented reality applications. All these new images were rendered using our global illumination software *Phoenix*, which first recovered the surface reflectances from the bottom left image of figure 8. The top left image shows the original scene removing some objects (the feet of the desk and the red cube). Note that the right mirror has taken into account the modification. The right top image shows the original scene rendered under a novel viewpoint. The bottom left image shows the scene with modified photometric properties, and the addition of an object (a small robot). The bottom right image presents the scene under novel illumination conditions with the addition and deletion of objects.

8 Conclusion and Future Work

In this course, we have presented a new technique that determine an approximation of the reflectance properties of the surfaces of a 3D scene [3, 2]. An incremental and hierarchical algorithm iteratively estimates the diffuse, specular, isotropic and anisotropic reflectance parameters. In a final step, the textured surfaces are considered as a special case of reflectances to be simulated. The method takes as input a single photograph of the scene taken under known illumination conditions as well as a 3D geometric model of the scene. The result is a complete description of the photometric properties of the scene which may be used to produce a photorealistic synthetic image very similar to the real one. We showed that the method is robust and allows for the visualization the original scene from any new angle, with any illumination conditions and with the addition, removal and modification of objects.

Our work has currently some limitations, especially regarding textured surfaces. Until now, we are not able to discriminate the shadows or highlights from an assumed textured surface. In this regard, it will be interesting to extend our method to these cases, although we think that this is a very difficult problem, if one sticks to the single image assumption.

While many challenges remain, we believe that algorithms for recovering an approximation of the reflectances inside a real scene are an important direction of research for both the Computer Vision and the Computer Graphics communities. In Computer Vision, it could be possible for example to use our method to enhance the positioning of mirrors using a minimization algorithm between the real and the synthetic image. Regarding Computer Graphics, we may extend the reflectance recovery algorithm to objects that have more complex photometric properties such as light beam, small fires, caustics, etc. The hierarchical property of our technique offers many possible extensions.

Aknowledgments

The author would like to thank Glenn Tsang for his helpful suggestions regarding the writing of these course notes.

References

[1] R. Baribeau, M. Rioux, and G. Godin. Color reflectance modeling using a polychromatic laser range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):263–269, February 1992.

[2] Samuel Boivin. *Simulation Photoréaliste de Scènes d'Intérieur à partir d'Images Réelles*. PhD thesis, Spécialité Informatique, École Polytechnique, Palaiseau, January 2001.

[3] Samuel Boivin and André Gagalowicz. Image-based rendering of diffuse, specular and glossy surfaces from a single image. *Proceedings of SIGGRAPH 2001*, pages 107–116, August 2001. ISBN 1-58113-292-1.

[4] M. F. Cohen, D. P. Greenberg, D. S. Immel, and P. J. Brock. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications*, 6(3):26–35, March 1986.

[5] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 75–84, August 1988.

[6] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, January 1999. ISSN 0730-0301.

[7] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In Michael Cohen, editor, *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, pages 189–198. Addison Wesley, July 1998.

[8] Paul Ernest Debevec. *Modeling and Rendering Architecture from Photographs*. PhD thesis, University of California, Berkeley, 1996.

[9] George Drettakis, Luc Robert, and Sylvain Bougnoux. Interactive common illumination for computer augmented reality. In Julie Dorsey and Philipp Slusallek, editors, *Eurographics Rendering Workshop 1997*, pages 45–56. Springer Wien, June 1997.

[10] Alain Fournier, Atjeng S. Gunawan, and Chris Romanzin. Common illumination between real and computer generated scenes. In *Graphics Interface '93*, pages 254–262. Canadian Information Processing Society, May 1993. Held in Toronto, Ontario, Canada.

[11] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):197–206, July 1991.

[12] Konrad F. Karner, Heinz Mayer, and Michael Gervautz. An image based measurement system for anisotropic reflection. *Computer Graphics Forum*, 15(3):119–128, August 1996.

[13] G. Kay and T. Caelli. Inverting an illumination model from range and intensity maps. *CGVIP: Image Understanding*, 59:183–201, 1994.

[14] C. Loscos, M. C. Frasson, G. Drettakis, B. Walter, X. Grainer, and P. Poulin. Interactive virtual relighting and remodeling of real scenes. Available from www.imagis.imag.fr/Publications RT-0230, Institut National de Recherche en Informatique en Automatique (INRIA), Grenoble, France, April 1999.

[15] Celine Loscos, George Drettakis, and Luc Robert. Interactive virtual relighting of real scenes. *IEEE Transactions on Visualization and Computer Graphics*, 6(3):289–305, 2000.

[16] J. Lu and J. Little. Reflectance function estimation and shape recovery from image sequence of rotating object. In *International Conference on Computer Vision*, pages 80–86, June 1995.

[17] Stephen R. Marschner and Donald P. Greenberg. Inverse lighting for photography. In *Proceedings of the Fifth Color Imaging Conference*. Society for Imaging Science and Technology, November 1997.

[18] Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, Kenneth E. Torrance, and Donald P.Greenberg. Image-based brdf measurement including human skin. In Dani Lischinski and Greg Ward Larson, editors, *Eurographics Rendering Workshop 1999*. Eurographics, June 1999.

[19] J. Neider, T. Davis, and M. Woo. *OpenGL Programming Guide: The Official Guide to Learning OpenGL*. Addison Wesley, New York, 1995.

[20] A. Rosenblum. *Data Visualization*, chapter Modeling Complex indoor scenes using an analysis/synthesis framework (André Gagalowicz). Academic Press, 1994.

[21] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Illumination distribution from brightness in shadows: Adaptive extimation of illumination distribution with unknown reflectance properties in shadow regions. In *Proceedings of IEEE ICCC'99*, pages 875–882, September 1999.

[22] Kosuke Sato and Katsushi Ikeuchi. Determining reflectance properties of an object using range and brightness images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1139–1153, 1991.

[23] Yoichi Sato and Katsushi Ikeuchi. Temporal-color space analysis of reflection. *Journal of Optical Society of America*, 11(11):2990–3002, November 1994.

[24] Yoichi Sato and Katsushi Ikeuchi. Reflectance analysis for 3d computer graphics model generation. *Graphical Models and Image Processing*, 58(5):437–451, 1996.

[25] Yoichi Sato, Mark D. Wheeler, and Katsushi Ikeuchi. Object shape and reflectance modeling from observation. In Turner Whitted, editor, *Computer Graphics, Proceedings of SIGGRAPH 97*, pages 379–388. Addison Wesley, August 1997.

[26] Gregory J. Ward. Measuring and modeling anisotropic reflection. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 265–272. ACM Press, July 1992.

[27] George Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, 1990.

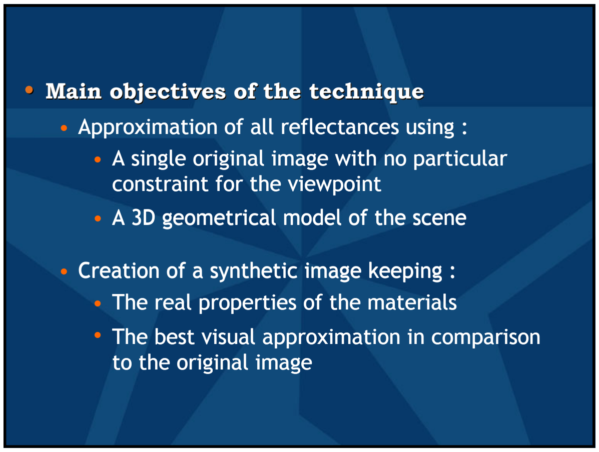
[28] Tien-Tsin Wong, Pheng-Ann Heng, Siu-Hang Or, and Wai-Yin Ng. Image-based rendering with controllable illumination. In Julie Dorsey and Phillip Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 13–22, New York,, 1997. Springer Wien. ISBN 3-211-83001-4.

[29] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse global illumination : Recovering reflectance models of real scenes from photographs. In A. Rockwood, editor, *Computer Graphics (SIGGRAPH '99 Proceedings)*, volume 19, pages 215–224. Addison Wesley Longman, August 1999.

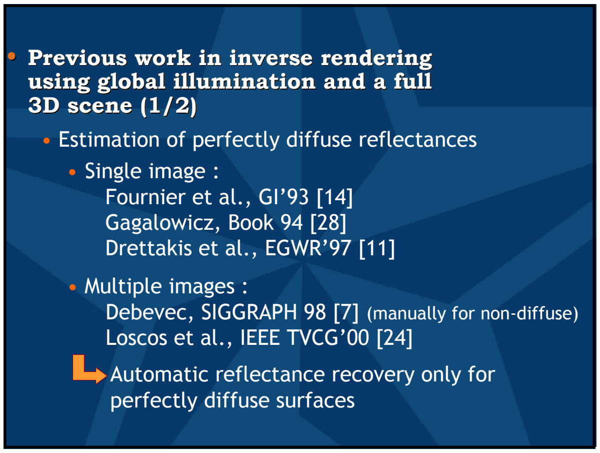
SAN ANTONIO
SIGGRAPH
2002

Fitting Complex Material Properties From A Single Image

Samuel Boivin
Dynamic Graphics Project
Department of Computer Science
University of Toronto

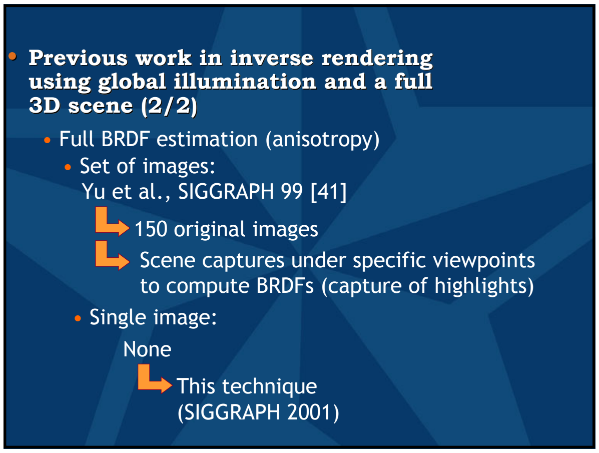


- **Main objectives of the technique**
  - Approximation of all reflectances using :
    - A single original image with no particular constraint for the viewpoint
    - A 3D geometrical model of the scene
  - Creation of a synthetic image keeping :
    - The real properties of the materials
    - The best visual approximation in comparison to the original image

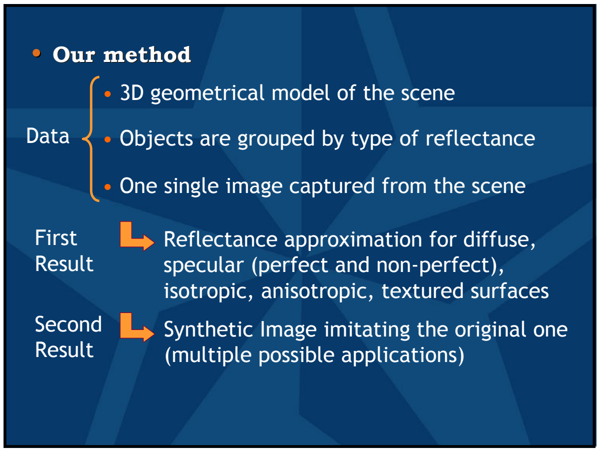


- **Previous work in inverse rendering using global illumination and a full 3D scene (1/2)**
  - Estimation of perfectly diffuse reflectances
    - Single image :
      Fournier et al., GI'93 [14]
      Gagalowicz, Book 94 [28]
      Drettakis et al., EGWR'97 [11]
    - Multiple images :
      Debevec, SIGGRAPH 98 [7] (manually for non-diffuse)
      Loscos et al., IEEE TVCG'00 [24]

      ↳ Automatic reflectance recovery only for perfectly diffuse surfaces



- **Previous work in inverse rendering using global illumination and a full 3D scene (2/2)**
  - Full BRDF estimation (anisotropy)
    - Set of images:
      Yu et al., SIGGRAPH 99 [41]
      ↳ 150 original images
      ↳ Scene captures under specific viewpoints to compute BRDFs (capture of highlights)
    - Single image:
      None
      ↳ This technique (SIGGRAPH 2001)

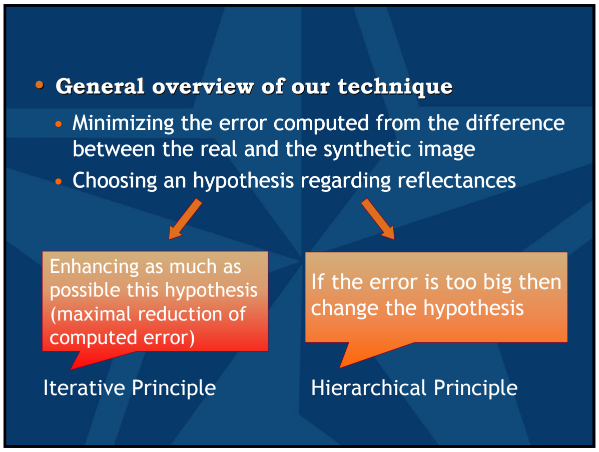


- **Our method**

Data
- 3D geometrical model of the scene
- Objects are grouped by type of reflectance
- One single image captured from the scene

First Result ↳ Reflectance approximation for diffuse, specular (perfect and non-perfect), isotropic, anisotropic, textured surfaces

Second Result ↳ Synthetic Image imitating the original one (multiple possible applications)



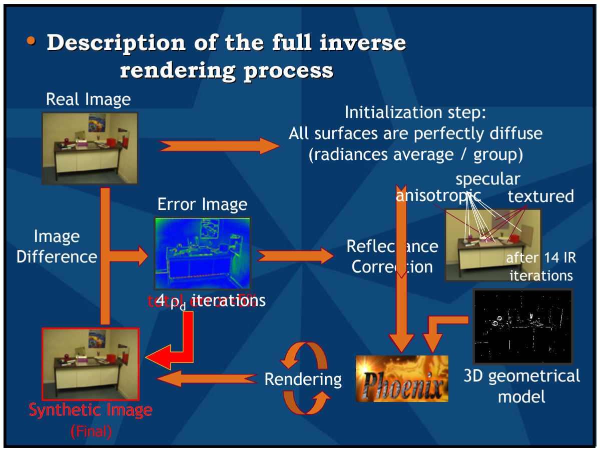
- **General overview of our technique**
  - Minimizing the error computed from the difference between the real and the synthetic image
  - Choosing an hypothesis regarding reflectances

Enhancing as much as possible this hypothesis (maximal reduction of computed error)

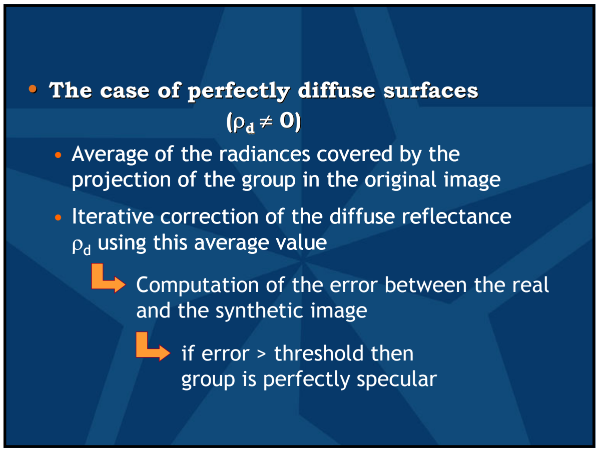
If the error is too big then change the hypothesis

Iterative Principle

Hierarchical Principle



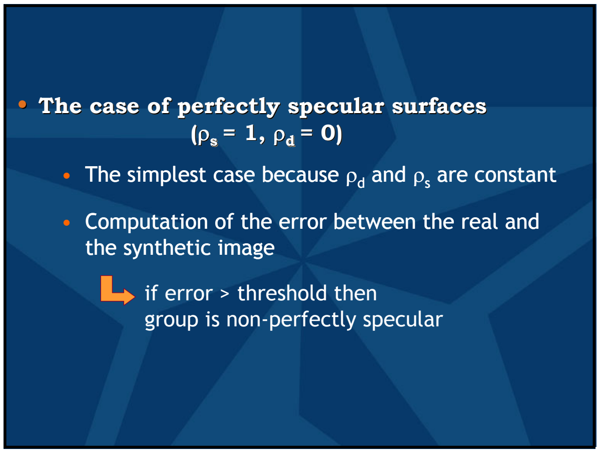
- **Description of the full inverse rendering process**



- **The case of perfectly diffuse surfaces ($\rho_d \neq 0$)**
  - Average of the radiances covered by the projection of the group in the original image
  - Iterative correction of the diffuse reflectance ρ_d using this average value

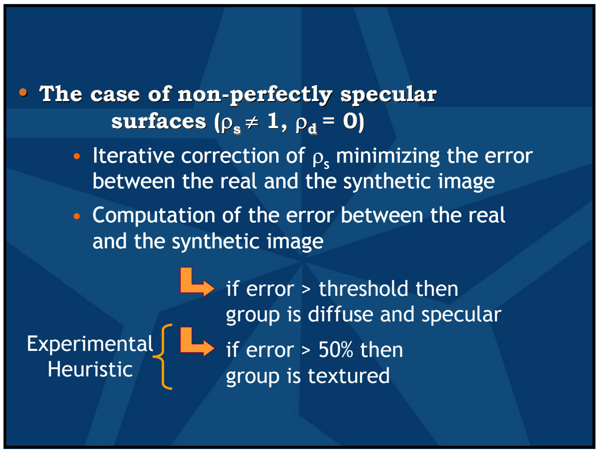
    ↳ Computation of the error between the real and the synthetic image

    ↳ if error > threshold then group is perfectly specular



- **The case of perfectly specular surfaces ($\rho_s = 1$, $\rho_d = 0$)**
  - The simplest case because ρ_d and ρ_s are constant
  - Computation of the error between the real and the synthetic image

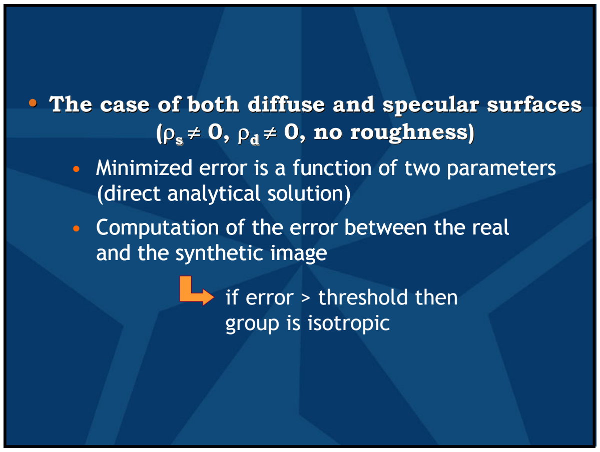
    ↳ if error > threshold then group is non-perfectly specular



- **The case of non-perfectly specular surfaces ($\rho_s \neq 1$, $\rho_d = 0$)**
  - Iterative correction of ρ_s minimizing the error between the real and the synthetic image
  - Computation of the error between the real and the synthetic image

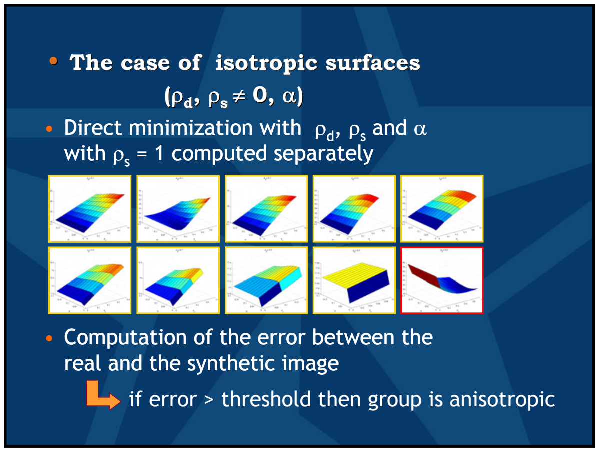
    ↳ if error > threshold then group is diffuse and specular

Experimental Heuristic { ↳ if error > 50% then group is textured



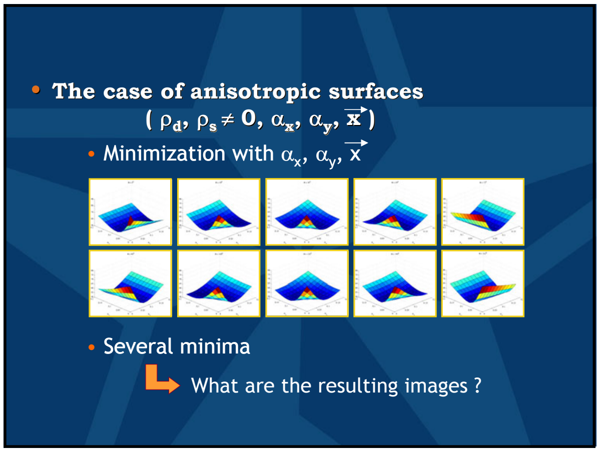
- **The case of both diffuse and specular surfaces ($\rho_s \neq 0$, $\rho_d \neq 0$, no roughness)**
  - Minimized error is a function of two parameters (direct analytical solution)
  - Computation of the error between the real and the synthetic image

    ↳ if error > threshold then group is isotropic

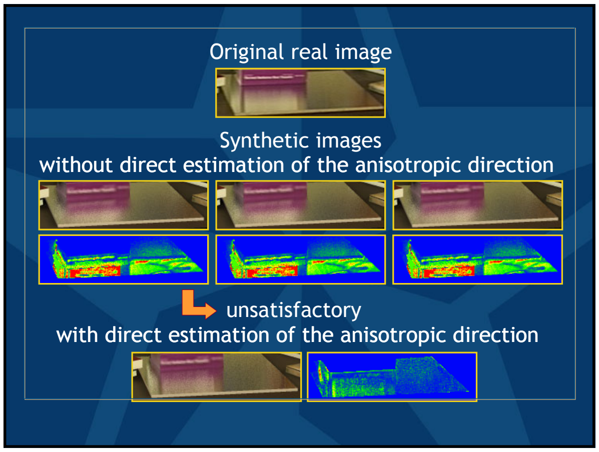


- **The case of isotropic surfaces (ρ_d, $\rho_s \neq 0$, α)**
- Direct minimization with ρ_d, ρ_s and α with $\rho_s = 1$ computed separately
- Computation of the error between the real and the synthetic image

  ↳ if error > threshold then group is anisotropic



- **The case of anisotropic surfaces**
  (ρ_d, $\rho_s \neq 0$, α_x, α_y, $\vec{x}$)
  - Minimization with α_x, α_y, $\vec{x}$
  - Several minima
    What are the resulting images ?

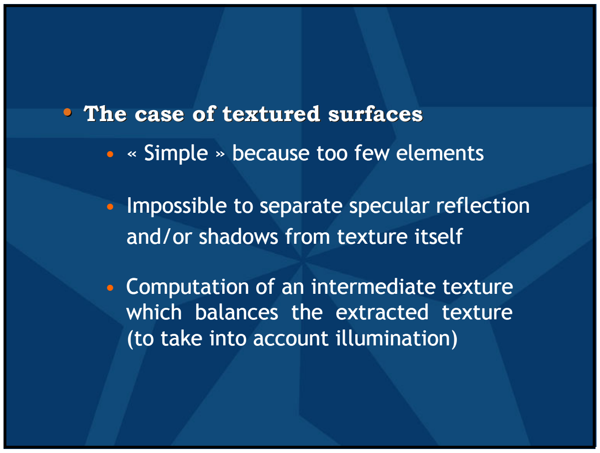


Original real image

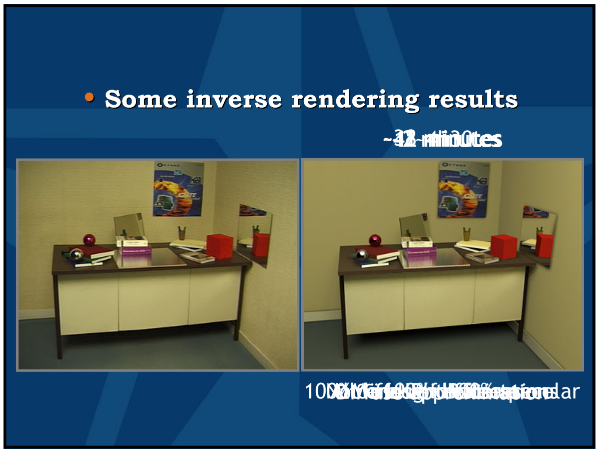
Synthetic images
without direct estimation of the anisotropic direction

unsatisfactory

with direct estimation of the anisotropic direction



- **The case of textured surfaces**
  - « Simple » because too few elements
  - Impossible to separate specular reflection and/or shadows from texture itself
  - Computation of an intermediate texture which balances the extracted texture (to take into account illumination)



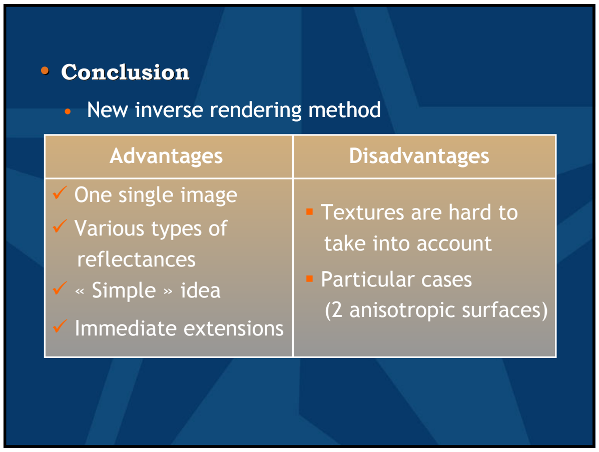
- **Some inverse rendering results**

[illegible] minutes

[illegible]

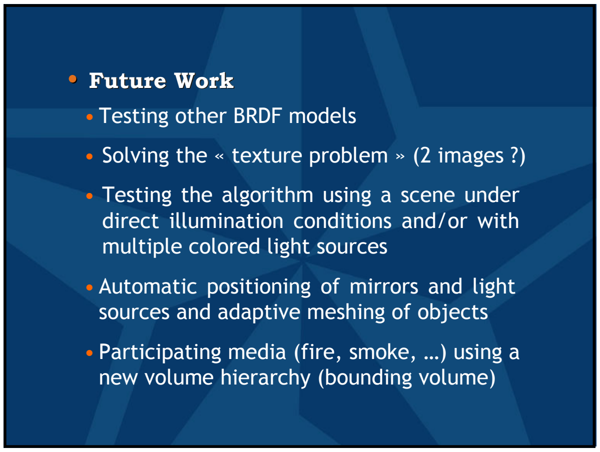
- **Some applications in Augmented Reality**

Illumination control
[illegible]
+ Geometry control

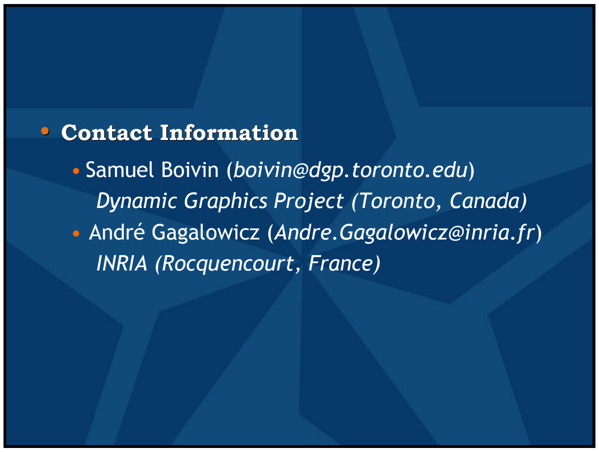


- **Conclusion**
  - New inverse rendering method

Advantages	Disadvantages
✓ One single image ✓ Various types of reflectances ✓ « Simple » idea ✓ Immediate extensions	▪ Textures are hard to take into account ▪ Particular cases (2 anisotropic surfaces)



- **Future Work**
  - Testing other BRDF models
  - Solving the « texture problem » (2 images ?)
  - Testing the algorithm using a scene under direct illumination conditions and/or with multiple colored light sources
  - Automatic positioning of mirrors and light sources and adaptive meshing of objects
  - Participating media (fire, smoke, …) using a new volume hierarchy (bounding volume)



- **Contact Information**
  - Samuel Boivin (*boivin@dgp.toronto.edu*)
    Dynamic Graphics Project (Toronto, Canada)
  - André Gagalowicz (*Andre.Gagalowicz@inria.fr*)
    INRIA (Rocquencourt, France)

Image-Based Rendering of Diffuse, Specular and Glossy Surfaces from a Single Image

Samuel Boivin[*] André Gagalowicz[*]

Mirages Project
INRIA-Rocquencourt

Abstract

In this paper, we present a new method to recover an approximation of the bidirectional reflectance distribution function (BRDF) of the surfaces present in a real scene. This is done from a single photograph and a 3D geometric model of the scene. The result is a full model of the reflectance properties of all surfaces, which can be rendered under novel illumination conditions with, for example, viewpoint modification and the addition of new synthetic objects. Our technique produces a reflectance model using a small number of parameters. These parameters nevertheless approximate the BRDF and allow the recovery of the photometric properties of diffuse, specular, isotropic or anisotropic textured objects. The input data are a geometric model of the scene including the light source positions and the camera properties, and a single image captured using this camera. Our algorithm generates a new synthetic image using classic rendering techniques, and a lambertian hypothesis about the reflectance model of the surfaces. Then, it iteratively compares the original image to the new one, and chooses a more complex reflectance model if the difference between the two images is greater than a user-defined threshold.
We present several synthetic images that are compared to the original ones, and some possible applications in augmented reality.

CR Categories: I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—modeling and recovery of physical attributes; I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Radiosity, Ray Tracing; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Color, Photometry, Shading;

Keywords: Image-Based Rendering, Reflectance Recovery, BRDF Models, Radiance, Radiosity, Rendering, Inverse Rendering, Rerendering, Global Illumination

1 Introduction

1.1 Overview of the problem

Since its origin, Computer Graphics has aimed at depicting reality. Rendering algorithms have been developed specifically to generate near-perfect images under realistic illumination conditions. It is often difficult to say if such images are realistic or not because there is no real reference such as a photograph. Moreover, the application may need to create novel viewpoints and/or novel illumination conditions from a sparse set of photographs. This is difficult to achieve without using image-based modeling and rendering algorithms. For example, suppose we want to insert a new synthetic object on top of a real anisotropic mirror inside a real scene. This operation clearly requires taking into account the interaction between the new object and its environment (especially this mirror). This is impossible to do, if we do not have an approximation of the reflectance properties of the real surfaces in the image. Therefore specific algorithms are necessary to recover these reflectance properties from the real images.

Many authors have contributed to the resolution of this problem [21, 25, 32, 31, 33, 26, 27, 34, 7, 41, 23, 24, 30, 29, 14, 11, 28]. The algorithms that they have produced vary greatly and not all can be re-used for our applications. Considerable work has been done for the reflectance estimation of an isolated object in particular illumination conditions [21, 25, 32, 31, 33, 26, 27] . Although these techniques often bring very detailed reflectance information (i.e. a full BRDF sometimes), their goal is more to replace the use of an expensive gonioreflectometer rather than to be able to change the viewpoint and/or the illumination. Recently, several methods have been developed to take into account the interaction between objects inside a real scene, from a sparse set of photographs [7, 41, 23, 24]. Fournier [14] proposed a different approach but with the use of a single image. However, his technique was limited to perfectly diffuse environments and was not be able to take into account specular surfaces. Our method has the similar ambition to recover an approximation of the BRDF of the surfaces from a single image, including the processing of specular, isotropic or anisotropic surfaces. This is extremely difficult to achieve because it is not possible to compute a full BRDF correctly without having several images, except for trivial cases.

We propose a hierarchical and iterative technique that computes the best possible approximation of a real image, using the error computed between the rerendered image and the real one. Each of the new images is generated by making more and more complex assumptions about the reflectance properties of the real surfaces. It is rendered by a global illumination software that takes into account these reflectance changes (see figure 1). The main advantages of our approach are: it does not need any special device to capture the real image (a classical camera is enough), and it estimates the reflectances of all types of surfaces (including anisotropic mirrors) from a single image without any particular constraint for the view-

[*] email: {Samuel.Boivin|Andre.Gagalowicz}@inria.fr

ACM SIGGRAPH 2001, 12-17 August 2001, Los Angeles, CA, USA
© 2001 ACM 1-58113-374-X/01/08...$5.00

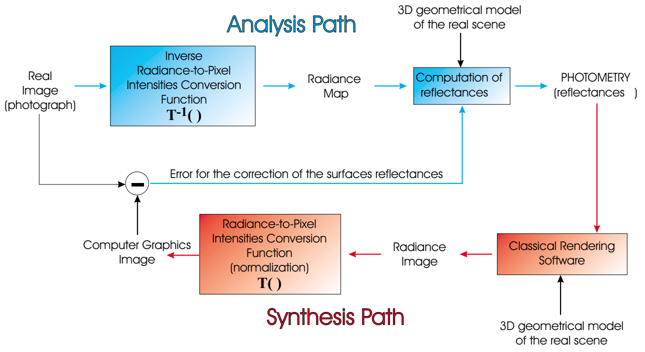


Figure 1: **General Principle of our Method** this figure shows the global scheme of the inverse rendering process. Initial data are: one real image and a 3D geometrical model of the scene.

point position, the light sources[1] or the objects orientation. The goal of our method is to recover an approximation of the BRDF of the surfaces, and to compute the best synthetic image preserving the real properties of the scene (a real mirror has to be simulated as a specular surface and not as a textured surface for example).

1.2 Organization of the paper

The paper is organized as follows. In the next section, we discuss previous work related to image-based rendering. Section 3 describes the bases and the tools necessary to our algorithm. In particular we introduce the notion of group which solves the problem of objects that are not directly seen in the image, and the BRDF model that we use. We also give a short description of the input data and the rendering software that we have developed. Section 4 describes the algorithm in full detail using the previously discussed tools. In particular, we explain the methods to process each case of reflectance property separately. In section 5, we describe one of the advantages inherent in our methodology: the possibility of analyzing some surfaces that are not directly seen in the real image, but indirectly through a mirror. Section 6 completes the technical discussion by explaining the optimizations that we have implemented to accelarate the rerendering process. Section 7 shows several results of rerendering, including images containing many kinds of photometric properties. Some applications are given in the domain of augmented reality, including rendering of new images under novel viewpoints, novel illumination conditions and the insertion/removal of objects. The last section gives some conclusions and future research directions.

2 Background and Related Work

All the techniques and ideas in this paper have been inspired by works about photorealistic rendering including global illumination and ray tracing, image-based modeling and BRDF modeling. However, the most relevant domains deal with *inverse rendering*, *image-based rendering* and *reflectance recovery*. We can split the *reflectance recovery* algorithms into three parts: direct measure of reflectances on the object using a specific device [37, 20, 2, 6], the extraction of reflectances from a set of images [21, 25, 32, 31, 33, 26, 27, 34, 7, 41, 23, 24], and the extraction of reflectances from a single image [30, 29, 14, 11, 28]. The last two parts may be subdivided into two categories, depending on whether the method takes into account energetic interreflections (using a global illumination algorithm for example) or not.

2.1 Reflectance Recovery using a Specific Device

Ward [37] proposed to directly measure the reflectances of an object, using a low-cost device. Ward introduced a device to estimate the five parameters of his anisotropic BRDF model, that he developed for these purposes. Karner et al. [20] presented another device using the Ward's BRDF model.

Baribeau et al. [2] described a method for measuring three reflectance parameters of several objects inside a scene. The diffuse reflectance, the Fresnel term and the roughness of the objects are estimated using a polychromatic laser range sensor. However, this method is limited to uniform reflectance properties over each object.

Dana et al. [6] suggest using a device containing a robotic manipulator and CCD camera to allow simultaneous measurement of the BTF (Bidirectional Texture Function) and the BRDF of large samples (about 10cm × 10cm).

2.2 Reflectance Recovery from Several Images

2.2.1 Methods without Global Illumination

Kay et al. [21] described a method to compute the surface reflectances using the Torrance-Sparrow light reflection model [35]. They used a depth map and four or eight images obtained with different point light sources. By increasing the number of intensity images, they estimated the parameters of the Torrance-Sparrow's model, reduced to three terms: the diffuse reflection coefficient k_d, the specular reflection coefficient k_s and the roughness factor c. Lu et al. [25] did not use any reflection model, but directly estimated the reflectances from the pixel intensities. Nineteen black and white images were captured using a custom device that turns around the object. For each incident angle of light, they built a reflection function, depending on the maximum pixel intensity in the image.

Y. Sato et al. [32, 31] proposed to register a range map (to get a 3D geometric model) and a set of color images of an object, using a 360 degrees rotation device. Next, they extracted the pixel intensities from the images and from the 3D model of the object reprojected onto the images by a Z-buffer algorithm. These parameters were used to separate and then compute the diffuse component and the specular component, i.e. the k_d term of the Lambert's model and the k_s and c terms of a simplified Torrance-Sparrow reflection model.

Y. Sato et al. [33] needed 120 color images and 12 range maps to compute the Torrance-Sparrow's parameters, separating the diffuse and the specular component. They recovered the BRDF of highly textured objects (this was impossible to do with previous techniques presented in [21, 25, 32]), and proposed the creation of new images under novel viewpoints and novel illumination conditions.

Marschner et al. [26, 27] directly estimated the Lafortune's et al. BRDF [22] of an object from a set of images (30). To obtain the BRDF, the radiance received by the pixels from the object is divided by the irradiance received by this object from the light source. He applied this computation to the rerendering of objects under novel illumination conditions.

Finally, Wong et al. [34] described a method that recovers the reflectance of each pixel of an image, considered as a set of small facets, each one having its own BRDF. The BRDFs are estimated from a set of images taken under different viewpoint and illumination conditions, as the ratio of the pixel intensity divided by the light source intensity. Wong et al. applied their method to the reillumination of the scene with new light sources.

2.2.2 Methods with Global Illumination

Debevec [7] used global illumination for augmented reality applications. To insert new objects inside a real image, he needed to take into account interreflections and compute the reflectances of

[1] In fact, the emittances of the light sources are supposed to be known. However, if it is not the case Fournier et al. [14] propose a method to recover them automatically.

the surfaces in the part of the scene influenced by this insertion. He created a geometrical 3D model of this part of the scene (called the *local scene*), and calculated manually the reflectance parameters of all the modeled objects. Each of the non-diffuse BRDF parameters are changed by the user iteratively until the rerendered image becomes close enough to the original one. The perfectly diffuse parameters are set by an automatic procedure.

Yu et al. [41] proposed a complete solution for the recovery of the surfaces BRDF from a sparse set of images captured with a camera (twelve of the 150 images were taken specifically to get specular highlights on surfaces). They built 40 radiance maps for the estimation of the reflectance parameters and the computation of the radiance-to-pixel intensities conversion function (camera transfer function) [8]. Using an image-based modeling software such as *Facade* [9], a 3D geometrical model of the scene was built from the set of images. All these data were then utilized to recover the BRDF of the modeled surfaces. Their method minimized the error on the parameters of the Ward's anisotropic BRDF model [37] to estimate the best possible BRDF for each object. This work was applied to the insertion of new objects in the scene, to the modification of the illumination conditions and to the rendering of a new scene under novel viewpoints. However, this method only works if at least one specular highlight is visible on an object. Otherwise this object is simulated as perfectly diffuse.

Loscos et al. [23] proposed a method based on an original idea from Fournier et al. [14]. Their algorithm recovers the diffuse reflectances of the surfaces inside a set of photographs of a scene, taking into account the textures of the objects (each surface has to be unshadowed in at least one image of the set). They applied their technique, to insert/remove objects and to modify the lighting conditions of the original scene (insertion of a new light source for example). More recently, Loscos et al. [24] extended this technique by removing the constraint of the unshadowed surfaces. To improve the results, they transformed their reflectance recovery algorithm into an iterative process. However, the method remained limited to perfectly diffuse surfaces (the mirrors are considered to be diffuse textured objects for example).

2.3 Reflectance Recovery from a Single Image

2.3.1 Methods without Global Illumination

K. Sato et al. [30] described an algorithm for the reflectance recovery of an isolated object from a single image and a 3D geometrical model of this object. They applied some constraints on the light source position and the camera parameters. In addition, they simplified the Torrance-Sparrow reflection model. This way, they estimated separately the diffuse component and the specular component to recover the uniform reflectance of the surface.

More recently, I. Sato et al. [29] proposed to recover the BRDF of an object, using the shadows generated by the surfaces of the scene. They used a single omnidirectional image of the environment and a 3D geometrical description of the surfaces. They developed a 6-step iterative algorithm to minimize the error between the real and the synthetic image with respect to the BRDF parameters of the surfaces.

2.3.2 Methods with Global Illumination

A pioneering work in this domain was completed by Fournier et al. [14] in 1993. He proposed to rerender an original image using a 3D representation of the scene (including the positions of the light source and the camera parameters) and a single image of this scene. All the surfaces are considered as perfectly diffuse, and they used their reprojection on the real image to estimate their reflectances. A radiosity-based algorithm then computes an image applying these reflectances to a progressive radiosity technique [4] to obtain a new synthetic image.

An extension of the previous method was developed by Drettakis et al. [11]. They proposed an interactive version of the initial paper and added a vision algorithm for the camera calibration and the 3D geometrical model automatic positioning. They described a slightly different technique for the estimation of the reflectances of the surfaces and they used a hierarchical radiosity algorithm [18] to compute a new synthetic image close to the real one.

An approach similar to Fournier et al.'s was chosen by Gagalowicz [28]. It included a feedback that compares the real image to the synthetic one. He described a technique to generate a new synthetic image from a single one (except the 3D geometrical model, which was built from two stereo images) using an iterative method that minimizes the error between the real image and the synthetic one. However, this technique is limited to a pure lambertian approximation of the surface reflectances.

3 Elements of Reflectance Recovery

3.1 The Notion of Group

The inputs of our reflectance recovery algorithm are separated into two categories, the 3D geometrical model of the scene and a single image of this scene captured with a standard camera. This method is based on the extraction of the object reflectances from the pixels covered by the projection of these objects in the image (as described later in section 4).

Using a single image to recover all the surface reflectances of the scene raises several problems related to the geometrical model and the size of the projection of the objects in the image. First of all, there are generally many surfaces that are not directly visible in the real image. It is then extremely difficult (sometimes impossible) to compute their reflectances because no information is available about them. This is not important if the position of the observer is never changed. However, it is usual to modify this position especially in augmented reality applications. Therefore, we introduce the notion of *group* of objects and surfaces. These *groups* specify the objects and the surfaces which have the same reflectance properties. This is a very fast manual operation left to the user after or during the geometrical modeling process. For example, in figure 2, the 'red cube' was modeled as a *group* containing six planar objects which have the same reflectance properties. Our reflectance algorithm will then use this description to propagate the estimated reflectance from the three visible faces of the cube to the three other ones.

This *group* notion often solves the second modeling problem which could happen during the reflectance estimation. Indeed, the area covered by the projection of some objects in the real image could be too small to give a good approximation to the reflectance of these objects. Therefore, if the user joins these objects with others which have the same reflectance and a bigger projection area in the real image, it becomes possible to obtain a better approximation of their reflectance. However, if there are no other bigger objects, a very rough approximation of the reflectance will be computed for these small objects, and the resulting image may be biased. This problem is inherent in all image-based rendering methods [7, 41, 23, 24, 14, 11, 28] which use the area covered by the projection of an object in the real image to determine its reflectance. Nevertheless, as our method uses a feedback through the comparison between the real and synthetic image, bias is considerably reduced.

3.2 Reflectance Model and Data Description

For the past several years, the construction of a 3D geometrical model from a single image or a set of images has been widely investigated and is known as *image-based modeling* (see [9] for an overview of these methods). In our paper, the 3D geometrical model is built interactively using *Alias|Wavefront*'s *Maya* modeler. The positioning of the full 3D geometrical model of figure 2 took around six hours to be complete, including the recovery of the camera parameters and the light sources positions. Typically, for the camera parameters we use the Dementhon and Davis [10] technique combined with a *downhill simplex* minimization method [17, 19]. The light sources have been modeled approximately (because of their complex geometry) and they have been placed manually with a precision of ± 5cm[2]. Our photometric recovery method is based on the use of Ward's reflectance model [37]. We chose the same BRDF model as Yu et al. [41] because of its small number of parameters and its ability to simulate anisotropic surfaces. This model only requires the knowledge of five parameters for a complex BRDF: ρ_d the diffuse reflectance, ρ_s the specular reflectance, $\vec{x}$ the anisotropy direction (called the *brushed direction*) and the anisotropic roughness parameters α_x and α_y (see [37] for a detailed description of this BRDF model). Furthermore, this model avoids the costly computation of the Fresnel term which has been replaced by a normalization factor.

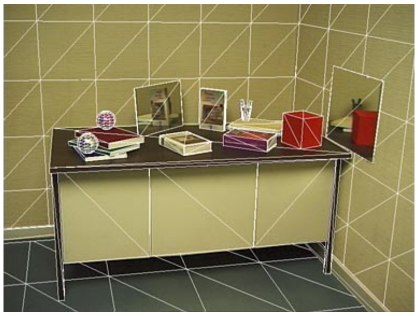


Figure 2: Example of a real image with the superposition of its 3D reconstructed geometrical model (in white)

When the 3D geometrical model (objects, camera and light sources positions) and the photometric model (reflectances and light sources intensity) are determined, it is possible to render a synthetic image using a classical rendering software such as *Radiance* [38]. We developed our own rendering software called *Phoenix* to obtain a high-performance computing power and to take advantage of the specific architecture of the Silicon Graphics workstations used[3]. *Phoenix* is a global illumination software. It computes the form factors of a progressive radiosity system [4] using a 64 bit A-Buffer [3, 13] mapped on each face of the hemicube [5]. This increases the resolution of each face of the hemicube by a factor of 64 with a negligible increase in computation time, with respect to a classical Z-Buffer software.

Moreover, *Phoenix* uses advanced *OpenGL* programming techniques called *offscreen rendering* to compute the index buffers (or *item buffers* [39]) necessary for the extraction of the pixel intensities from the original image and the synthetic one. Each number in the index buffer indicates either a group number, or an object number, depending on whether we need to compute the reflectance of a group or of an object.

4 Inverse Rendering from a Single Image

4.1 Overview of the Algorithm

The core of our technique is incremental and hierarchical (see figure 3). It is incremental because the surface reflectances evolve to their optimum value. It is hierarchical because the general algorithm forces the surface BRDFs to be more and more complex if the error between the real and the synthetic image does not decrease for these surfaces. This algorithm is iterative and will proceed to successive corrections of the surface reflectances by minimizing the error between the real and the synthetic image. Indeed, each computed error for a group of objects having the same photometric properties drives the correction of their reflectance. Our technique successively applies the selected assumption on the group reflectances until the error became smaller than a user-defined threshold. The notion of threshold and how to fix its value to give them will be discussed in the section 6.

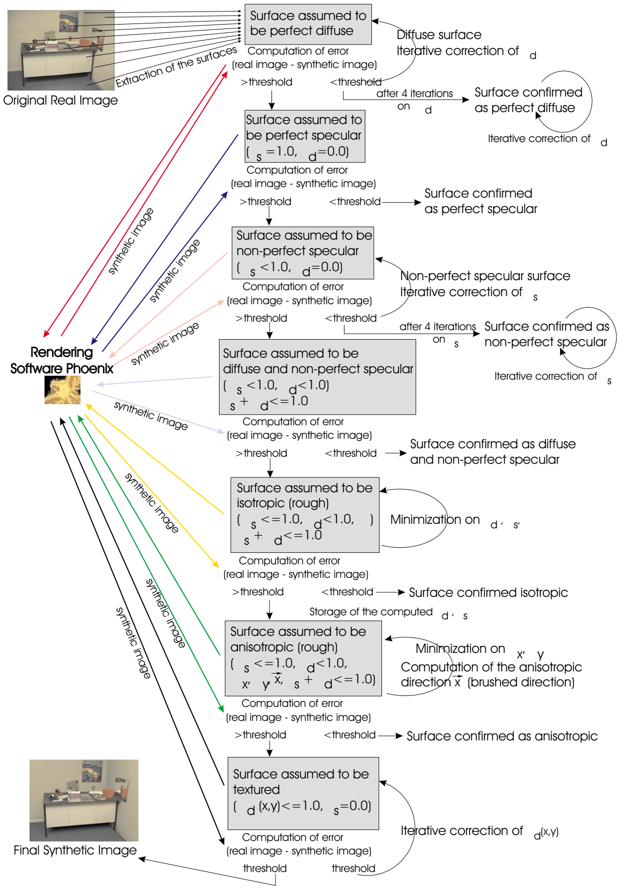

Original Real Image
Extraction of the surfaces
Surface assumed to be perfect diffuse
Computation of error (real image - synthetic image)
>threshold
<threshold
Diffuse surface
Iterative correction of d
after 4 iterations on d
Surface confirmed as perfect diffuse
Iterative correction of d
Surface assumed to be perfect specular (s =1.0, d=0.0)
Computation of error (real image - synthetic image)
>threshold
<threshold
Surface confirmed as perfect specular
Surface assumed to be non-perfect specular (s <1.0, d=0.0)
Computation of error (real image - synthetic image)
>threshold
<threshold
Non-perfect specular surface
Iterative correction of s
after 4 iterations on s
Surface confirmed as non-perfect specular
Iterative correction of s
Rendering Software Phoenix
synthetic image
synthetic image
synthetic image
synthetic image
synthetic image
synthetic image
synthetic image
Surface assumed to be diffuse and non-perfect specular (s <1.0, d<1.0) s + d<=1.0
Computation of error (real image - synthetic image)
>threshold
<threshold
Surface confirmed as diffuse and non-perfect specular
Surface assumed to be isotropic (rough) (s <=1.0, d<1.0,) s + d<=1.0
Minimization on d ' s'
Computation of error (real image - synthetic image)
>threshold
<threshold
Surface confirmed isotropic
Storage of the computed d ' s
Surface assumed to be anisotropic (rough) (s <=1.0, d<1.0, x' y' x, s + d<=1.0)
Minimization on x' y
Computation of the anisotropic direction x (brushed direction)
Computation of error (real image - synthetic image)
>threshold
<threshold
Surface confirmed as anisotropic
Surface assumed to be textured (d (x,y)<=1.0, s=0.0)
Iterative correction of d(x,y)
Computation of error (real image - synthetic image)
threshold
threshold
Final Synthetic Image

Figure 3: General iterative and hierarchical algorithm for reflectance recovery. Each surface of the scene is analyzed separately, depending on the assumption about its reflectance (perfectly diffuse, perfectly specular, etc.). If the assumption is false (the error between the real and the synthetic image is big), then the surface reflectance is assumed to be more complex (hierarchical principle). If the assumption is correct then the surface reflectance is modified accordingly in order to minimize the error between the two images (iterative principle). During each global rerendering iteration, the reflectances of all surfaces are then continuously updated, to take into account the incident energy coming from any surface for which the BRDF has changed (a diffuse surface which became *perfectly specular* for example).

We start the algorithm with the *perfectly diffuse* case without considering texture (the diffuse reflectance of a group is computed averaging the radiances covered by its projection in the real image). All the surfaces are then considered as perfectly lambertian, and the rendering software (*Phoenix* in this case[4]) computes a new approximation of the image. If the difference between the real and the synthetic image for a group is greater than a fixed threshold on all the group projection, then the reflectance of this group is considered as *perfectly specular* for the next rerendering iteration. If, after *Phoenix* has recomputed a new image using the new assump-

[2] Our technique can be used regardless of how the geometry is acquired.

[3] This work was carried out on a SGI Octane SI 2× R12000 300Mhz.

[4] It is possible to use any other global illumination rendering software, such as *Radiance* [38] for example.

tion, the error for this group remains large, then its reflectance is simulated as *non-perfectly specular*. We apply the same principle to change again the group reflectance to a *both diffuse and specular* one. Until then, all the surfaces were considered with no roughness term (only a ρ_d and a ρ_s were estimated). In the next assumption, if the difference between the two images still produces big errors, they are considered as isotropic and a roughness factor (α) has to be evaluated. This assumption is extended to anisotropic properties if the user-defined threshold for the error has not been reached. If all assumptions have failed, the group is presumed to be highly textured. Since only a single image is available, it is extremely difficult and sometimes impossible to create a combination between this texture and other reflectance properties (a glossy textured surface for example). This situation is discussed in paragraph 4.7.

4.2 The case of perfectly diffuse surfaces

One of the simplest cases of reflectances is the *perfectly diffuse* one. During the first inverse rendering iteration, all the objects of the scene are simulated as perfectly diffuse. A diffuse reflectance (ρ_d) is then computed for each group, as the average of radiances covered by the projection of the groups in the original image. This technique is different from Drettakis et al. [11, 14] because we do not pay attention to the texture of the surfaces. It is interesting to note that some textured surface may be simulated using a pure diffuse reflectance (as shown in figure 14), to create a good visual approximation. This method is very different from [11, 14] because it is not limited to the computation of the average reflectance to produce the new final synthetic image. We correct this reflectance iteratively until the error between the original and the rerendered image becomes small. For an object, this error is computed as the ratio between the average of the radiances[5] covered by the projection of the groups in the original image, and the average of the radiances covered by the projection of the groups in the synthetic image (see equation 1).

$$\widehat{\varepsilon_j} = \frac{\widehat{B_{o_j}}}{\widehat{B_{n_j}}} = \frac{T^{-1}(\widehat{P_{o_j}})}{T^{-1}(\widehat{P_{n_j}})} \qquad (1)$$

$\widehat{B_{o_j}}$ and $\widehat{P_{o_j}}$ are respectively the average of the radiances and the pixels covered by the projection of object j in the original image.
$\widehat{B_{n_j}}$ and $\widehat{P_{n_j}}$ are respectively the average of the radiances and the pixels covered by the projection of object j in the synthetic image.
$T()$ is the camera transfer function (a γ correction function here).

Since the average radiance $\widehat{B_j}$ of object j is proportional to the diffuse reflectance ρ_{dj}, the iterative correction of the ρ_{dj} can be written for each rerendering iteration k as:

$$\rho_{di_{k+1}} = \rho_{di_k} \times \widehat{\varepsilon_i} \qquad (2)$$

$$\rho_{di_{k+1}} = \rho_{di_k} \times \frac{\sum_{j=1}^{n_i} f(\widehat{\varepsilon_j}) \cdot (\widehat{\varepsilon_j} \times m_j)}{\underbrace{\sum_{j=1}^{n_i} f(\widehat{\varepsilon_j}) \cdot m_j}_{\neq 0}} \qquad (3)$$

$$\text{and } f(\widehat{\varepsilon_j}) = \begin{cases} 0 & \text{if } \widehat{\varepsilon_j} \geq (1+\lambda) \cdot md \\ 1 & \text{else} \end{cases}$$

$\widehat{\varepsilon_i}$ and $\widehat{\varepsilon_j}$ are respectively the total error between the original and the synthetic image for group i and object j.
n_i is the number of objects for group i.
md is the median of the errors (selects the middle value of the sorted samples).
λ is the authorized dispersion criteria.
m_j is the number of pixels covered by the projection of object j.

The function $f()$ eliminates problems generated by smaller objects for which the error is very important, because they are more sensitive to the image noise (their projection in the image cover a small amount of pixels). An example of iterative correction of ρ_d is provided by figure 4 on a very simple synthetic scene, nevertheless containing high color bleeding effects (see how the green cube is influenced by the blue floor for example).

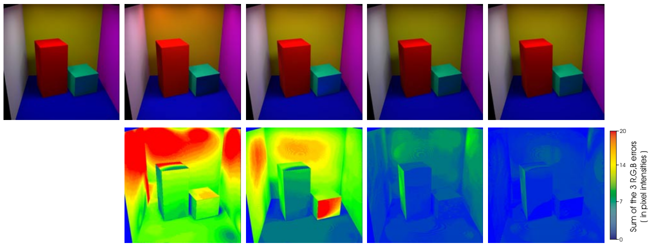


Figure 4: In the top row, from left to right: the original synthetic image (top left) generated using a rendering software was rerendered for 4 iterations (the next four images). The differences between this original image and the regenerated images are shown in the bottom row and displayed using a specific error colormap (at the bottom right). We observe a regular decrease of the error from left to right.

As textures are not taken into account in this section, we only consider a diffuse reflectance parameter ρ_d. It could be interesting and maybe faster to directly inverse the radiosity equation as suggested by Yu et al. [41]. If we know the radiances, the emittances and the full geometry (i.e. the form factors), it is possible to directly solve the radiosity equation [16] for the reflectances. However, this is not so simple, because we work with a single image. Because of this, there may be some surfaces that are not directly visible in the original image. Therefore, their radiosities are unknown and it is impossible to guess their values. Thus, we can not inverse the radiosity equation.

4.3 The case of perfectly and non-perfectly specular surfaces

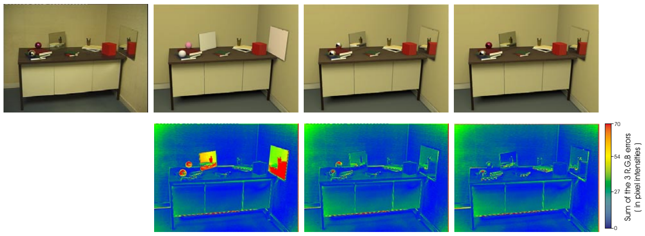


Figure 5: Simulation of hierarchical inverse rendering, where the top row from left to right consists of the real image captured with a camera, the synthetic image with a pure diffuse assumption (first iteration), the synthetic image with perfectly diffuse and perfectly specular assumptions (fifth iteration) and the synthetic image with pure diffuse and non-perfectly specular surfaces (seventh iteration). On the bottom row, we can see the error images corresponding to the difference between the real and the synthetic image.

If the previous diffuse hypothesis about the surface reflectance failed, it is now considered as a *perfect mirror*. It is the easiest case

[5] These radiances have been obtained using the inverse of the camera transfer function that was simulated as a γ correction function with a γ value of 2.2 according to Tumblin et al. [36]. However a more powerful algorithm could be applied if we had more than one photograph of our scene [8].

to solve because the diffuse reflectance of a perfect mirror has a null value ($\rho_d = 0$) and its specular reflectance is equal to 1 ($\rho_s = 1$). It is worth noting that there is no need to iterate on the specular reflectance and a new synthetic image can be directly rendered. On the other hand, the reflectance for a non-perfectly specular object has to be iteratively modified to obtain an optimum ρ_s. The iterative correction of ρ_s is similar to equation 3, except ρ_d has to be replaced by ρ_s. An example of the use of the hierarchical algorithm on a scene containing both diffuse, non-perfectly specular surfaces is shown in figure 5.

4.4 The case of both diffuse and specular surfaces with no roughness factor

At this point of the algorithm, surfaces with big errors are now considered as both diffuse and specular ($\rho_d \neq 0$ and $\rho_s \neq 0$) but still with no roughness.
The differences between the real image and the synthetic are minimized as a function of ρ_d and ρ_s (in the Ward's BRDF model [37]):

$$(T^{-1}(I_{synth}) - T^{-1}(I_o))^2 = \sum_{i=1}^{nbg}(\rho_d \cdot B_d + \rho_s \cdot B_s - T^{-1}(I_o))^2$$

with nbg, the number of pixels covered by the group projection.

This minimization has an analytical solution for each wavelength R, G, B:

$$\begin{pmatrix}\rho_d \\ \rho_s\end{pmatrix} = \begin{pmatrix}\sum\limits_{nbg} B_d T^{-1}(I_o) \\ \sum\limits_{nbg} B_s T^{-1}(I_o)\end{pmatrix}\begin{pmatrix}\sum\limits_{nbg} B_d^2 & \sum\limits_{nbg} B_d B_s \\ \sum\limits_{nbg} B_d B_s & \sum\limits_{nbg} B_s^2\end{pmatrix}^{-1}$$

In practice, such surfaces in real cases are very rare but not impossible. For example, the top face of the desk in the figure 14 presents some photometric properties very close to this approximation.

4.5 The case of isotropic surfaces

Until now, all the surfaces were supposed to be without roughness. In the case of an isotropic surface, the diffuse reflectance ρ_d, the specular reflectance ρ_s and a roughness coefficient α have to be recovered according to Ward's BRDF model.

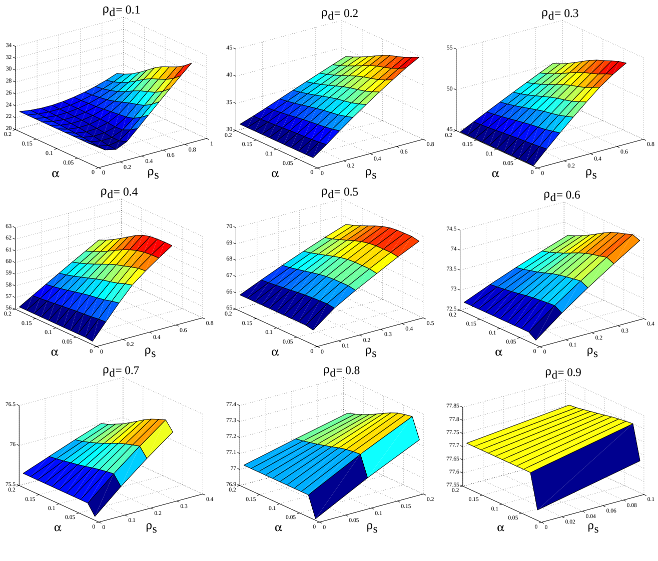


Figure 6: Error function (synthetic image - real image), for a fixed diffuse reflectance, with respect to variations of the isotropic values α, and ρ_s specular reflectance. The evolving steps are 0.018 for α and 0.1 for ρ_s.

A first idea is to use a classical minimization algorithm to solve for these three paremeters. However, the error function (difference between the real and the synthetic image) for an anisotropic surface is radically different if ρ_d is varying in $]0.0; 1.0[$ (figure 4.5) or if ρ_d has a null value (figure 7). Directly minimizing the error function for ρ_d, ρ_s and α in the interval $[0.0; 1.0[$ is thus not possible. We propose to miminize the error function using two separate error functions: one for the interval $]0.0; 1.0[$ and the other for the $\rho_d = 0$ particular case. The minimization algorithm (we use the *downhill simplex* method [17, 19] for the two minimizations) that provides the smallest error will determine the final value of ρ_d, ρ_s and α. One of the disadvantages of the method is that it could take a lot of time minimizing such functions. Indeed, these isotropic surfaces use ray-tracing [1] techniques for their correct simulation. Even if optimization techniques greatly accelerate the rendering [15, 12], it still could take around one hour and fifty minutes to recover the ρ_d, ρ_s and α values (using ten bounced rays for each primary ray (nine per pixel) that reached a glossy surface). In fact, the optimum values of ρ_d and ρ_s are found in only two minutes because the resulting value does not need to be obtained with a precision better than $1 \cdot 10^{-2}$ (the visual difference became imperceptible). On the other hand, α requires a determination with a $1 \cdot 10^{-4}$ precision (according to Ward [37], the α parameters may vary between 0.001 for a perfectly specular surface to 0.2 for a mostly diffuse surface).

Figure 8 shows the result of these minimizations: the aluminium surface (in the center of image) has been simulated as isotropic, and an optimum value of $\rho_d = 0.0$ and $\rho_s = 1.0$ has been found. However the error image shows that maybe a better approximation seems to be possible for this particular surface. The error remains important in the extent of the specular reflection area of the two books on this surface. Therefore a more complex BRDF is needed and the algorithms tries now to simulate the surface as an anisotropic one.

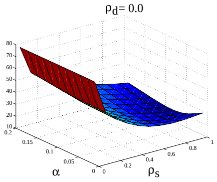


Figure 7: Error function (synthetic image - real image), for $\rho_d = 0$ with respect to variations of α (isotropy value), and ρ_s (specular reflectance).

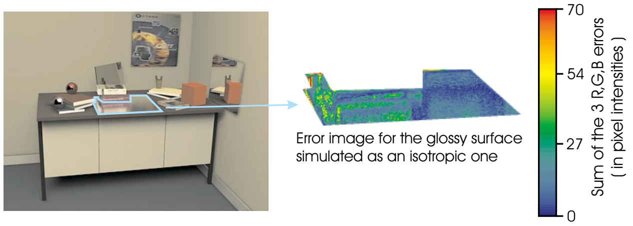

Error image for the glossy surface simulated as an isotropic one
70
54
27
0
Sum of the 3 R,G,B errors
(in pixel intensities)

Figure 8: Approximation of the aluminium surface (anisotropic) of the real image (left) by an isotropic surface in the synthetic image (center). The error between these two images for the aluminium surface is visible in the right image. We remark that the error is still important in the area of the specular reflection of the books. The red pixels correspond to a high error but they are not significant because they are coming from an approximative positioning of the 3D geometrical model ont the image, especially on the edges of the objects.

4.6 The case of anisotropic surfaces

Working with anisotropic surfaces is clearly the most complicated case of our algorithm because the anisotropic model of Ward requires minimizing a function of five parameters: the diffuse reflectance ρ_d, the specular reflectance ρ_s, the anisotropy direction $\vec{x}$ (or *brushed direction* [37]) and the roughness factors α_x, α_y. However, it is possible to keep the previous ρ_d and ρ_s values computed for the isotropic case: the error functions (see figure 4.5 and 7) show that the ρ_s parameter is not correlated to the α parameter, because these functions are quite constant with respect to α. We may then suppose that the ρ_d and ρ_s do no differ from the isotropic case to the anisotropic one.

The error function to minimize has now three parameters left (see figure 9). We remark on this figure that for a given rotating angle θ of the vector $\vec{x}$ and varying values of α_x and α_y, this error function presents several minima on all the curves, and they are very similar for all θ values. This confirms that a standard minimization algorithm will probably not find a global minimum.

To prove this assumption, we have computed the four images corresponding to the four smallest minima found by a *downhill simplex* minimization algorithm (figure 10). It is interesting to note that the rerendered images remain far from the original one and that the error is bigger than for the isotropic case. This brings us to the conclusion that a minimization procedure is not the correct way to solve the anisotropic case. Therefore, we propose to determine the anisotropy vector $\vec{x}$ directly from the real image.

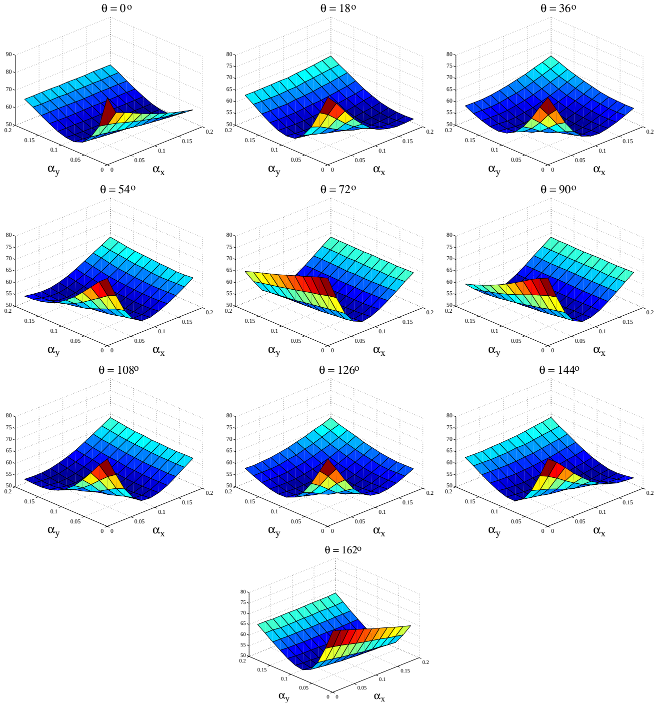


Figure 9: Error function (synthetic image - real image), for different anisotropy directions $\vec{x}$ (the vector is turned around the normal to the surface using a step of 18 degrees) with respect to variations of the roughness parameters α_x, α_y (with a step of 0.018). The diffuse reflectance and the specular reflectance terms have been estimated during the isotropy analysis.

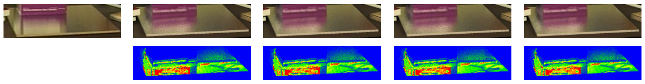


Figure 10: The first image (top left) is the original one (reduced here to the interest area). The next four images have been produced using the four smallest minima found by the minimization algorithm. We can see that all of these images are far from the original one (the vertical black line on the white book (see figure 8) has disappeared from the specular reflection) and that a lot of details have been smoothed. The error colormap remained the same as on figure 8.

In a first step, we consider the anisotropic surface as a perfect mirror and compute a synthetic image. Next, we estimate the difference between the real image and the synthetic one to visualize the part of the anisotropic mirror where the specular reflection is "extended". This area corresponds to an attenuation of the specular reflection, and this effect is always very important in the direction perpendicular to the brushed direction (or anisotropy direction). In a second step, we compute an index buffer for this mirror of all the surfaces visible through it. We then look for a reference surface that has the biggest reflection area on the anisotropic surface, while being as close as possible to it. This surface is then selected in a such manner that the ratio $\frac{\text{Area(reflected surface)}}{d(S,P)}$ is maximized (with d(S,P), the euclidean distance between the center of gravity of the selected surface and the center of gravity of the anisotropic mirror). The motivation of this choice resides in the fact that surfaces very far from the anisotropic object exhibit a reflection pattern that is too small or too noisy to be usable for the recovery of the brushed direction. In a third step, the anisotropy direction is sampled creating $\vec{x}$ vectors around the normal to the anisotropic surface. Each of these sampled directions determine a direction to traverse the error image and compute the average of the standard error deviations computed in the error image. Finally, the algorithm selects the direction for which this average value is the smallest one (see figure 11). Figure 12 summarizes the complete procedure.

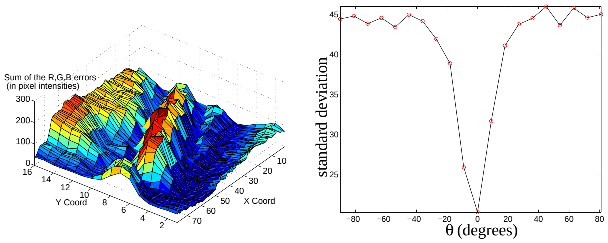


Figure 11: The selected object used here to recover the anisotropy direction is the violet book of the lower left real image of figure 14. The 3D surface (left image) shows the error image for the difference between the perfectly specular reflection area of this selected object, and its corresponding area in the real image. The 2D curve (right) shows the average of the standard error deviations computed from the error image along the sampled anisotropy directions (see also figure 12).

Once the anisotropy direction $\vec{x}$ has been recovered, a *downhill simplex* minimization algorithm is used to estimate the roughness parameters α_x and α_y. Typically, for the synthetic image in the lower right corner of the figure 14, it took 50 iterations and $2h30$ to recover the full BRDF of the anisotropic surface. The algorithm found an optimum anisotropy vector for a rotation angle of 0 degrees and then minimized the error function of the upper left corner of the figure 9. The estimated values of α_x and α_y were 0.01 and 0.062 respectively.

4.7 The case of textured surfaces

When the simulation of a surface as anisotropic still produces big errors in the difference image, we proceed to texture extraction. Extracting the texture from the real image is an easy task that can be realized using the technique proposed by [40] for example. However, we have to extract this texture while taking into account the fact that it already has received the energy from the light sources, and that the pixels covered by its projection in the real image contain this information. Otherwise, if we send the energy of the light sources to these textures again, they will be over-illuminated. Therefore, we introduce here a notion called *radiosity texture* that balances the extracted texture with an intermediate texture in order to minimize the error between the real and the synthetic image. As for the perfectly diffuse reflectance case, this intermediate texture is computed by an iterative method.

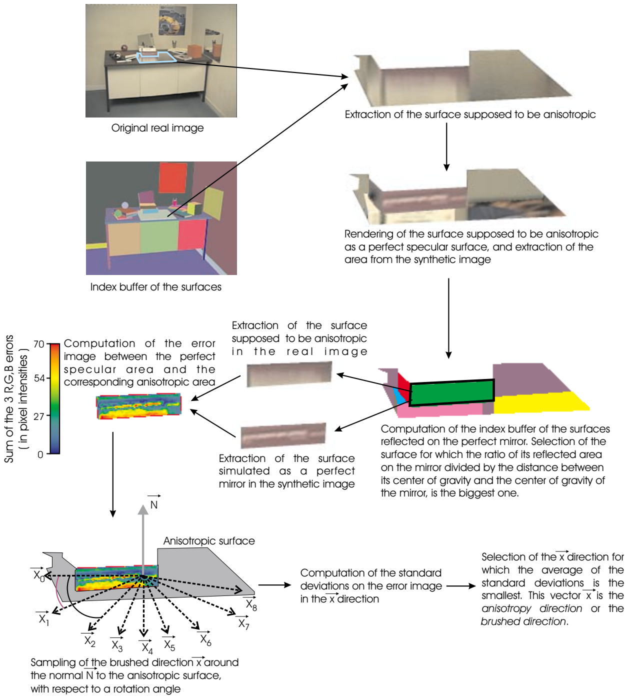

Original real image
Extraction of the surface supposed to be anisotropic
Index buffer of the surfaces
Rendering of the surface supposed to be anisotropic as a perfect specular surface, and extraction of the area from the synthetic image
Sum of the 3 R,G,B errors (in pixel intensities)
70
54
27
0
Computation of the error image between the perfect specular area and the corresponding anisotropic area
Extraction of the surface supposed to be anisotropic in the real image
Extraction of the surface simulated as a perfect mirror in the synthetic image
Computation of the index buffer of the surfaces reflected on the perfect mirror. Selection of the surface for which the ratio of its reflected area on the mirror divided by the distance between its center of gravity and the center of gravity of the mirror, is the biggest one.
$\vec{N}$
Anisotropic surface
$\vec{X_0}$
$\vec{X_1}$
$\vec{X_2}$
$\vec{X_3}$
$\vec{X_4}$
$\vec{X_5}$
$\vec{X_6}$
$\vec{X_7}$
$\vec{X_8}$
Computation of the standard deviations on the error image in the $\vec{x}$ direction
Selection of the $\vec{x}$ direction for which the average of the standard deviations is the smallest. This vector $\vec{x}$ is the *anisotropy direction* or the *brushed direction*.
Sampling of the brushed direction $\vec{x}$ around the normal $\vec{N}$ to the anisotropic surface, with respect to a rotation angle

Figure 12: Computation method of the anisotropy direction $\vec{x}$ for a glossy surface.

At the first iteration, the texture used to rerender the image is the texture directly extracted from the real image. At the second iteration, the texture used to obtain the resulting synthetic image is multiplied by the ratio between the newly extracted texture of this synthetic image and the texture of the real image. This iterative process stops when the user-defined threshold for textured surface has been reached. The textures of the poster and the books in the rerendered images of section 7 have been obtained using this technique. The problem of this method is that it computes a texture including the shadows, the specular reflections and the highlights. Typically, suppose that we have a marbled floor on which a sphere is reflected. The texture of this floor in the real image then includes the marble characteristics, its reflectance properties and the sphere reflection including its own reflectance properties. How to extract the marble characteristics only and independently of the rest of the scene ? This is an extremely hard problem, and according to Y. Sato et al. [33] no algorithm has been proposed yet to solve it using a single image.

5 Advanced Analysis of Reflectances

Our inverse rendering procedure provides the opportunity to analyze the reflectances of some surfaces that are not directly seen in the original image. Indeed, if a surface is detected and confirmed as a perfectly or non-perfectly specular one, we can extend our reflectance recovery algorithm to the surfaces that are seen through this mirror in the real image.

First of all, the index buffer of the groups visible through the mirror are computed using a ray tracing algorithm. If there exists a surface in this buffer that was not directly visible before in the real image, then its reflectance is computed taking into account the current assumption made for its group reflectance (the surface has the same photometric properties as its group). In the next iteration, this reflectance is balanced by the mirror reflectance (if it is a non-perfect one), and it is then considered for the correction of the group reflectance (see figure 13).

To our knowledge, this is the first time that an image-based rendering technique deliberatley exploits mirror surfaces to enhance the BRDF recovery process in a scene.

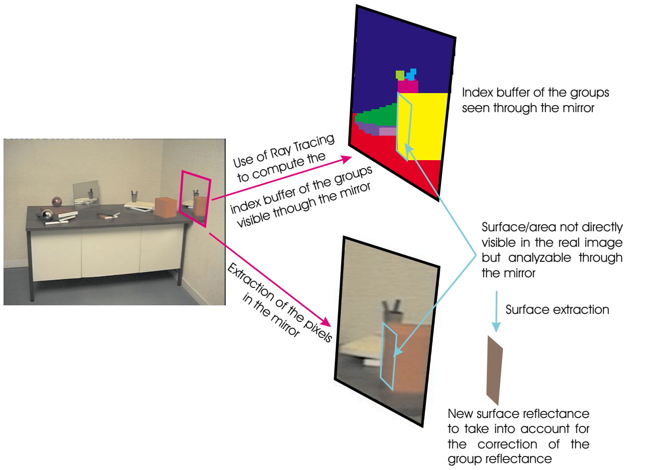

Index buffer of the groups seen through the mirror
Use of Ray Tracing to compute the index buffer of the groups visible through the mirror
Extraction of the pixels in the mirror
Surface/area not directly visible in the real image but analyzable through the mirror
Surface extraction
New surface reflectance to take into account for the correction of the group reflectance

Figure 13: Example of surface not directly seen in the original image. The reflectance of this surface becomes computable through the mirror put on the right wall. This surface belongs to the 'cube' group and now contributes to the estimation of its reflectance. If this face had a specific reflectance (new group), it would be computable only at this point.

6 Optimizations and Determination of Thresholds

Since the reflectance recovery algorithm takes around two hours to simulate an isotropic surface, and two more hours in an anisotropic case, this means that all textured surfaces (which is the final hypothesis after the isotropy and the anisotropy tests) will need four hours to be correctly estimated. This is not acceptable when a lot of surfaces are textured in a real image, but the computing time could be greatly reduced if we can find that the surface is textured before treating the isotropic case. Therefore we introduced a heuristic to solve this problem. It is related to the choice of the thresholds that determine if a surface is correctly simulated. Indeed, after each synthetic image has been produced, it is compared to the real one using a user-defined error threshold to know if the assumptions made about the surface reflectances are correct. For the presented images, the following thresholds were used to produce the results of the section 7. Such thresholds are not critical to the behavior of the technique but will affect speed because it will always find a solution regarding the fixed thresholds.

From the case of the perfectly diffuse assumption up to the isotropic one, the sum of the three R,G,B errors coming from the difference between the real and the synthetic image must have a value smaller than 5%. However, during the non-perfect specular assumption, if the error is greater than 50%, we can directly avoid the isotropic and the anisotropic cases and so greatly increase the performance of the algorithm. We do not have a formal characterization of this optimization, but in practice it seems to work well (see section 7). The isotropic-to-anisotropic threshold has been chosen equal to 1%, to ensure that the algorithm tries the anisotropic case. On the other hand, the threshold used to come to a texture assumption is equal to 5%. Finally, the last threshold is a global threshold that forces all the groups in the synthetic image to have an error smaller than 5%.

7 Results

All the following synthetic images have been generated using *Phoenix* as rendering and inverse rendering software. The first synthetic image at the top right of figure 14 has been generated in 37 minutes using the hierarchical algorithm, from the left real photograph. Two specular surfaces have been recovered and simulated as non-perfect mirrors. Neither the isotropic nor anisotropic hypotheses have been tried thanks to the optimization technique described in section 6, and 14 rerendering iterations were necessary to generate the final image.

The inverse algorithm tooks 4 hours and 40 minutes to produce the image at the bottom right of figure 14. Roughly 4 hours of this time were necessary to recover the anisotropic BRDF of the aluminium surface. The final rendering stage took 32 minutes to render the final image (100 bounced rays have been used for the anisotropic surface).

The images of figure 15 show examples of applications in augmented reality. Some synthetic objects have been added such as a small robot and a luxo-like desk lamp. It is also possible to modify the reflectances easily too. New viewpoints can be generated and new illumination conditions can be created as well.

8 Conclusion and Future Work

In this paper, we have presented a new technique that approximates the reflectance properties of the surfaces of a 3D scene. An incremental and hierarchical algorithm iteratively estimates the diffuse, specular, isotropic and anisotropic reflectance parameters. In a final step, the textured surfaces are considered as a special case of reflectances to be simulated. The method takes as input a single photograph of the scene taken under known illumination conditions as well as a 3D geometric model of the scene. The result is a complete description of the photometric properties of the scene which may be used to produce a photorealistic synthetic image very similar to the real one. We showed that the method is robust and provides the opportunity to visualize the original scene from new angle, with any illumination conditions and with the addition, removal and modification of objects.

Our work has currently some limitations, especially regarding textured surfaces. Until now, we have not been able to discriminate the shadows or highlights from an assumed textured surface. In this regard, it will be interesting to extend our method to these cases, although we think that this is a very difficult problem, if one sticks to the single image assumption.

While many challenges remain, we believe that algorithms for recovering an approximation of the reflectances inside a real scene are an important direction of research for both Computer Vision and Computer Graphics communities. In Computer Vision, it could be possible for example to use our method to enhance the positioning of mirrors using a minimization algorithm between the real and the synthetic image. Regarding Computer Graphics, we may extend the reflectance recovery algorithm to objects that have more complex photometric properties such as light beams, small fires, caustics, etc. The hierarchical property of our technique offers many possible extensions.

Acknowledgments

The authors would like to thanks Jean-Marc Vézien for providing the 3D geometrical model and the camera calibration for the scenes shown in the results section. We also acknowledge him for his helpful scientific commentaries regarding the techniques described in this paper.

References

[1] Arthur Appel. Some techniques for shading machine renderings of solids. *AFIPS 1968 Spring Joint Computer Conf.*, 32:37–45, 1968.

[2] R. Baribeau, M. Rioux, and G. Godin. Color reflectance modeling using a polychromatic laser range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):263–269, February 1992.

[3] Loren Carpenter. The a-buffer, an antialiased hidden surface method. *Computer Graphics (Proceedings of SIGGRAPH 84)*, 18(3):103–108, July 1984. Held in Minneapolis, Minnesota.

[4] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. In John Dill, editor, *Computer Graphics (Proceedings of SIGGRAPH 88)*, volume 22, pages 75–84, August 1988.

[5] Michael F. Cohen and Donald P. Greenberg. The Hemi-Cube: A radiosity solution for complex environments. In B. A. Barsky, editor, *Computer Graphics (Proceedings of SIGGRAPH 85)*, volume 19, pages 31–40, August 1985.

[6] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, January 1999. ISSN 0730-0301.

[7] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In Michael Cohen, editor, *Computer Graphics (Proceedings of SIGGRAPH 98)*, Annual Conference Series, pages 189–198. Addison Wesley, July 1998.

[8] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In Turner Whitted, editor, *Computer Graphics (Proceedings of SIGGRAPH 97)*, Annual Conference Series, pages 369–378. Addison Wesley, August 1997.

[9] Paul Ernest Debevec. *Modeling and Rendering Architecture from Photographs*. PhD thesis, University of California, Berkeley, 1996.

[10] D.F DeMenthon and L. Davis. Model-based object pose in 25 lines of code. In *Second European Conference on Computer Vision(ECCV)*, pages 335–343. Springer-Verlag, May 1992.

[11] George Drettakis, Luc Robert, and Sylvain Bougnoux. Interactive common illumination for computer augmented reality. In Julie Dorsey and Philipp Slusallek, editors, *Eurographics Rendering Workshop 1997*, pages 45–56. Springer Wien, June 1997.

[12] Frédo Durand. *Visibilité tridimensionnelle: étude analytique et applications*. Ph.D. thesis, Université Joseph Fourier, Grenoble, France, July 1999.

[13] E. Fiume, A. Fournier, and L. Rudolph. A parallel scan conversion algorithm with anti-aliasing for a general purpose ultracomputer. *Computer Graphics (Proceedings of SIGGRAPH 83)*, 17(3):141–150, July 1983. Held in Detroit, Michigan.

[14] Alain Fournier, Atjeng S. Gunawan, and Chris Romanzin. Common illumination between real and computer generated scenes. In *Graphics Interface '93*, pages 254–262. Canadian Information Processing Society, May 1993. Held in Toronto, Ontario, Canada.

[15] Andrew Glassner. *An Introduction to Ray Tracing*. Academic Press, New York, NY, 1989.

[16] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. In Hank Christiansen, editor, *Computer Graphics (Proceedings of SIGGRAPH 84)*, volume 18, pages 213–222, July 1984.

[17] Press W. H., Teukolsky S.A., Vetterling W.T., and Flannery B.P. *Numerical Recipes in C, The Art of Scientific Computing*, chapter 10.4 Downhill Simplex Method in Multidimensions, pages 305–309. Cambridge University Press, Cambridge, 1992.

[18] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):197–206, July 1991.

[19] Nelder J.A. and Mead R. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[20] Konrad F. Karner, Heinz Mayer, and Michael Gervautz. An image based measurement system for anisotropic reflection. *Computer Graphics Forum*, 15(3):119–128, August 1996.

[21] G. Kay and T. Caelli. Inverting an illumination model from range and intensity maps. *CGVIP: Image Understanding*, 59:183–201, 1994.

[22] Eric P. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear approximation of reflectance functions. In *Computer Graphics (Proceedings of SIGGRAPH 97)*, volume 31, pages 117–126, 1997.

[23] C. Loscos, M. C. Frasson, G. Drettakis, B. Walter, X. Grainer, and P. Poulin. Interactive virtual relighting and remodeling of real scenes. Available from www.imagis.imag.fr/Publications RT-0230, Institut National de Recherche en Informatique en Automatique (INRIA), Grenoble, France, April 1999.

[24] Celine Loscos, George Drettakis, and Luc Robert. Interactive virtual relighting of real scenes. *IEEE Transactions on Visualization and Computer Graphics*, 6(3):289–305, 2000.

[25] J. Lu and J. Little. Reflectance function estimation and shape recovery from image sequence of rotating object. In *International Conference on Computer Vision*, pages 80–86, June 1995.

[26] Stephen R. Marschner and Donald P. Greenberg. Inverse lighting for photography. In *Proceedings of the Fifth Color Imaging Conference*. Society for Imaging Science and Technology, November 1997.

[27] Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, Kenneth E. Torrance, and Donald P.Greenberg. Image-based brdf measurement including human skin. In Dani Lischinski and Greg Ward Larson, editors, *Eurographics Rendering Workshop 1999*. Eurographics, June 1999.

[28] A. Rosenblum. *Data Visualization*, chapter Modeling Complex indoor scenes using an analysis/synthesis framework (André Gagalowicz). Academic Press, 1994.

[29] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Illumination distribution from brightness in shadows: Adaptive extimation of illumination distribution with unknown reflectance properties in shadow regions. In *Proceedings of IEEE ICCC'99*, pages 875–882, September 1999.

[30] Kosuke Sato and Katsushi Ikeuchi. Determining reflectance properties of an object using range and brightness images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1139–1153, 1991.

[31] Yoichi Sato and Katsushi Ikeuchi. Temporal-color space analysis of reflection. *Journal of Optical Society of America*, 11(11):2990–3002, November 1994.

[32] Yoichi Sato and Katsushi Ikeuchi. Reflectance analysis for 3d computer graphics model generation. *Graphical Models and Image Processing*, 58(5):437–451, 1996.

[33] Yoichi Sato, Mark D. Wheeler, and Katsushi Ikeuchi. Object shape and reflectance modeling from observation. In Turner Whitted, editor, *Computer Graphics (Proceedings of SIGGRAPH 97)*, pages 379–388. Addison Wesley, August 1997.

[34] Siu-Hang Or Tien-Tsin Wong, Pheng-Ann Heng and Wai-Yin Ng. Image-based rendering with controllable illumination. In Julie Dorsey and Phillip Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 13–22, New York, NY, 1997. Springer Wien. ISBN 3-211-83001-4.

[35] K.E. Torrance, E.M. Sparrow, and R.C. Birkebak. Polarization, directional distribution, and off-specular peak phenomena in light reflected from roughened surfaces. *Journal of Optical Society of America*, 56(7):916–925, 1966.

[36] Jack Tumblin and Holly Rushmeier. Tone reproduction for realistic images. *IEEE Computer Graphics and Applications*, 13(6):42–48, November 1993.

[37] Gregory J. Ward. Measuring and modeling anisotropic reflection. In Edwin E. Catmull, editor, *Computer Graphics (Proceedings of SIGGRAPH 92)*, volume 26, pages 265–272. ACM Press, July 1992.

[38] Gregory J. Ward. The radiance lighting simulation and rendering system. In Andrew Glassner, editor, *Computer Graphics (Proceedings of SIGGRAPH 94)*, Annual Conference Series, pages 459–472. ACM Press, July 1994. ISBN 0-89791-667-0. Held in Orlando, Florida.

[39] Hank Weghorst, Gary Hooper, and Donald P. Greenberg. Improved computational methods for ray tracing. *ACM Transactions on Graphics*, 3(1):52–69, January 1984.

[40] George Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, 1990.

[41] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse global illumination : Recovering reflectance models of real scenes from photographs. In A. Rockwood, editor, *Computer Graphics (Proceedings of SIGGRAPH 99)*, volume 19, pages 215–224. Addison Wesley Longman, August 1999.

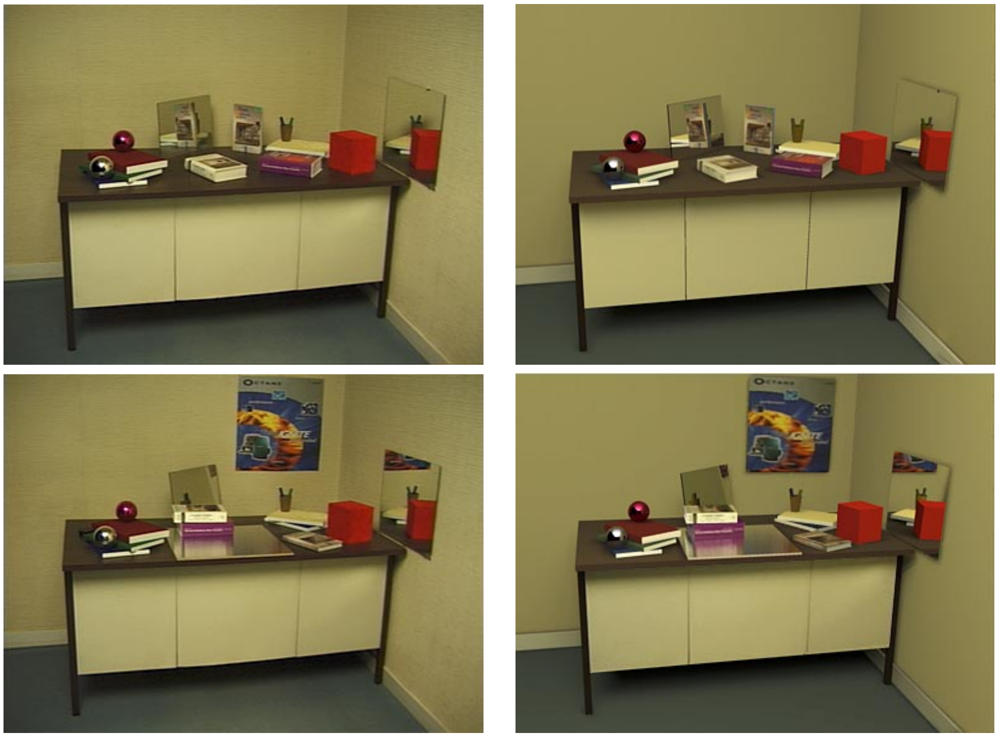


Figure 14: Two different examples of synthetic images (right) rerendered from a single real image (left). We remark that the perfectly diffuse assumption is realistic enough for many surfaces (including the walls, the floor, the desk, etc.).

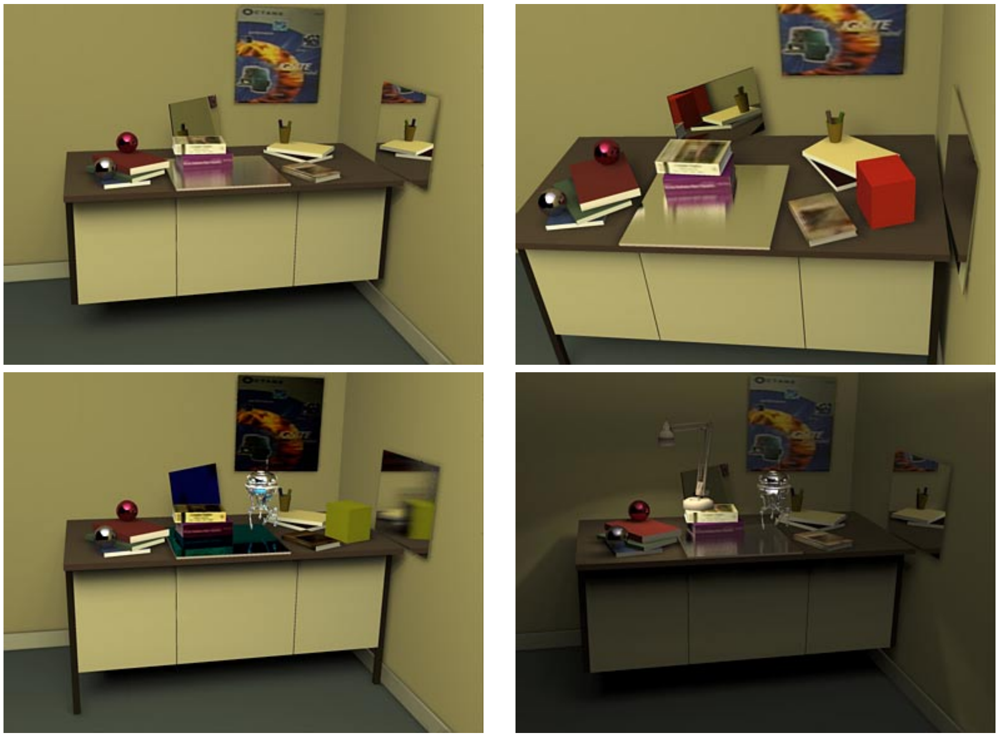


Figure 15: Examples of several augmented reality applications. All these new images were rendered using our global illumination software *Phoenix*, which first recovered the surface reflectances from the bottom left image of figure 14. The top left image shows the original scene removing some objects (the feet of the desk and the red cube). Note that the right mirror has taken into account the modification. The right top image shows the original scene rendered under a novel viewpoint. The bottom left image shows the scene with modified photometric properties, and the addition of an object (a small robot). The bottom right image presents the scene under novel illumination conditions with the addition and deletion of objects.

Practical Algorithms for Inverse Rendering under Complex Illumination

Ravi Ramamoorthi

Stanford University [*]

1 Introduction

In the last few years, there has been a significant amount of research in inverse rendering, which we are showcasing in this course. The methods of Debevec et al. [2], Marschner et al. [10], Sato et al. [16], and others have produced high quality measurements, leading to the creation of very realistic images. However, most previous work has been conducted in highly controlled lighting conditions, usually by careful active positioning of a single point source. Even methods that work in outdoor conditions, such as those of Yu and Malik [19], Sato and Ikeuchi [15] and Love [8], are designed specifically for natural illumination, and assume a simple parametric model for skylight.

The usefulness of inverse rendering would be greatly enhanced if it could be applied under general uncontrolled, and possibly unknown, lighting. For instance, this would allow for application in general unconstrained indoor or outdoor settings, or for estimation of BRDFs under unknown illumination. There are also a number of applications to human vision and perception. For instance, Dror et al. [3] have studied reflectance classification from a single image of a sphere under complex illumination to clarify how well the human visual system perceives materials, and to develop computational vision methods for the same task.

One reason there has previously been relatively little work in considering complex illumination is the lack of a common theoretical framework for determining under what conditions inverse problems can and cannot be solved, and for making principled approximations. Recently, we [12, 13] have developed a signal-processing framework for reflection on a curved surface, whereby the reflected light field can be viewed as a spherical convolution of the incident illumination and the BRDF. This framework can be used to determine the well-posedness of inverse problems, i.e. analyze which inverse problems can be solved, and to make appropriate approximations. In particular, *inverse rendering* may be viewed as *deconvolution*.

The purpose of these course notes is to provide a more detailed account of the practical representations and algorithms developed by us for inverse rendering under complex illumination, than in our earlier SIGGRAPH paper [13], which is also included in the course notes. Here, we assume the theoretical framework and focus in more detail on the practical algorithms for inverse rendering under complex illumination. Specifically, we present a more detailed version of the derivations and algorithms in sections 5.3 and 6 of the SIGGRAPH paper. That paper will be the point of reference for the background, theoretical analysis, and results, and we will frequently refer to the appropriate sections there, as required.

The rest of these course notes are organized as follows. In section 2, we summarize the complications that arise in practice, and the key theoretical ideas that are relevant for practical inverse rendering problems. In section 3, we use these ideas to derive a new low-parameter dual angular and frequency space representation applied in the next section. Section 4 presents our new practical algorithms in detail, and illustrates the concepts using spheres of different materials. Finally, section 5 concludes these course notes and discusses unsolved problems that may be directions for future work.

2 Preliminaries

The input to our algorithms consists of object geometry (acquired using a laser range scanner and a volumetric merging algorithm [1]) and photographs from a number of different viewing directions, with known extrinsic and intrinsic camera parameters. We assume static scenes, i.e. that the object remains stationary and the lighting remains the same between views. Our method is a passive-vision approach; we do not actively disturb the environment. We will also assume the illumination comes from distant sources, and is a function only of the global incident direction, which can be represented with an environment map. For simplicity, we will restrict ourselves to isotropic BRDFs and neglect the effects of interreflection. Our theoretical analysis also discounts self-shadowing for concave surfaces, although our practical algorithms will account for it where necessary. Our assumptions (known geometry, distant illumination, isotropic BRDFs and no interreflections) are commonly made in computer vision and interactive computer graphics.

Recently, we have developed a signal-processing framework [13] for reflection based on the assumptions outlined above, ignoring concavities and self-shadowing. Thus, the reflected light field can be expressed as a spherical convolution of the incident illumination and the BRDF, and written as a product of spherical harmonic coefficients of the lighting and BRDF. This allows us to view inverse rendering as *deconvolution*, or as a *factorization* of the reflected light field into the lighting and BRDF. Our theory leads to several new insights by reformulating reflection in the frequency domain. However, the frequency-space ideas must be put into practice carefully. This is analogous to practical implementation of the Fourier-space theory of aliasing. The ideal Fourier-space bandpass filter in the spatial domain, the sinc function, is usually modified for practical purposes because it has infinite extent and leads to ringing. Similarly, representing BRDFs purely as a linear combination of spherical harmonics leads to ringing. Moreover, in practical applications, the following complications often arise. Below, we list the important practical issues, followed by a discussion of the practical implications of the theory, and how we address these issues.

2.1 Practical Complications

Small number of photographs: Our signal-processing framework [13] allows us to derive analytic formulae for the lighting and BRDF spherical harmonic coefficients. However, these formulae require knowledge of all the reflected light field coefficients. In practical applications, we often want to work with a small number of photographs. Just as it is difficult to compute Fourier spectra from sparse irregularly sampled data, it is difficult to compute the reflected light field coefficients from a few photographs; we would require a very large number of input images, densely sampling the entire sphere of possible directions. Therefore, we use ideas from the theory to derive a simple practical low-parameter model of the reflected light field for the microfacet BRDF, that simultaneously uses the angular domain and the frequency domain.

Concave objects and self shadowing: Real geometry can often be complex with self shadowing due to concave regions. As discussed in the next subsection, we use associativity of convolution to blur the illumination and treat the specular BRDF component like a mirror. This allows us to simply check the reflected ray

[*] ravir@graphics.stanford.edu

for shadowing, which is straightforward in a raytracer.

Concavities (or multiple objects) also lead to interreflections. We have not yet addressed this issue, but we believe an iterative solution, similar to that used by Yu et al. [18] should be applicable.

Textured objects: Finally, our theoretical analysis does not account for spatial variation in the BRDF. Here, we make the common assumption that this spatial variation can be accounted for simply by texture-mapping the BRDF parameters, i.e. the diffuse and specular reflectivities.

2.2 Practical implications of theory

We now discuss a number of key insights and quantitative results obtained from the theory that influence our practical representations, which address the issues raised above.

Dual Angular and Frequency-Space Representations: Quantities local in angular space have broad frequency spectra and vice-versa. By developing a frequency-space view of reflection, we ensure that we can use either the angular-space or frequency-space representation, or even a combination of the two. The diffuse BRDF component is slowly varying in angular-space, but is local in frequency-space, while the specular BRDF component is local in the angular domain. For representing the lighting, the frequency-space view is appropriate for the diffuse BRDF component, while the angular-space view is appropriate for the specular component. In the next section, we derive a dual angular and frequency-space representation for the reflected light field from the microfacet BRDF model. A similar form can be derived for other common BRDFs like the Phong reflection model. This low-parameter representation may be used for inverse rendering when we have only a small number of photographs available, i.e. sparse sampling of the reflected light field.

Irradiance formula: For the Lambertian BRDF component, we have derived [12] a simple analytic formula, and have shown that the irradiance at all surface orientations can be approximated to within 1% using only 9 parameters, i.e. coefficients of spherical harmonics up to order 2. Thus, it makes sense to apply this simple formula where possible, representing the diffuse component of the reflected light field in the frequency domain.

Associativity of convolution: Because the coefficients of the reflected light field in the frequency domain are simply a product of the spherical harmonic coefficients of the incident illumination and the BRDF, we may apply the associativity of convolution. Thus, we can blur the illumination and sharpen the BRDF without changing the final results. In the extreme case, for specular models like the Phong BRDF, we may treat the BRDF as a mirror, while blurring the illumination, convolving it with the BRDF filter. Within the context of environment map rendering [4, 11], this is known as prefiltering. Besides increased efficiency, this approach also allows for very efficient approximate computation of cast shadows due to concave surfaces. One need simply check the reflected ray, as if the surface were a mirror, which is a simple operation in a raytracer.

Separation of slow and fast-varying lighting: In general, because the lighting and BRDF are not one-dimensional quantities, applying the associativity property above destroys the symmetries and reciprocity of the BRDF, so we cannot simply blur the illumination and treat the BRDF as a perfect mirror. However, for radially symmetric specular BRDFs, like the Phong model, where the BRDF depends only on the angle between the incident illumination and the reflection of the viewing direction about the surface normal, this is a valid operation. Therefore, we separate the illumination into slow and fast-varying components, corresponding to area sources and point sources. It can be shown that for low-frequency lighting, models like the microfacet BRDF (Torrance-Sparrow [17] model) behave much like a Phong model (the dominant term is Phong-like reflection), so that we may blur the illumination and treat the BRDF as a mirror. Furthermore, the largest errors in this approximation occur for grazing angles, where measurements are accorded low confidence in practical applications anyway. The fast-varying lighting components may be treated as point sources, which makes it easy to find angular-space formulae for the reflected light field.

It should be noted that the theoretical analysis is conducted without taking concavities into account. We will derive our representation in the next section under the convex-surface assumption. However, we will also show there how the representation can be simply extended to account for textured objects and cast shadows.

3 Dual angular and frequency-space representation

In a sense, the practical implications discussed above simply formalize a reflection model commonly used when rendering with environment maps. In that context, the BRDF is assumed to be a combination of Lambertian diffuse and Phong specular reflection. The reflected light is then the combination of a diffuse irradiance map due to the Lambertian BRDF component and a specular reflection map due to the specular Phong lobe. Our theoretical analysis allows for two practical improvements to be made. Firstly, the irradiance map can be represented using only 9 parameters in the frequency domain, which makes computations more efficient and compact. Secondly, we may use a single angular-space reflection map as a good approximation for the specular reflections, even for more complex physically-based BRDFs like the microfacet model [17], provided we first separate the lighting into slow and fast-varying components. This section is an expanded version of section 5.3 in our SIGGRAPH paper [13].

We will use a simplified Torrance-Sparrow [17] model, defined as follows.

$$\begin{aligned} \rho(\vec{\omega}_i, \vec{\omega}_o) &= K_d + K_s \frac{FS}{4\cos\theta_i \, \cos\theta_o} \\ \vec{\omega}_h &= \frac{\vec{\omega}_i + \vec{\omega}_o}{\parallel \vec{\omega}_i + \vec{\omega}_o \parallel} \\ F &= \frac{F(\mu, \theta_o)}{F(\mu, 0)} \\ S &= \frac{1}{\pi\sigma^2} \exp\left[-\left(\theta_h/\sigma\right)^2 \right] \end{aligned} \tag{1}$$

Here, ρ is the BRDF, and σ is the surface roughness parameter. The BRDF is a function of incoming and outgoing directions $(\vec{\omega_i}, \vec{\omega_o})$. The incident and outgoing angles with respect to the surface normal are denoted by θ_i and θ_o. Note that for clarity, we have dropped the primes as compared to the SIGGRAPH paper, since there is now no confusion regarding global or local coordinates. The subscript h stands for the half-way vector. $F(\mu, \theta_o)$ is the Fresnel term for refractive index μ; we normalize it to be 1 at normal exitance. Actually, F depends on the angle with respect to the half-way vector; in practice, this angle is usually very close to θ_o. For simplicity in the analysis, we have omitted the geometric attenuation factor G. In practice, this omission is not very significant except for observations made at grazing angles, which are usually assigned low confidence anyway in practical applications.

3.1 Model for reflected light field

Our model for the reflected light from the microfacet BRDF now includes three terms.

$$B = B_d + B_{s,slow} + B_{s,fast} \tag{2}$$

Here, B is the net reflected light field. The component because of the diffuse part in the BRDF is denoted B_d. $B_{s,slow}$ represents specularities from the slowly-varying lighting, and $B_{s,fast}$ specular highlights from the fast varying lighting component.

We may represent and compute B_d in the frequency domain by using the irradiance formula (which corresponds directly to the reflection from a Lambertian surface). We use the 9 parameter representation, explicitly noting the frequency $l \leq 2$.

$$\begin{aligned} B_d &= K_d E(\alpha,\beta) \\ E(\alpha,\beta) &= \sum_{l=0}^{2}\left(\hat{A}_l \sum_{m=-l}^{+l} L_{lm} Y_{lm}(\alpha,\beta)\right) \end{aligned} \tag{3}$$

Here, E is the irradiance, and K_d is the albedo or coefficient for diffuse reflection. The surface is parameterized by its orientation or surface normal in spherical coordinates (α,β). The spherical harmonics are denoted by Y_{lm}, and the spherical harmonic coefficients of the lighting by L_{lm}. The numerical values of $\hat{A}_l$ are given by

$$\hat{A}_0 = \pi \qquad \hat{A}_1 = 2\pi/3 \qquad \hat{A}_2 = \pi/4 \tag{4}$$

For $B_{s,slow}$, we filter the lighting, and treat the BRDF as a mirror. With $\vec{R}$ denoting the reflected direction, and L_{slow} the filtered version of the lighting, we obtain

$$B_{s,slow} = K_s F(\mu,\theta_o) L_{slow}(\vec{R}) \tag{5}$$

The filtered version of the illumination L_{slow} is obtained by multiplying the illumination coefficients by those of a filter corresponding to the term S in the microfacet BRDF of equation 1, i.e.

$$L_{lm}^{slow} = \exp\left[-(\sigma l)^2\right] L_{lm} \tag{6}$$

In the angular domain, this corresponds to convolving with a filter of angular width approximately σ^{-1}, or using a normalized Phong lobe with Phong exponent $\frac{1}{2}\sigma^{-2}$.

For the *fast varying* portion of the lighting—corresponding to sources of angular width $\ll \sigma$—we treat the total energy of the source, given by an integral over the (small) solid angle subtended, as located at its center, so the lighting is a delta function. $B_{s,fast}$ is given by the standard equation for the specular highlight from a directional source. The extra factor of $4\cos\theta_o$ in the denominator as compared to equation 5 comes from the relation between differential microfacet and global solid angles.

$$\begin{aligned} B_{s,fast} &= \frac{K_s F(\mu,\theta_o)}{4\cos\theta_o} \sum_j T_j(\sigma) \\ T_j(\sigma) &= \exp\left[-(\theta_h/\sigma)^2\right]\left(\frac{L_{j,fast}}{\pi\sigma^2}\right) \end{aligned} \tag{7}$$

The subscript j denotes a particular directional source; there could be several. Note that $L_{j,fast}$ is now the *total energy* of the source.

For BRDF estimation, it is convenient to expand out these equations, making dependence on the BRDF parameters explicit.

$$\begin{aligned} B &= K_d \sum_{l=0}^{2}\left(\hat{A}_l \sum_{m=-l}^{+l} L_{lm} Y_{lm}(\alpha,\beta)\right) \\ &+ K_s F(\mu,\theta_o)\left[L_{slow}(\vec{R}) + \frac{1}{4\cos\theta_o}\sum_j T_j(\sigma)\right] \end{aligned} \tag{8}$$

3.2 Textures and shadowing

We now show how to extend our representation to account for object textures and self-shadowing on complex concave geometry. The representation can be extended to textured surfaces simply by letting the BRDF parameters (such as K_d and K_s) be functions of surface location. It would appear that concave regions, where one part of the surface may shadow another, are a more serious problem since our theory is developed for convex objects and assumes no self-shadowing. However, in the remainder of this section, we will see that the extensions necessary mainly just involve checking for shadowing of the reflected ray and directional sources, which are routine operations in a raytracer.

We consider each of the three terms in our model of the reflected light field. In the presence of shadows, the 9 parameter model can no longer be used to directly compute B_d. Instead, the irradiance may be computed in the more conventional angular-space way by integrating the scene lighting while considering visibility. Alternatively, we can continue to use a spherical harmonic approximation, making use of the linearity of light transport. Note that the irradiance can still be written as a linear combination of lighting coefficients. Thus, we may replace equation 3 by

$$\begin{aligned} B_d &= K_d E(\vec{x}) \\ E(\vec{x}) &= \sum_{l=0}^{l_{max}} \sum_{m=-l}^{+l} L_{lm} \tilde{Y}_{lm}(\vec{x}) \end{aligned} \tag{9}$$

Here, we have increased the maximum frequency from 2 to l_{max}, where l_{max} can be larger than 2. Further, we have replaced the spherical harmonics with $\tilde{Y}_{lm}$. $\tilde{Y}_{lm}$ is the effect of the illumination spherical harmonic Y_{lm}. Since this effect now depends on the specific shadowing patterns, we have replaced the surface normal (α,β) with the position $\vec{x}$. For convex objects, as per equation 3, $\tilde{Y}_{lm}(\vec{x}) = \hat{A}_l Y_{lm}(\alpha,\beta)$.

For the specular components of the reflected light field, we simply check if the reflected ray (for the "slow" component) or the point sources (for the "fast" component) are shadowed. The main benefit is for slow specularities, where instead of a complex integration including visibility, the effects of shadowing are approximated simply by checking the reflected ray. It should be emphasized that in all cases, the corrections for visibility depend only on object geometry and viewing configuration (to determine the reflected direction), and can be precomputed for each point on the object using a ray tracer. Thus, we may replace equation 5 by

$$B_{s,slow} = K_s F(\mu,\theta_o) L_{slow}(\vec{R}) V(\vec{R}) \tag{10}$$

where V is a binary value specifying if the reflected ray is unshadowed. Similarly, a visibility term needs to multiply T_j in equation 7. Putting it all together, and including the effects of textures, by making the diffuse and specular reflectances function of position $\vec{x}$, equation 8 becomes

$$\begin{aligned} B &= K_d(\vec{x}) \sum_{l=0}^{l_{max}} \sum_{m=-l}^{+l} L_{lm} \tilde{Y}_{lm}(\vec{x}) \\ &+ K_s(\vec{x}) F(\mu,\theta_o)\left[V(\vec{R}) L_{slow}(\vec{R}) + \frac{1}{4\cos\theta_o}\sum_j V_j T_j(\sigma)\right] \end{aligned} \tag{11}$$

4 Algorithms

This section presents our practical algorithms for a broad range of inverse rendering problems under complex illumination, with simple illustrations using spheres of different materials. The objective is to describe some of our practical algorithms in greater detail than in sections 6.1-6.3 of our SIGGRAPH paper. The first part of section 6 in that paper contains details on image acquisition and experimental setup, which will not be duplicated here. Similarly, refer to section 6.4 of our SIGGRAPH paper for results using complex geometric and textured objects.

We describe two types of methods—algorithms that recover coefficients of a purely frequency-space description of the lighting or BRDF by representing these quantities as a sum of spherical harmonic terms, and algorithms that estimate parameters corresponding to our dual angular and frequency-space model of section 3. Section 4.1 on BRDF estimation discusses direct recovery of spherical harmonic BRDF coefficients, as well as estimation of parametric microfacet BRDFs using equations 8 and 11. Similarly, section 4.2 demonstrates direct recovery of spherical harmonic lighting coefficients, as well as estimation of a dual angular and frequency-space lighting description as per the model of section 3. Finally, section 4.3 shows how to combine BRDF and lighting estimation techniques to simultaneously recover the lighting and BRDF parameters, when both are unknown. In this case, we do not show direct recovery of spherical harmonic coefficients, as we have thus far found this to be impractical.

4.1 Inverse BRDF with known lighting

Estimation of Spherical Harmonic BRDF coefficients: Spherical harmonics and Zernike polynomials have been fit [7] to measured BRDF data, but previous work has not tried to estimate coefficients directly. Since the BRDF is linear in the coefficients $\hat{\rho}_{lpq}$, we simply solve a linear system to determine $\hat{\rho}_{lpq}$, to minimize the RMS error with respect to image observations[1]. It should be noted that in so doing, we are effectively interpolating (and extrapolating) the reflected light field to the entire 4D space, from a limited number of images.

Figure 1 compares the parametric BRDFs estimated under complex lighting to BRDFs measured using a single point source with the method of Marschner et al. [10]. As expected [7], the recovered BRDFs exhibit ringing. One way to reduce ringing is to attenuate high-frequency coefficients. According to our theory, this is equivalent to using low frequency lighting. Therefore, as seen in figure 1, images rendered with low-frequency lighting do not exhibit ringing and closely match real photographs, since only the low-frequency components of the BRDF are important. However, images rendered using directional sources show significant ringing.

For practical applications, it is usually more convenient to recover low-parameter BRDF models since these are compact, can be estimated from relatively fewer observations, and do not exhibit ringing. In the rest of this section, we will derive improved inverse rendering algorithms, assuming a parametric microfacet BRDF model.

Estimation of parametric BRDF model: We estimate BRDF parameters under general known lighting distributions using equation 8. The inputs are images that sample the reflected light field B. We perform the estimation using nested procedures. In the outer procedure, a simplex algorithm adjusts the nonlinear parameters μ and σ to minimize RMS error with respect to image pixels. In the inner procedure, a linear problem is solved for K_d and K_s. For numerical work, we use the simplex method `e04ccc` and linear solvers `f01qcc and f01qdc` in the NAG [5] C libraries. The main difference from previous work is that equation 8 provides a principled way of accounting for all components of the lighting and BRDF, allowing for the use of general illumination conditions.

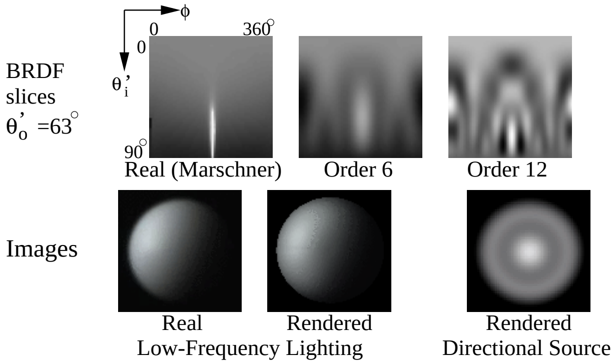


Figure 1: **Top:** *Slices of the BRDF transfer function of a teflon sphere for fixed exitant angle of* 63°. θ_o *varies linearly from* 0° *to* 90° *from top to bottom, and* $\mid \phi_o - \phi_i \mid$ *linearly from* 0° *to* 360° *from left to right. The central bright feature is the specular highlight. Left is the BRDF slice independently measured using the approach of Marschner et al. [10], middle is the recovered value using a maximum order 6, and right is the recovered version for order 12. Ringing is apparent in both recovered BRDFs. The right version is sharper, but exhibits more pronounced ringing.* **Bottom:** *Left is an actual photograph; the lighting is low-frequency from a large area source. Middle is a rendering using the recovered BRDF for order 6 and the same lighting. Since the lighting is low-frequency, only low-frequency components of the BRDF are important, and the rendering appears very similar to the photograph even though the recovered BRDF does not include frequencies higher than order 6. Right is a rendering with a directional source at the viewpoint, and exhibits ringing.*

We tested our algorithm on the spheres. Since the lighting includes high and low-frequency components (a directional source and an area source), the theory predicts that parameter estimation is well-conditioned. To validate our algorithm, we compared parameters recovered under complex lighting for one of the samples, a white teflon sphere, to those obtained by fitting to the full BRDF separately measured by us using the method of Marschner et al. [10]. Unlike most previous work on BRDF estimation, we consider the Fresnel term. It should be noted that accurate estimates for the refractive index μ require correct noise-free measurements at grazing angles. Since these measurements tend to be the most error-prone, there will be small errors in the estimated values of μ for some materials. Nevertheless, we find the Fresnel term important for reproducing accurate specular highlights at oblique angles. It should also be noted that while the results are quite accurate, there is still potential for future work on appropriate error metrics, especially for estimation of the roughness σ; a linear RMS error may not always be optimal.

Parameter	Our Method	Fit to Data
Reflectance	0.86	0.87
$K_d/(K_d+K_s)$	0.89	0.91
$K_s/(K_d+K_s)$	0.11	0.09
μ	1.78	1.85
σ	0.12	0.13
RMS	9.3%	8.5%

Figure 2: *Comparison of BRDF parameters recovered by our algorithm under complex lighting to those fit to measurements made by the method of Marschner et al. [10].*

The results in figure 2 show that the estimates of BRDF parameters from our method are quite accurate, and there is only a small

[1] Since the number of image pixels in a number of views can be very large, we randomly subsample the data for computational simplicity. We have used 12000 randomly selected image pixels.

increase in the error-of-fit when using parameters recovered by our algorithm to fit the measured BRDF. We also determined percentage RMS errors between images rendered using recovered BRDFs and real photographs to be between 5 and 10%. A visual comparison is shown in the first and third rows of figure 7. All these results indicate that, as expected theoretically, we can accurately estimate BRDFs even under complex lighting.

Textured objects with complex geometry: Handling concavities in complex geometric objects is not significantly more difficult, since we simply need to take visibility into account, and use equation 11 instead of equation 8. Equation 11 can also be used directly to estimate textured BRDFs. However, there are a number of subtle differences from direct BRDF estimation, which are noted below.

In considering textured surfaces, we essentially wish to consider each point on the surface separately, estimating a BRDF for each point independently from observations of that point alone. However, we now have only a few observations for each point (the number of images used). If there were no image noise, and our simplified four parameter microfacet model were a perfectly accurate description of the surface, this would still be sufficient. However, in practice, we are not able to reliably estimate the nonlinear parameters from such sparse data. This is true even for point source illumination, and has been observed by many authors. In our case, since we have complex illumination, the problem is even harder. Therefore, like much previous work, we assume the nonlinear parameters σ and μ are constant across the surface. A weaker assumption would be to allow them to vary slowly, or break the surface into regions of constant μ and σ.

Therefore, we will solve for the global nonlinear parameters σ and μ, as well as the diffuse and specular textures, $K_d(\vec{x})$ and $K_s(\vec{x})$. The corresponding radiance values for each image observation can be written as

$$B = K_d(\vec{x})D + K_s(\vec{x})S(\mu, \sigma) \tag{12}$$

where D and S stand for the diffuse and specular components computed from equation 11. These depend only on the lighting and viewing configuration, and S also depends on the nonlinear parameters μ and σ. It should be noted that much previous work has assumed constant values for the specular coefficient. The reason is that specularities are not usually observed over the whole object surface. By using complex illumination, we alleviate this problem somewhat, since large regions of the object can exhibit specularity in a single image. Nevertheless, there might be dimly lit regions or places where no specularities are observed in a sequence of views, and we will not be able to estimate coefficients in these regions. Therefore, we introduce confidence measures to enscapsulate the importance of each observation.

$$\begin{aligned} W_d &= \frac{D \cos\theta_o}{\epsilon + S} \\ W_s &= S \cos\theta_o \end{aligned} \tag{13}$$

Here, W_d and W_s are the confidence parameters for diffuse and specular reflection respectively. The multiplication by $\cos\theta_o$ is to give less weight to observations made at grazing exitant angles. ϵ is a small constant to avoid divisions by 0. In the diffuse weight W_d, we give greater importance to well illuminated pixels (high values of D) without too much specularity. In the specular weight W_s, we give importance to pixels observing strong specular reflections S.

Parameter estimation now proceeds much as BRDF estimation for untextured surfaces. Initially, we solve for values of the nonlinear parameters μ and σ using a simplex algorithm (outer loop). To account for regions where specularity is not strongly observed, in this phase, we include K_s as a global parameter to be solved for. In the inner loop of the procedure, we solve (at each point separately) for $K_d(\vec{x})$ to minimize the RMS error over all views. The output from this stage are parameters K_s, μ and σ, as well as an initial estimate of the diffuse texture $K_d(\vec{x})$. We use these global values of μ and σ. The global estimated value of K_s will be used in regions where a better estimate is not possible, but will in general be refined. In this first pass of the algorithm, we weight each observation using the confidence weight W_d.

We then use an iterative scheme to refine the estimates of K_s and K_d. While we could simply solve a linear system, corresponding to equation 12, for each vertex on the object, we have obtained better results using an iterative scheme, alternatively solving for $K_d(\vec{x})$ and $K_s(\vec{x})$ while keeping the other fixed. Since we use the dielectric model, K_s has no color, and we recover 4 linear texture parameters for each pixel (a diffuse RGB color and a specular coefficient). It should be noted that different confidence weights (W_d or W_s) are used in the iteration, depending on whether we are estimating the diffuse or specular component of the texture. We start by using a constant value of K_s, and the corresponding value of $K_d(\vec{x})$ recovered in the first phase, where we solved for μ and σ. We then hold $K_d(\vec{x})$ fixed and solve for $K_s(\vec{x})$. Thereafter, we hold $K_s(\vec{x})$ fixed and solve for $K_d(\vec{x})$, and repeat this process till convergence to the desired tolerance, which usually takes a few iterations.

There can of course be cases where $\sum W_s$ or $\sum W_d$ (the summation is over all views of that point) are too low (numerically zero) to accurately estimate specular or diffuse textures respectively. This corresponds to not observing specularities (when $\sum W_s$ is close to 0), or having the point being so dimly lit that the texture isn't discernible (when $\sum W_d$ is close to 0). In the former case, we simply use the mean value of the specular texture, while in the latter case, we mark the diffuse texture estimate as unreliable. It should be noted that using complex illumination greatly reduces the number of points where this is an issue, since much more of the object receives illumination and exhibits specularities than with a point light source.

4.2 Inverse Lighting with Known BRDF

Previous methods for estimating the lighting have been developed only for the special cases of mirror BRDFs (a gazing sphere), Lambertian BRDFs (Marschner and Greenberg [9]), and when shadows are present (Sato et al. [14]). Previous methods [9, 14] have also required regularization using penalty terms with user-specified weights, and have been limited by the computational complexity of their formulations to a coarse discretization of the sphere. We present two new algorithms for curved surfaces with general BRDFs. The first method directly recovers spherical harmonic lighting coefficients L_{lm}. The second algorithm estimates parameters of the dual angular and frequency-space lighting model of section 2. This method requires no explicit regularization, and yields high-resolution results that are sharper than those from the first algorithm, but is more difficult to extend to concave surfaces.

The theory tells us that inverse lighting is ill-conditioned for high-frequencies. Therefore, we will recover only low-frequency continuous lighting distributions, and will not explicitly account for directional sources, i.e. we assume that $B_{s,fast} = 0$. The reflected light field is essentially independent of the surface roughness σ under these conditions, so our algorithms do not explicitly use σ. The theory predicts that the recovered illumination will be a filtered version of the real lighting. Directional sources will appear as filtered into continuous distributions of angular width approximately σ.

Estimation of Spherical Harmonic Lighting coefficients: We may represent the lighting entirely in frequency-space by coefficients L_{lm} with $l \leq l^*$, and solve a linear least-squares system for L_{lm}. The first term in parentheses below corresponds to B_d, and the second to $B_{s,slow}$. The cutoff l^* is used for regularization, and should be of order $l^* \sim \sigma^{-1}$. Since most materials have $\sigma \sim .1$, we use $l^* = 12$.

$$B = \sum_{l=0}^{l^*} \sum_{m=-l}^{l} L_{lm} \left(K_d \hat{A}_l Y_{lm}(\alpha, \beta) + K_s F Y_{lm}(\theta_R, \phi_R) \right) \tag{14}$$

To extend this to concave surfaces, we simply need to add terms corresponding to visibility and shadowing, following equation 11, but the problem remains a linear system.

$$B = \sum_{l=0}^{l^*} \sum_{m=-l}^{l} L_{lm} \left(K_d \tilde{Y}_{lm}(\vec{x}) + K_s F V(\theta_R, \phi_R) Y_{lm}(\theta_R, \phi_R) \right) \tag{15}$$

Estimation of Parametric Dual Lighting Model: Another approach is to estimate the dual angular and frequency-space lighting model of section 3. Our algorithm is based on subtracting out the diffuse component B_d of the reflected light field. After this, we treat the object as a mirror sphere, recovering a high-resolution angular-space version of the illumination from the specular component alone. To determine B_d, we need only the 9 lowest frequency-space coefficients L_{lm} with $l \leq 2$. Our algorithm uses the following methods to convert between angular and frequency-space:

1. **9 parameters to High-Resolution Lighting:** The inputs to phase 1 are the coefficients L^1_{lm}. These suffice to find B^1_d by equation 3. Since we assumed that $B_{s,fast} = 0$,

   $$\begin{aligned} B_{s,slow} &= K_s F(\mu, \theta_o) L_{slow}(\vec{R}) \; = B - B^1_d(L^1_{lm}) \\ L_{slow}(\vec{R}) &= \frac{B - B^1_d(L^1_{lm})}{K_s F(\mu, \theta_o)} \end{aligned}$$

   We assume the BRDF parameters are known, and B is the input to the algorithm, so the right-hand side can be evaluated.

   In practice, we will have several observations corresponding to the reflected direction, and these will be weighted by the appropriate confidence and combined. For simplicity, the rest of the mathematical discussion will assume without loss of generality, that there is a single image observation for each reflected direction.

2. **High-Resolution Lighting to 9 parameters:** Using the angular space values L found from the first phase, we can easily find the 9 frequency-space parameters of the lighting L^2_{lm}.

Now, assume we run phase 1 (with inputs L^1_{lm}) and phase 2 (with outputs L^2_{lm}) sequentially. The *consistency condition* is that $L^1_{lm} = L^2_{lm}$—converting from frequency to angular to frequency space must not change the result. Equivalently, the computed diffuse components must match, i.e. $B^1_d(L^1_{lm}) = B^2_d(L^2_{lm})$. This is illustrated in figure 3. Since everything is linear in terms of the lighting coefficients, the consistency condition reduces to a system of 9 simultaneous equations. After solving for L_{lm}, we run phase 1 to determine the high-resolution lighting in angular space.

More formally, phase 1 can be written as a linear system in terms of constants U and W_{lm}, with (α, β) the coordinates of the surface normal.

$$\begin{aligned} L_{slow}(\theta_R, \phi_R) &= U(\theta_R, \phi_R) - \sum_{l=0}^{2} \sum_{m=-l}^{l} W_{lm}(\theta_R, \phi_R) L^1_{lm} \\ U(\theta_R, \phi_R) &= \frac{B}{K_s F(\mu, \theta_o)} \\ W_{lm}(\theta_R, \phi_R) &= \frac{K_d \hat{A}_l Y_{lm}(\alpha, \beta)}{K_s F(\mu, \theta_o)} \end{aligned} \tag{16}$$

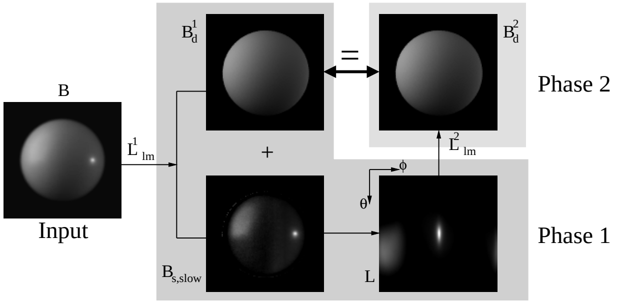


Figure 3: *Estimation of dual lighting representation. In phase 1, we use frequency-space parameters L^1_{lm} to compute diffuse component B^1_d. This is subtracted from the input image, leaving the specular component, from which the angular-space lighting is found. In phase 2, we compute coefficients L^2_{lm}, which can be used to determine B^2_d. The consistency condition is that $B^1_d = B^2_d$ or $L^1_{lm} = L^2_{lm}$. In this and all subsequent figures, the lighting is visualized by unwrapping the sphere so θ ranges in equal increments from 0 to π from top to bottom, and ϕ ranges in equal increments from 0 to 2π from left to right (so the image wraps around in the horizontal direction).*

Phase 2 to compute the lighting coefficients can also be written as a linear expression in terms of all the (discretized) reflected directions.

$$L^2_{lm} = \frac{2\pi^2}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sin\theta_i \, L_{slow}(\theta_i, \phi_j) Y^*_{lm}(\theta_i, \phi_j) \tag{17}$$

Here, N is the angular resolution, with the summation being a discrete version of the integral to find lighting coefficients.

But, the summation on the right hand side can be written in terms of lighting coefficients L^1_{lm}, simply by plugging in the formula for L_{slow}. We now obtain,

$$L^2_{lm} = \frac{2\pi^2}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sin\theta_i \left(U(\theta_i, \phi_j) - \sum_{l',m'} W_{l'm'}(\theta_i, \phi_j) L^1_{l'm'} \right) Y^*_{lm}(\theta_i, \phi_j) \tag{18}$$

Mathematically, the consistency condition allows us to drop the superscripts, reducing the above to a linear system for L_{lm}. This will involve a simple 9×9 linear system expressed in terms of a matrix $Q_{l'm',lm}$.

$$\begin{aligned} \sum_{l'=0}^{2} \sum_{m'=-l'}^{l'} Q_{lm,l'm'} L_{l'm'} &= P_{lm} \qquad (19) \\ P_{lm} &= \frac{2\pi^2}{N^2} \sum_{i,j} \sin\theta_i \, U(\theta_i, \phi_j) Y^*_{lm}(\theta_i, \phi_j) \\ Q_{lm,l',m'} &= \delta_{lm,l'm'} + \frac{2\pi^2}{N^2} \sum_{i,j} \sin\theta_i \, W_{l'm'}(\theta_i, \phi_j) Y^*_{lm}(\theta_i, \phi_j) \end{aligned}$$

The summations are just discrete versions of integrals that determine the appropriate spherical harmonic coefficients. The above equation has a very intuitive explanation. It may be derived direction from equation 8, considering the linear system that results for the first 9 lighting terms. The key idea is that we have reparameterized by the reflection vector, so we may simply take the first 9 coefficients of the reflected light field. The formula for the irradiance becomes more complicated (because of the reparameterization) but can still be expressed in terms of the first 9 lighting coefficients. Mathematically, we can rewrite equation 8 for our purposes as

$$\begin{aligned} B(\theta_R, \phi_R) &= K_d \sum_{l'=0}^{2} \sum_{m'=-l'}^{l'} \hat{A}_{l'} L_{l'm'} Y_{l'm'}(\alpha, \beta) \\ &+ K_s F(\mu, \theta_o) L_{slow}(\theta_R, \phi_R) \end{aligned} \tag{20}$$

Here, we have simply parameterized the reflected light field by the reflected direction (θ_R, ϕ_R). Remember that for simplicity, we're assuming a single image, i.e. one value of (α, β) corresponding to each (θ_R, ϕ_R), with (α, β) a function of (θ_R, ϕ_R). With multiple images, we would have to weight contributions appropriately.

Now, it's a simple enough matter to compute coefficients obtaining

$$B_{lm} = K_d \sum_{l',m'} \hat{A}_{l'} L_{l'm'} \left\langle Y^*_{lm}(\theta_R, \phi_R), Y_{l'm'}(\alpha, \beta) \right\rangle + K_s F(\mu, \theta_o) L_{lm} \tag{21}$$

Here, we have used the notation $< \cdot, \cdot >$ for the integral or inner product over the spherical domain of integration. This is what is computed discretely in equation 19. It can now be seen that equation 21 has the same form as equation 19. Note that in equation 21, we have multiplied out the denominators, and we use B_{lm} here instead of P_{lm}.

This inverse lighting method is difficult to extend to concave surfaces, since the 9 parameter diffuse model is no longer entirely valid. It is a subject of future work to see if it can be applied simply by increasing the number of parameters and the size of the matrix of simultaneous equations to be solved.

Positive regularization: So far, we have not explicitly tried to ensure positivity of the illumination. In practical applications, the methods above when applied naively will result in negative values, especially where the illumination is dark, and there is uncertainty about the precise value. Regularizing so the results are positive can also substantially improve the quality of the results by reducing high-frequency noise centered close to the zero point in dark regions.

We apply positive regularization to the unregularized solution from either of the previous two methods. For the first method (direct solution of linear system to determine lighting coefficients), we simply add another term to the RMS error which penalizes negative regions. While this method is a soft constraint and can still leave some negative regions, we have observed that it works well in practice. We use a conjugate gradient method to minimize

$$\nu' = \nu + \lambda \left(\frac{\nu_0}{\kappa_0} \right) \kappa \tag{22}$$

Here, ν is the RMS error corresponding to equation 14 or 15. κ is a new penalty term added to penalize negative values of the lighting. ν_0 and κ_0 are initial values (for the unregularized solution), and λ weights the importance of the penalty term. λ is a dimensionless quantity, and we have found experimentally that $\lambda = 1$ works well. The penalty term κ is simply the sum of squares of all lighting pixels having negative values. Thus, negative values are penalized, but no penalty is imposed for positive values.

For our second method (using a dual angular and frequency-space method to estimate the lighting), regularization may be enforced (in step 1) simply by clamping L_{slow} to 0 if the right hand side in the first line of equation 16 is negative. This must be taken into account in the 9×9 simultaneous equations, and we solve the *positivity enforced* equations with a conjugate gradient method, using as a starting guess the solution without enforced positivity.

Comparison: Figure 4 compares the methods to each other, and to a reference solution from a gazing sphere. Both algorithms give reasonably accurate results. As predicted by the theory, high-frequency components are filtered by the roughness σ. In the first method, involving direct recovery of L_{lm}, there will still be some residual energy for $l > l^*$. Since we regularize by not considering higher frequencies—we could increase l^*, but this makes the result noisier—the recovered lighting is somewhat blurred compared to our dual angular and frequency-space algorithm (second method). As expected, positive regularization in algorithm 2 results in a smoother solution.

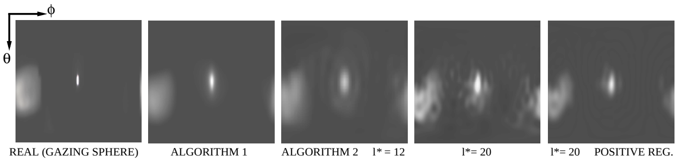


Figure 4: *Comparison of inverse lighting methods. From left to right, real lighting (from a gazing sphere), recovered illumination by direct estimation of spherical harmonic coefficients with* $l^* = 12$ *and* $l^* = 20$*, and estimation of dual angular and frequency-space lighting model. To make the artifacts more apparent, we have set* 0 *to gray. The results from the dual algorithm are sharper, but still somewhat blurred because of filtering by* σ*. A small amount of ringing occurs for direct coefficient recovery, and can be seen for* $l^* = 12$*. Using* $l^* = 20$ *makes the solution very noisy. Positive regularization (rightmost) gives a smoother solution.*

4.3 Factorization—Unknown Lighting and BRDF

We can combine the inverse-BRDF and inverse-lighting methods to *factor* the reflected light field, simultaneously recovering the lighting and BRDF when both are unknown. Therefore, we are able to accurately recover BRDFs of curved surfaces under unknown complex illumination, something which has not previously been demonstrated. There is an unrecoverable global scale factor, so we set $K_d + K_s = 1$; we cannot find absolute reflectance. Also, the theory predicts that for low-frequency lighting, estimation of the surface roughness σ is ill-conditioned—blurring the lighting while sharpening the BRDF does not significantly change the reflected light field. However, for high-frequency lighting, this ambiguity can be removed. We will use a single manually specified directional source in the recovered lighting distribution to estimate σ.

Algorithm: The algorithm consists of nested procedures. In the outer loop, we effectively solve an inverse-BRDF problem—a nonlinear simplex algorithm adjusts the BRDF parameters to minimize error with respect to image pixels. Since $K_d + K_s = 1$, and σ will not be solved for till after the lighting and other BRDF parameters have been recovered, there are only 2 free parameters, K_s and μ. In the inner procedure, a linear problem is solved to estimate the lighting for a given set of BRDF parameters, using the methods of the previous subsection. Pseudocode is given below.

```
global $B_{input}$                                // Input images
global $K_d$,$K_s$,$\mu$,$\sigma$                 // BRDF parameters
global $L$                                        // Lighting
procedure Factor
    Minimize($K_s$,$\mu$,ObjFun)                  // Simplex Method
    $\sigma$ = FindRoughness($L$)                 // Figure 5, Equation 23
function ObjFun($K_s$,$\mu$)
    $K_d = 1 - K_s$                               // $K_d + K_s = 1$
    $L$ = Lighting($K_d$,$K_s$,$\mu$)             // Inverse Lighting
    $B_{pred}$ = Predict($L$,$K_d$,$K_s$,$\mu$)   // Predicted Light Field
    return RMS($B_{input}$,$B_{pred}$)            // RMS Error
```

Finding σ using a directional source: If a directional source is present—and manually specified by us in the recovered lighting—we can estimate σ by equating specular components predicted by equations 5 and 7 for the center, i.e. brightest point, of the light source at normal exitance. An illustration is in figure 5.

$$L_{cen} \approx \frac{L_{total}}{4\pi\sigma^2} \tag{23}$$

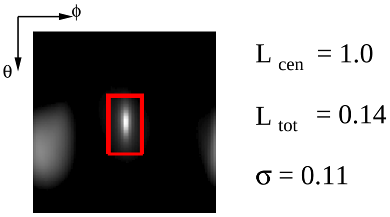


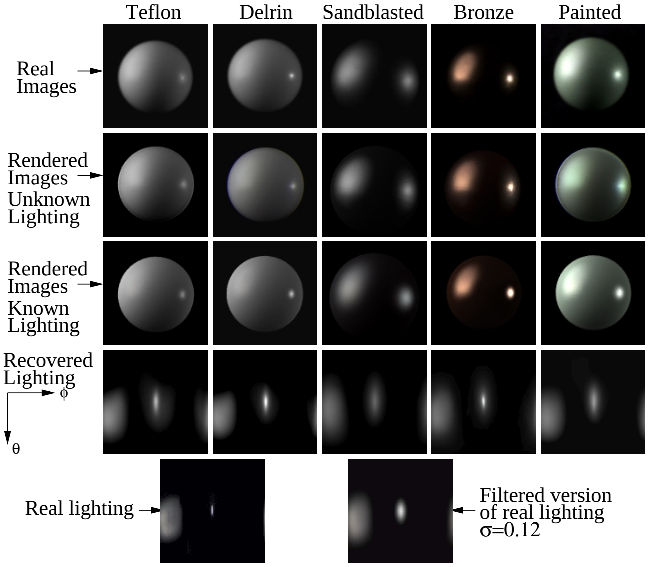
Figure 5: *We manually specify (red box) the region corresponding to the directional source in a visualization of the lighting. The algorithm then determines L_{cen}, the intensity at the center (brightest point), L_{tot}, the total energy integrated over the region specified by the red box, and computes σ using equation 23. The method does not depend on the size of the red box—provided it encloses the entire (filtered) source—nor the precise shape into which the source is filtered in the recovered lighting.*

Color: We have so far ignored issues of color, assuming the three color channels are considered separately. However, in the case of BRDF recovery under unknown lighting, there is a separate scale factor associated with each color channel. In order to obtain accurate colors for the BRDF and lighting components, we need some way to relate these 3 scale factors. For dielectrics, the specular component K_s is not spectrally sensitive, i.e. it is the same for red, green, and blue channels. The recovered BRDFs are scaled in order to make this hold. The issue is trickier for metals. There is a fundamental ambiguity between the color of the BRDF and the color of the lighting. We resolve this by considering the average color of the metallic surface as corresponding to white light. The use of more sophisticated color-space separation methods such as that of Klinker et al [6] might bring further benefits.

Results: We used the method of this subsection—with the dual angular and frequency-space algorithm for inverse lighting—to factor the light field for the spheres, simultaneously estimating the BRDF and lighting. The same setup and lighting were used for all the spheres so we could compare the recovered illumination distributions.

We see from figure 6 that the BRDF estimates under unknown lighting are accurate. Absolute errors are small, compared to parameters recovered under known lighting. The only significant anomalies are the slightly low values for the refractive index μ—caused because we cannot know the high-frequency lighting components, which are necessary for more accurately estimating the Fresnel term. We are also able to estimate a filtered version of the lighting. As shown in figure 7, the recovered lighting distributions from all the samples are largely consistent. As predicted by the theory, the directional source is spread out to different extents depending on how rough the surface is, i.e. the value of σ. Finally, figure 7 shows that rendered images using the estimated lighting and BRDF are almost indistinguishable from real photographs.

This section has been a detailed presentation of our practical algorithms for inverse rendering under complex illumination, using spheres of different materials to illustrate the concepts. As mentioned in the introduction to this section, our results using complex geometric models and textured objects are found in section 6.4 of our SIGGRAPH paper [13], to which we refer the reader for examples of images created by applying our algorithms to real-world objects under complex illumination conditions.



Teflon
Delrin
Sandblasted
Bronze
Painted
Real Images
Rendered Images Unknown Lighting
Rendered Images Known Lighting
Recovered Lighting
φ
θ
Real lighting
Filtered version of real lighting σ=0.12

Figure 7: *Spheres rendered using BRDFs estimated under known (section 4.1) and unknown (section 4.3) lighting. The algorithm in section 4.3 also recovers the lighting. Since there is an unknown global scale, we scale the recovered lighting distributions in order to compare them. The recovered illumination is largely consistent between all samples, and is similar to a filtered version of the real lighting. As predicted by the theory, the different roughnesses σ cause the directional source to be spread out to different extents. The filtered source is slightly elongated or asymmetric because the microfacet BRDF is not completely symmetric about the reflection vector.*

5 Conclusions and Future Work

We have considered the problem of complex illumination in inverse rendering. The use of realistic lighting and materials is likely to be of increasing importance in graphics and vision, and inverse rendering to acquire real-world material properties is likely to be a significant future direction. The use of natural lighting is becoming increasingly common in computer graphics and vision, and the techniques presented here allow for inverse rendering to be applied in arbitrary uncontrolled conditions rather than a laboratory setting. Furthermore, in certain cases, complex illumination may help in solving inverse problems, such as by allowing a much larger fraction of an image to exhibit specularity. Dror et al. [3] have also shown that people perceive reflectance properties much more easily under natural illumination.

However, this is only the first step in solving inverse problems under complex illumination. There are a number of open problems that remain subjects for future work. There has been relatively little work on BRDF representations in between low-parameter models and full measured representations. It is not clear what the best way to measure factored or separable BRDF representations is, or how to estimate these or higher-dimensional representations under complex illumination. We have not considered statistical properties of the illumination, that may allow us to simultaneously determine the lighting, BRDF and texture. All of our examples use a small number of images, and an interesting future direction is whether

Material	K_d		K_s		μ		σ	
	Known	Unknown	Known	Unknown	Known	Unknown	Known	Unknown
Teflon	0.89	0.87	0.11	0.13	1.78	1.48	0.12	0.14
Delrin	0.87	0.88	0.13	0.12	1.44	1.35	0.10	0.11
Neoprene Rubber	0.92	0.93	0.08	0.07	1.49	1.34	0.10	0.10
Sandblasted Steel	0.20	0.14	0.80	0.86			0.20	0.19
Bronze	(.15,.08,.05)	(.09,.07,.07)	(.85,.68,.59)	(.91,.69,.55)			0.12	0.10
Painted	(.62,.71,.62)	(.67,.75,.64)	0.29	0.25	1.38	1.15	0.15	0.15

Figure 6: *BRDFs of various spheres, recovered under known (section 4.1) and unknown (section 4.3) lighting. The reported values are normalized so $K_d + K_s = 1$. RGB values are reported for colored objects. We see that K_s is much higher for the more specular metallic spheres, and that σ is especially high for the rough sandblasted sphere. The Fresnel effect is very close to 1 for metals, so we do not consider the Fresnel term for these spheres.*

a single image suffices. We also believe there are more insights to be obtained from frequency space analysis and new frequency domain algorithms to be explored for inverse problems. Another interesting future direction is to preserve and extrapolate the original images to new conditions, using inverse rendering only as a guide in this extrapolation, but without explicitly computing lighting and reflectance parameters. This would have the potential to unify inverse and image-based (interpolatory) rendering methods.

References

[1] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH 96*, pages 303–312, 1996.

[2] P. Debevec, T. Hawkins, C. Tchou, H.P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *SIGGRAPH 00*, pages 145–156.

[3] R. Dror, E. Adelson, and A. Willsky. Estimating surface reflectance properties from images under unknown illumination. In *SPIE Photonics West: Human Vision and Electronic Imaging VI*, pages 231–242, 2001.

[4] N. Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics & Applications*, 6(11):21–29, 1986.

[5] Numerical Algorithms Group. *NAG C Library Manual, Mark 5*. 1999.

[6] G.J. Klinker, S.A. Shafer, and T. Kanade. The measurement of highlights in color images. *IJCV*, 2(1):7–32, 1988.

[7] J. J. Koenderink and A. J. van Doorn. Phenomenological description of bidirectional surface reflection. *JOSA A*, 15(11):2903–2912, 1998.

[8] R. Love. *Surface Reflection Model Estimation from Naturally Illuminated Image Sequences*. PhD thesis, Leeds, 1997.

[9] S.R. Marschner and D.P. Greenberg. Inverse lighting for photography. In *Fifth Color Imaging Conference*, pages 262–265, 1997.

[10] S.R. Marschner, S.H. Westin, E.P.F. Lafortune, and K.E. Torrance. Image-Based BRDF measurement. *Applied Optics*, 39(16):2592–2600, 2000.

[11] G. Miller and C. Hoffman. Illumination and reflection maps: Simulated objects in simulated and real environments. *SIGGRAPH 84 Advanced Computer Graphics Animation seminar notes*, 1984.

[12] R. Ramamoorthi and P. Hanrahan. On the relationship between radiance and irradiance: Determining the illumination from images of a convex lambertian object. *JOSA A*, 18(10):2448–2459, 2001.

[13] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. In *SIGGRAPH 01*, pages 117–128, 2001.

[14] I. Sato, Y. Sato, and K. Ikeuchi. Illumination distribution from brightness in shadows: adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. In *ICCV 99*, pages 875 – 882, 1999.

[15] Y. Sato and K. Ikeuchi. Reflectance analysis under solar illumination. Technical Report CMU-CS-94-221, CMU, 1994.

[16] Y. Sato, M. D. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. In *SIGGRAPH 97*, pages 379–388, 1997.

[17] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *JOSA*, 57(9):1105–1114, 1967.

[18] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *SIGGRAPH 99*, pages 215–224, 1999.

[19] Y. Yu and J. Malik. Recovering photometric properties of architectural scenes from photographs. In *SIGGRAPH 98*, pages 207–218, 1998.

[20] Z. Zhang. A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research, 1998.

Practical Algorithms for Inverse Rendering under Complex Illumination

Ravi Ramamoorthi

Stanford University [*]

1 Introduction

In the last few years, there has been a significant amount of research in inverse rendering, which we are showcasing in this course. The methods of Debevec et al. [2], Marschner et al. [10], Sato et al. [16], and others have produced high quality measurements, leading to the creation of very realistic images. However, most previous work has been conducted in highly controlled lighting conditions, usually by careful active positioning of a single point source. Even methods that work in outdoor conditions, such as those of Yu and Malik [19], Sato and Ikeuchi [15] and Love [8], are designed specifically for natural illumination, and assume a simple parametric model for skylight.

The usefulness of inverse rendering would be greatly enhanced if it could be applied under general uncontrolled, and possibly unknown, lighting. For instance, this would allow for application in general unconstrained indoor or outdoor settings, or for estimation of BRDFs under unknown illumination. There are also a number of applications to human vision and perception. For instance, Dror et al. [3] have studied reflectance classification from a single image of a sphere under complex illumination to clarify how well the human visual system perceives materials, and to develop computational vision methods for the same task.

One reason there has previously been relatively little work in considering complex illumination is the lack of a common theoretical framework for determining under what conditions inverse problems can and cannot be solved, and for making principled approximations. Recently, we [12, 13] have developed a signal-processing framework for reflection on a curved surface, whereby the reflected light field can be viewed as a spherical convolution of the incident illumination and the BRDF. This framework can be used to determine the well-posedness of inverse problems, i.e. analyze which inverse problems can be solved, and to make appropriate approximations. In particular, *inverse rendering* may be viewed as *deconvolution*.

The purpose of these course notes is to provide a more detailed account of the practical representations and algorithms developed by us for inverse rendering under complex illumination, than in our earlier SIGGRAPH paper [13], which is also included in the course notes. Here, we assume the theoretical framework and focus in more detail on the practical algorithms for inverse rendering under complex illumination. Specifically, we present a more detailed version of the derivations and algorithms in sections 5.3 and 6 of the SIGGRAPH paper. That paper will be the point of reference for the background, theoretical analysis, and results, and we will frequently refer to the appropriate sections there, as required.

The rest of these course notes are organized as follows. In section 2, we summarize the complications that arise in practice, and the key theoretical ideas that are relevant for practical inverse rendering problems. In section 3, we use these ideas to derive a new low-parameter dual angular and frequency space representation applied in the next section. Section 4 presents our new practical algorithms in detail, and illustrates the concepts using spheres of different materials. Finally, section 5 concludes these course notes and discusses unsolved problems that may be directions for future work.

2 Preliminaries

The input to our algorithms consists of object geometry (acquired using a laser range scanner and a volumetric merging algorithm [1]) and photographs from a number of different viewing directions, with known extrinsic and intrinsic camera parameters. We assume static scenes, i.e. that the object remains stationary and the lighting remains the same between views. Our method is a passive-vision approach; we do not actively disturb the environment. We will also assume the illumination comes from distant sources, and is a function only of the global incident direction, which can be represented with an environment map. For simplicity, we will restrict ourselves to isotropic BRDFs and neglect the effects of interreflection. Our theoretical analysis also discounts self-shadowing for concave surfaces, although our practical algorithms will account for it where necessary. Our assumptions (known geometry, distant illumination, isotropic BRDFs and no interreflections) are commonly made in computer vision and interactive computer graphics.

Recently, we have developed a signal-processing framework [13] for reflection based on the assumptions outlined above, ignoring concavities and self-shadowing. Thus, the reflected light field can be expressed as a spherical convolution of the incident illumination and the BRDF, and written as a product of spherical harmonic coefficients of the lighting and BRDF. This allows us to view inverse rendering as *deconvolution*, or as a *factorization* of the reflected light field into the lighting and BRDF. Our theory leads to several new insights by reformulating reflection in the frequency domain. However, the frequency-space ideas must be put into practice carefully. This is analogous to practical implementation of the Fourier-space theory of aliasing. The ideal Fourier-space bandpass filter in the spatial domain, the sinc function, is usually modified for practical purposes because it has infinite extent and leads to ringing. Similarly, representing BRDFs purely as a linear combination of spherical harmonics leads to ringing. Moreover, in practical applications, the following complications often arise. Below, we list the important practical issues, followed by a discussion of the practical implications of the theory, and how we address these issues.

2.1 Practical Complications

Small number of photographs: Our signal-processing framework [13] allows us to derive analytic formulae for the lighting and BRDF spherical harmonic coefficients. However, these formulae require knowledge of all the reflected light field coefficients. In practical applications, we often want to work with a small number of photographs. Just as it is difficult to compute Fourier spectra from sparse irregularly sampled data, it is difficult to compute the reflected light field coefficients from a few photographs; we would require a very large number of input images, densely sampling the entire sphere of possible directions. Therefore, we use ideas from the theory to derive a simple practical low-parameter model of the reflected light field for the microfacet BRDF, that simultaneously uses the angular domain and the frequency domain.

Concave objects and self shadowing: Real geometry can often be complex with self shadowing due to concave regions. As discussed in the next subsection, we use associativity of convolution to blur the illumination and treat the specular BRDF component like a mirror. This allows us to simply check the reflected ray

[*] ravir@graphics.stanford.edu

for shadowing, which is straightforward in a raytracer.

Concavities (or multiple objects) also lead to interreflections. We have not yet addressed this issue, but we believe an iterative solution, similar to that used by Yu et al. [18] should be applicable.

Textured objects: Finally, our theoretical analysis does not account for spatial variation in the BRDF. Here, we make the common assumption that this spatial variation can be accounted for simply by texture-mapping the BRDF parameters, i.e. the diffuse and specular reflectivities.

2.2 Practical implications of theory

We now discuss a number of key insights and quantitative results obtained from the theory that influence our practical representations, which address the issues raised above.

Dual Angular and Frequency-Space Representations: Quantities local in angular space have broad frequency spectra and vice-versa. By developing a frequency-space view of reflection, we ensure that we can use either the angular-space or frequency-space representation, or even a combination of the two. The diffuse BRDF component is slowly varying in angular-space, but is local in frequency-space, while the specular BRDF component is local in the angular domain. For representing the lighting, the frequency-space view is appropriate for the diffuse BRDF component, while the angular-space view is appropriate for the specular component. In the next section, we derive a dual angular and frequency-space representation for the reflected light field from the microfacet BRDF model. A similar form can be derived for other common BRDFs like the Phong reflection model. This low-parameter representation may be used for inverse rendering when we have only a small number of photographs available, i.e. sparse sampling of the reflected light field.

Irradiance formula: For the Lambertian BRDF component, we have derived [12] a simple analytic formula, and have shown that the irradiance at all surface orientations can be approximated to within 1% using only 9 parameters, i.e. coefficients of spherical harmonics up to order 2. Thus, it makes sense to apply this simple formula where possible, representing the diffuse component of the reflected light field in the frequency domain.

Associativity of convolution: Because the coefficients of the reflected light field in the frequency domain are simply a product of the spherical harmonic coefficients of the incident illumination and the BRDF, we may apply the associativity of convolution. Thus, we can blur the illumination and sharpen the BRDF without changing the final results. In the extreme case, for specular models like the Phong BRDF, we may treat the BRDF as a mirror, while blurring the illumination, convolving it with the BRDF filter. Within the context of environment map rendering [4, 11], this is known as prefiltering. Besides increased efficiency, this approach also allows for very efficient approximate computation of cast shadows due to concave surfaces. One need simply check the reflected ray, as if the surface were a mirror, which is a simple operation in a raytracer.

Separation of slow and fast-varying lighting: In general, because the lighting and BRDF are not one-dimensional quantities, applying the associativity property above destroys the symmetries and reciprocity of the BRDF, so we cannot simply blur the illumination and treat the BRDF as a perfect mirror. However, for radially symmetric specular BRDFs, like the Phong model, where the BRDF depends only on the angle between the incident illumination and the reflection of the viewing direction about the surface normal, this is a valid operation. Therefore, we separate the illumination into slow and fast-varying components, corresponding to area sources and point sources. It can be shown that for low-frequency lighting, models like the microfacet BRDF (Torrance-Sparrow [17] model) behave much like a Phong model (the dominant term is Phong-like reflection), so that we may blur the illumination and treat the BRDF as a mirror. Furthermore, the largest errors in this approximation occur for grazing angles, where measurements are accorded low confidence in practical applications anyway. The fast-varying lighting components may be treated as point sources, which makes it easy to find angular-space formulae for the reflected light field.

It should be noted that the theoretical analysis is conducted without taking concavities into account. We will derive our representation in the next section under the convex-surface assumption. However, we will also show there how the representation can be simply extended to account for textured objects and cast shadows.

3 Dual angular and frequency-space representation

In a sense, the practical implications discussed above simply formalize a reflection model commonly used when rendering with environment maps. In that context, the BRDF is assumed to be a combination of Lambertian diffuse and Phong specular reflection. The reflected light is then the combination of a diffuse irradiance map due to the Lambertian BRDF component and a specular reflection map due to the specular Phong lobe. Our theoretical analysis allows for two practical improvements to be made. Firstly, the irradiance map can be represented using only 9 parameters in the frequency domain, which makes computations more efficient and compact. Secondly, we may use a single angular-space reflection map as a good approximation for the specular reflections, even for more complex physically-based BRDFs like the microfacet model [17], provided we first separate the lighting into slow and fast-varying components. This section is an expanded version of section 5.3 in our SIGGRAPH paper [13].

We will use a simplified Torrance-Sparrow [17] model, defined as follows.

$$\begin{aligned}
\rho(\vec{\omega}_i, \vec{\omega}_o) &= K_d + K_s \frac{FS}{4\cos\theta_i\,\cos\theta_o} \\
\vec{\omega}_h &= \frac{\vec{\omega}_i + \vec{\omega}_o}{\parallel \vec{\omega}_i + \vec{\omega}_o \parallel} \\
F &= \frac{F(\mu, \theta_o)}{F(\mu, 0)} \\
S &= \frac{1}{\pi\sigma^2} \exp\left[-\left(\theta_h/\sigma\right)^2\right] \qquad (1)
\end{aligned}$$

Here, ρ is the BRDF, and σ is the surface roughness parameter. The BRDF is a function of incoming and outgoing directions $(\vec{\omega}_i, \vec{\omega}_o)$. The incident and outgoing angles with respect to the surface normal are denoted by θ_i and θ_o. Note that for clarity, we have dropped the primes as compared to the SIGGRAPH paper, since there is now no confusion regarding global or local coordinates. The subscript h stands for the half-way vector. $F(\mu, \theta_o)$ is the Fresnel term for refractive index μ; we normalize it to be 1 at normal exitance. Actually, F depends on the angle with respect to the half-way vector; in practice, this angle is usually very close to θ_o. For simplicity in the analysis, we have omitted the geometric attenuation factor G. In practice, this omission is not very significant except for observations made at grazing angles, which are usually assigned low confidence anyway in practical applications.

3.1 Model for reflected light field

Our model for the reflected light from the microfacet BRDF now includes three terms.

$$B = B_d + B_{s,slow} + B_{s,fast} \qquad (2)$$

Here, B is the net reflected light field. The component because of the diffuse part in the BRDF is denoted B_d. $B_{s,slow}$ represents specularities from the slowly-varying lighting, and $B_{s,fast}$ specular highlights from the fast varying lighting component.

We may represent and compute B_d in the frequency domain by using the irradiance formula (which corresponds directly to the reflection from a Lambertian surface). We use the 9 parameter representation, explicitly noting the frequency $l \leq 2$.

$$\begin{aligned} B_d &= K_d E(\alpha, \beta) \\ E(\alpha, \beta) &= \sum_{l=0}^{2} \left(\hat{A}_l \sum_{m=-l}^{+l} L_{lm} Y_{lm}(\alpha, \beta) \right) \end{aligned} \qquad (3)$$

Here, E is the irradiance, and K_d is the albedo or coefficient for diffuse reflection. The surface is parameterized by its orientation or surface normal in spherical coordinates (α, β). The spherical harmonics are denoted by Y_{lm}, and the spherical harmonic coefficients of the lighting by L_{lm}. The numerical values of $\hat{A}_l$ are given by

$$\hat{A}_0 = \pi \qquad \hat{A}_1 = 2\pi/3 \qquad \hat{A}_2 = \pi/4 \qquad (4)$$

For $B_{s,slow}$, we filter the lighting, and treat the BRDF as a mirror. With $\vec{R}$ denoting the reflected direction, and L_{slow} the filtered version of the lighting, we obtain

$$B_{s,slow} = K_s F(\mu, \theta_o) L_{slow}(\vec{R}) \qquad (5)$$

The filtered version of the illumination L_{slow} is obtained by multiplying the illumination coefficients by those of a filter corresponding to the term S in the microfacet BRDF of equation 1, i.e.

$$L_{lm}^{slow} = \exp\left[-(\sigma l)^2\right] L_{lm} \qquad (6)$$

In the angular domain, this corresponds to convolving with a filter of angular width approximately σ^{-1}, or using a normalized Phong lobe with Phong exponent $\frac{1}{2}\sigma^{-2}$.

For the *fast varying* portion of the lighting—corresponding to sources of angular width $\ll \sigma$—we treat the total energy of the source, given by an integral over the (small) solid angle subtended, as located at its center, so the lighting is a delta function. $B_{s,fast}$ is given by the standard equation for the specular highlight from a directional source. The extra factor of $4 \cos\theta_o$ in the denominator as compared to equation 5 comes from the relation between differential microfacet and global solid angles.

$$\begin{aligned} B_{s,fast} &= \frac{K_s F(\mu, \theta_o)}{4 \cos\theta_o} \sum_j T_j(\sigma) \\ T_j(\sigma) &= \exp\left[-(\theta_h/\sigma)^2\right] \left(\frac{L_{j,fast}}{\pi\sigma^2} \right) \end{aligned} \qquad (7)$$

The subscript j denotes a particular directional source; there could be several. Note that $L_{j,fast}$ is now the *total energy* of the source.

For BRDF estimation, it is convenient to expand out these equations, making dependence on the BRDF parameters explicit.

$$\begin{aligned} B &= K_d \sum_{l=0}^{2} \left(\hat{A}_l \sum_{m=-l}^{+l} L_{lm} Y_{lm}(\alpha, \beta) \right) \\ &+ K_s F(\mu, \theta_o) \left[L_{slow}(\vec{R}) + \frac{1}{4\cos\theta_o} \sum_j T_j(\sigma) \right] \end{aligned} \qquad (8)$$

3.2 Textures and shadowing

We now show how to extend our representation to account for object textures and self-shadowing on complex concave geometry. The representation can be extended to textured surfaces simply by letting the BRDF parameters (such as K_d and K_s) be functions of surface location. It would appear that concave regions, where one part of the surface may shadow another, are a more serious problem since our theory is developed for convex objects and assumes no self-shadowing. However, in the remainder of this section, we will see that the extensions necessary mainly just involve checking for shadowing of the reflected ray and directional sources, which are routine operations in a raytracer.

We consider each of the three terms in our model of the reflected light field. In the presence of shadows, the 9 parameter model can no longer be used to directly compute B_d. Instead, the irradiance may be computed in the more conventional angular-space way by integrating the scene lighting while considering visibility. Alternatively, we can continue to use a spherical harmonic approximation, making use of the linearity of light transport. Note that the irradiance can still be written as a linear combination of lighting coefficients. Thus, we may replace equation 3 by

$$\begin{aligned} B_d &= K_d E(\vec{x}) \\ E(\vec{x}) &= \sum_{l=0}^{l_{max}} \sum_{m=-l}^{+l} L_{lm} \tilde{Y}_{lm}(\vec{x}) \end{aligned} \qquad (9)$$

Here, we have increased the maximum frequency from 2 to l_{max}, where l_{max} can be larger than 2. Further, we have replaced the spherical harmonics with $\tilde{Y}_{lm}$. $\tilde{Y}_{lm}$ is the effect of the illumination spherical harmonic Y_{lm}. Since this effect now depends on the specific shadowing patterns, we have replaced the surface normal (α, β) with the position $\vec{x}$. For convex objects, as per equation 3, $\tilde{Y}_{lm}(\vec{x}) = \hat{A}_l Y_{lm}(\alpha, \beta)$.

For the specular components of the reflected light field, we simply check if the reflected ray (for the "slow" component) or the point sources (for the "fast" component) are shadowed. The main benefit is for slow specularities, where instead of a complex integration including visibility, the effects of shadowing are approximated simply by checking the reflected ray. It should be emphasized that in all cases, the corrections for visibility depend only on object geometry and viewing configuration (to determine the reflected direction), and can be precomputed for each point on the object using a ray tracer. Thus, we may replace equation 5 by

$$B_{s,slow} = K_s F(\mu, \theta_o) L_{slow}(\vec{R}) V(\vec{R}) \qquad (10)$$

where V is a binary value specifying if the reflected ray is unshadowed. Similarly, a visibility term needs to multiply T_j in equation 7. Putting it all together, and including the effects of textures, by making the diffuse and specular reflectances function of position $\vec{x}$, equation 8 becomes

$$\begin{aligned} B &= K_d(\vec{x}) \sum_{l=0}^{l_{max}} \sum_{m=-l}^{+l} L_{lm} \tilde{Y}_{lm}(\vec{x}) \\ &+ K_s(\vec{x}) F(\mu, \theta_o) \left[V(\vec{R}) L_{slow}(\vec{R}) + \frac{1}{4\cos\theta_o} \sum_j V_j T_j(\sigma) \right] \end{aligned} \qquad (11)$$

4 Algorithms

This section presents our practical algorithms for a broad range of inverse rendering problems under complex illumination, with simple illustrations using spheres of different materials. The objective is to describe some of our practical algorithms in greater detail than in sections 6.1-6.3 of our SIGGRAPH paper. The first part of section 6 in that paper contains details on image acquisition and experimental setup, which will not be duplicated here. Similarly, refer to section 6.4 of our SIGGRAPH paper for results using complex geometric and textured objects.

We describe two types of methods—algorithms that recover coefficients of a purely frequency-space description of the lighting or BRDF by representing these quantities as a sum of spherical harmonic terms, and algorithms that estimate parameters corresponding to our dual angular and frequency-space model of section 3. Section 4.1 on BRDF estimation discusses direct recovery of spherical harmonic BRDF coefficients, as well as estimation of parametric microfacet BRDFs using equations 8 and 11. Similarly, section 4.2 demonstrates direct recovery of spherical harmonic lighting coefficients, as well as estimation of a dual angular and frequency-space lighting description as per the model of section 3. Finally, section 4.3 shows how to combine BRDF and lighting estimation techniques to simultaneously recover the lighting and BRDF parameters, when both are unknown. In this case, we do not show direct recovery of spherical harmonic coefficients, as we have thus far found this to be impractical.

4.1 Inverse BRDF with known lighting

Estimation of Spherical Harmonic BRDF coefficients: Spherical harmonics and Zernike polynomials have been fit [7] to measured BRDF data, but previous work has not tried to estimate coefficients directly. Since the BRDF is linear in the coefficients $\hat{\rho}_{lpq}$, we simply solve a linear system to determine $\hat{\rho}_{lpq}$, to minimize the RMS error with respect to image observations[1]. It should be noted that in so doing, we are effectively interpolating (and extrapolating) the reflected light field to the entire 4D space, from a limited number of images.

Figure 1 compares the parametric BRDFs estimated under complex lighting to BRDFs measured using a single point source with the method of Marschner et al. [10]. As expected [7], the recovered BRDFs exhibit ringing. One way to reduce ringing is to attenuate high-frequency coefficients. According to our theory, this is equivalent to using low frequency lighting. Therefore, as seen in figure 1, images rendered with low-frequency lighting do not exhibit ringing and closely match real photographs, since only the low-frequency components of the BRDF are important. However, images rendered using directional sources show significant ringing.

For practical applications, it is usually more convenient to recover low-parameter BRDF models since these are compact, can be estimated from relatively fewer observations, and do not exhibit ringing. In the rest of this section, we will derive improved inverse rendering algorithms, assuming a parametric microfacet BRDF model.

Estimation of parametric BRDF model: We estimate BRDF parameters under general known lighting distributions using equation 8. The inputs are images that sample the reflected light field B. We perform the estimation using nested procedures. In the outer procedure, a simplex algorithm adjusts the nonlinear parameters μ and σ to minimize RMS error with respect to image pixels. In the inner procedure, a linear problem is solved for K_d and K_s. For numerical work, we use the simplex method `e04ccc` and linear solvers `f01qcc and f01qdc` in the NAG [5] C libraries. The main difference from previous work is that equation 8 provides a principled way of accounting for all components of the lighting and BRDF, allowing for the use of general illumination conditions.

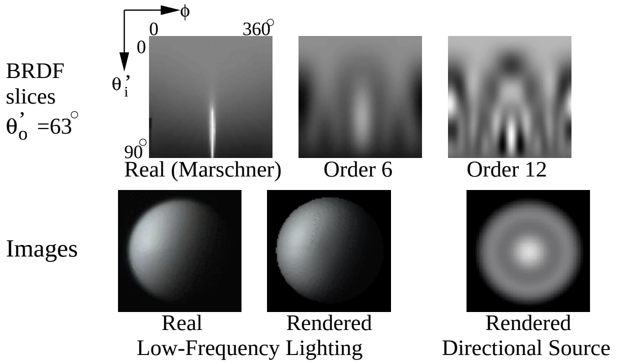


Figure 1: **Top:** *Slices of the BRDF transfer function of a teflon sphere for fixed exitant angle of* 63°. θ_o *varies linearly from* 0° *to* 90° *from top to bottom, and* $\mid \phi_o - \phi_i \mid$ *linearly from* 0° *to* 360° *from left to right. The central bright feature is the specular highlight. Left is the BRDF slice independently measured using the approach of Marschner et al. [10], middle is the recovered value using a maximum order 6, and right is the recovered version for order 12. Ringing is apparent in both recovered BRDFs. The right version is sharper, but exhibits more pronounced ringing.* **Bottom:** *Left is an actual photograph; the lighting is low-frequency from a large area source. Middle is a rendering using the recovered BRDF for order 6 and the same lighting. Since the lighting is low-frequency, only low-frequency components of the BRDF are important, and the rendering appears very similar to the photograph even though the recovered BRDF does not include frequencies higher than order 6. Right is a rendering with a directional source at the viewpoint, and exhibits ringing.*

We tested our algorithm on the spheres. Since the lighting includes high and low-frequency components (a directional source and an area source), the theory predicts that parameter estimation is well-conditioned. To validate our algorithm, we compared parameters recovered under complex lighting for one of the samples, a white teflon sphere, to those obtained by fitting to the full BRDF separately measured by us using the method of Marschner et al. [10]. Unlike most previous work on BRDF estimation, we consider the Fresnel term. It should be noted that accurate estimates for the refractive index μ require correct noise-free measurements at grazing angles. Since these measurements tend to be the most error-prone, there will be small errors in the estimated values of μ for some materials. Nevertheless, we find the Fresnel term important for reproducing accurate specular highlights at oblique angles. It should also be noted that while the results are quite accurate, there is still potential for future work on appropriate error metrics, especially for estimation of the roughness σ; a linear RMS error may not always be optimal.

Parameter	Our Method	Fit to Data
Reflectance	0.86	0.87
$K_d/(K_d + K_s)$	0.89	0.91
$K_s/(K_d + K_s)$	0.11	0.09
μ	1.78	1.85
σ	0.12	0.13
RMS	9.3%	8.5%

Figure 2: *Comparison of BRDF parameters recovered by our algorithm under complex lighting to those fit to measurements made by the method of Marschner et al. [10].*

The results in figure 2 show that the estimates of BRDF parameters from our method are quite accurate, and there is only a small

[1] Since the number of image pixels in a number of views can be very large, we randomly subsample the data for computational simplicity. We have used 12000 randomly selected image pixels.

increase in the error-of-fit when using parameters recovered by our algorithm to fit the measured BRDF. We also determined percentage RMS errors between images rendered using recovered BRDFs and real photographs to be between 5 and 10%. A visual comparison is shown in the first and third rows of figure 7. All these results indicate that, as expected theoretically, we can accurately estimate BRDFs even under complex lighting.

Textured objects with complex geometry: Handling concavities in complex geometric objects is not significantly more difficult, since we simply need to take visibility into account, and use equation 11 instead of equation 8. Equation 11 can also be used directly to estimate textured BRDFs. However, there are a number of subtle differences from direct BRDF estimation, which are noted below.

In considering textured surfaces, we essentially wish to consider each point on the surface separately, estimating a BRDF for each point independently from observations of that point alone. However, we now have only a few observations for each point (the number of images used). If there were no image noise, and our simplified four parameter microfacet model were a perfectly accurate description of the surface, this would still be sufficient. However, in practice, we are not able to reliably estimate the nonlinear parameters from such sparse data. This is true even for point source illumination, and has been observed by many authors. In our case, since we have complex illumination, the problem is even harder. Therefore, like much previous work, we assume the nonlinear parameters σ and μ are constant across the surface. A weaker assumption would be to allow them to vary slowly, or break the surface into regions of constant μ and σ.

Therefore, we will solve for the global nonlinear parameters σ and μ, as well as the diffuse and specular textures, $K_d(\vec{x})$ and $K_s(\vec{x})$. The corresponding radiance values for each image observation can be written as

$$B = K_d(\vec{x})D + K_s(\vec{x})S(\mu, \sigma) \tag{12}$$

where D and S stand for the diffuse and specular components computed from equation 11. These depend only on the lighting and viewing configuration, and S also depends on the nonlinear parameters μ and σ. It should be noted that much previous work has assumed constant values for the specular coefficient. The reason is that specularities are not usually observed over the whole object surface. By using complex illumination, we alleviate this problem somewhat, since large regions of the object can exhibit specularity in a single image. Nevertheless, there might be dimly lit regions or places where no specularities are observed in a sequence of views, and we will not be able to estimate coefficients in these regions. Therefore, we introduce confidence measures to enscapsulate the importance of each observation.

$$\begin{aligned} W_d &= \frac{D\cos\theta_o}{\epsilon + S} \\ W_s &= S\cos\theta_o \end{aligned} \tag{13}$$

Here, W_d and W_s are the confidence parameters for diffuse and specular reflection respectively. The multiplication by $\cos\theta_o$ is to give less weight to observations made at grazing exitant angles. ϵ is a small constant to avoid divisions by 0. In the diffuse weight W_d, we give greater importance to well illuminated pixels (high values of D) without too much specularity. In the specular weight W_s, we give importance to pixels observing strong specular reflections S.

Parameter estimation now proceeds much as BRDF estimation for untextured surfaces. Initially, we solve for values of the nonlinear parameters μ and σ using a simplex algorithm (outer loop). To account for regions where specularity is not strongly observed, in this phase, we include K_s as a global parameter to be solved for. In the inner loop of the procedure, we solve (at each point separately) for $K_d(\vec{x})$ to minimize the RMS error over all views. The output from this stage are parameters K_s, μ and σ, as well as an initial estimate of the diffuse texture $K_d(\vec{x})$. We use these global values of μ and σ. The global estimated value of K_s will be used in regions where a better estimate is not possible, but will in general be refined. In this first pass of the algorithm, we weight each observation using the confidence weight W_d.

We then use an iterative scheme to refine the estimates of K_s and K_d. While we could simply solve a linear system, corresponding to equation 12, for each vertex on the object, we have obtained better results using an iterative scheme, alternatively solving for $K_d(\vec{x})$ and $K_s(\vec{x})$ while keeping the other fixed. Since we use the dielectric model, K_s has no color, and we recover 4 linear texture parameters for each pixel (a diffuse RGB color and a specular coefficient). It should be noted that different confidence weights (W_d or W_s) are used in the iteration, depending on whether we are estimating the diffuse or specular component of the texture. We start by using a constant value of K_s, and the corresponding value of $K_d(\vec{x})$ recovered in the first phase, where we solved for μ and σ. We then hold $K_d(\vec{x})$ fixed and solve for $K_s(\vec{x})$. Thereafter, we hold $K_s(\vec{x})$ fixed and solve for $K_d(\vec{x})$, and repeat this process till convergence to the desired tolerance, which usually takes a few iterations.

There can of course be cases where $\sum W_s$ or $\sum W_d$ (the summation is over all views of that point) are too low (numerically zero) to accurately estimate specular or diffuse textures respectively. This corresponds to not observing specularities (when $\sum W_s$ is close to 0), or having the point being so dimly lit that the texture isn't discernible (when $\sum W_d$ is close to 0). In the former case, we simply use the mean value of the specular texture, while in the latter case, we mark the diffuse texture estimate as unreliable. It should be noted that using complex illumination greatly reduces the number of points where this is an issue, since much more of the object receives illumination and exhibits specularities than with a point light source.

4.2 Inverse Lighting with Known BRDF

Previous methods for estimating the lighting have been developed only for the special cases of mirror BRDFs (a gazing sphere), Lambertian BRDFs (Marschner and Greenberg [9]), and when shadows are present (Sato et al. [14]). Previous methods [9, 14] have also required regularization using penalty terms with user-specified weights, and have been limited by the computational complexity of their formulations to a coarse discretization of the sphere. We present two new algorithms for curved surfaces with general BRDFs. The first method directly recovers spherical harmonic lighting coefficients L_{lm}. The second algorithm estimates parameters of the dual angular and frequency-space lighting model of section 2. This method requires no explicit regularization, and yields high-resolution results that are sharper than those from the first algorithm, but is more difficult to extend to concave surfaces.

The theory tells us that inverse lighting is ill-conditioned for high-frequencies. Therefore, we will recover only low-frequency continuous lighting distributions, and will not explicitly account for directional sources, i.e. we assume that $B_{s,fast} = 0$. The reflected light field is essentially independent of the surface roughness σ under these conditions, so our algorithms do not explicitly use σ. The theory predicts that the recovered illumination will be a filtered version of the real lighting. Directional sources will appear as filtered into continuous distributions of angular width approximately σ.

Estimation of Spherical Harmonic Lighting coefficients: We may represent the lighting entirely in frequency-space by coefficients L_{lm} with $l \le l^*$, and solve a linear least-squares system for L_{lm}. The first term in parentheses below corresponds to B_d, and the second to $B_{s,slow}$. The cutoff l^* is used for regularization, and should be of order $l^* \sim \sigma^{-1}$. Since most materials have $\sigma \sim .1$, we use $l^* = 12$.

$$B = \sum_{l=0}^{l^*} \sum_{m=-l}^{l} L_{lm} \left(K_d \hat{A}_l Y_{lm}(\alpha, \beta) + K_s F Y_{lm}(\theta_R, \phi_R) \right) \tag{14}$$

To extend this to concave surfaces, we simply need to add terms corresponding to visibility and shadowing, following equation 11, but the problem remains a linear system.

$$B = \sum_{l=0}^{l^*} \sum_{m=-l}^{l} L_{lm} \left(K_d \tilde{Y}_{lm}(\vec{x}) + K_s F V(\theta_R, \phi_R) Y_{lm}(\theta_R, \phi_R) \right) \tag{15}$$

Estimation of Parametric Dual Lighting Model: Another approach is to estimate the dual angular and frequency-space lighting model of section 3. Our algorithm is based on subtracting out the diffuse component B_d of the reflected light field. After this, we treat the object as a mirror sphere, recovering a high-resolution angular-space version of the illumination from the specular component alone. To determine B_d, we need only the 9 lowest frequency-space coefficients L_{lm} with $l \leq 2$. Our algorithm uses the following methods to convert between angular and frequency-space:

1. **9 parameters to High-Resolution Lighting:** The inputs to phase 1 are the coefficients L^1_{lm}. These suffice to find B^1_d by equation 3. Since we assumed that $B_{s,fast} = 0$,

   $$\begin{aligned} B_{s,slow} &= K_s F(\mu, \theta_o) L_{slow}(\vec{R}) \; = B - B^1_d(L^1_{lm}) \\ L_{slow}(\vec{R}) &= \frac{B - B^1_d(L^1_{lm})}{K_s F(\mu, \theta_o)} \end{aligned}$$

   We assume the BRDF parameters are known, and B is the input to the algorithm, so the right-hand side can be evaluated.

   In practice, we will have several observations corresponding to the reflected direction, and these will be weighted by the appropriate confidence and combined. For simplicity, the rest of the mathematical discussion will assume without loss of generality, that there is a single image observation for each reflected direction.

2. **High-Resolution Lighting to 9 parameters:** Using the angular space values L found from the first phase, we can easily find the 9 frequency-space parameters of the lighting L^2_{lm}.

Now, assume we run phase 1 (with inputs L^1_{lm}) and phase 2 (with outputs L^2_{lm}) sequentially. The *consistency condition* is that $L^1_{lm} = L^2_{lm}$—converting from frequency to angular to frequency space must not change the result. Equivalently, the computed diffuse components must match, i.e. $B^1_d(L^1_{lm}) = B^2_d(L^2_{lm})$. This is illustrated in figure 3. Since everything is linear in terms of the lighting coefficients, the consistency condition reduces to a system of 9 simultaneous equations. After solving for L_{lm}, we run phase 1 to determine the high-resolution lighting in angular space.

More formally, phase 1 can be written as a linear system in terms of constants U and W_{lm}, with (α, β) the coordinates of the surface normal.

$$\begin{aligned} L_{slow}(\theta_R, \phi_R) &= U(\theta_R, \phi_R) - \sum_{l=0}^{2} \sum_{m=-l}^{l} W_{lm}(\theta_R, \phi_R) L^1_{lm} \\ U(\theta_R, \phi_R) &= \frac{B}{K_s F(\mu, \theta_o)} \\ W_{lm}(\theta_R, \phi_R) &= \frac{K_d \hat{A}_l Y_{lm}(\alpha, \beta)}{K_s F(\mu, \theta_o)} \end{aligned} \tag{16}$$

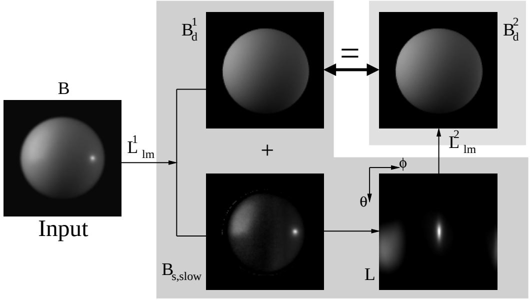


Figure 3: *Estimation of dual lighting representation. In phase 1, we use frequency-space parameters L^1_{lm} to compute diffuse component B^1_d. This is subtracted from the input image, leaving the specular component, from which the angular-space lighting is found. In phase 2, we compute coefficients L^2_{lm}, which can be used to determine B^2_d. The consistency condition is that $B^1_d = B^2_d$ or $L^1_{lm} = L^2_{lm}$. In this and all subsequent figures, the lighting is visualized by unwrapping the sphere so θ ranges in equal increments from 0 to π from top to bottom, and ϕ ranges in equal increments from 0 to 2π from left to right (so the image wraps around in the horizontal direction).*

Phase 2 to compute the lighting coefficients can also be written as a linear expression in terms of all the (discretized) reflected directions.

$$L^2_{lm} = \frac{2\pi^2}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sin\theta_i \, L_{slow}(\theta_i, \phi_j) Y^*_{lm}(\theta_i, \phi_j) \tag{17}$$

Here, N is the angular resolution, with the summation being a discrete version of the integral to find lighting coefficients.

But, the summation on the right hand side can be written in terms of lighting coefficients L^1_{lm}, simply by plugging in the formula for L_{slow}. We now obtain,

$$L^2_{lm} = \frac{2\pi^2}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sin\theta_i \left(U(\theta_i, \phi_j) - \sum_{l',m'} W_{l'm'}(\theta_i, \phi_j) L^1_{l'm'} \right) Y^*_{lm}(\theta_i, \phi_j) \tag{18}$$

Mathematically, the consistency condition allows us to drop the superscripts, reducing the above to a linear system for L_{lm}. This will involve a simple 9×9 linear system expressed in terms of a matrix $Q_{l'm',lm}$.

$$\begin{aligned} \sum_{l'=0}^{2} \sum_{m'=-l'}^{l'} Q_{lm,l'm'} L_{l'm'} &= P_{lm} \qquad (19) \\ P_{lm} &= \frac{2\pi^2}{N^2} \sum_{i,j} \sin\theta_i \, U(\theta_i, \phi_j) Y^*_{lm}(\theta_i, \phi_j) \\ Q_{lm,l',m'} &= \delta_{lm,l'm'} + \frac{2\pi^2}{N^2} \sum_{i,j} \sin\theta_i \, W_{l'm'}(\theta_i, \phi_j) Y^*_{lm}(\theta_i, \phi_j) \end{aligned}$$

The summations are just discrete versions of integrals that determine the appropriate spherical harmonic coefficients. The above equation has a very intuitive explanation. It may be derived direction from equation 8, considering the linear system that results for the first 9 lighting terms. The key idea is that we have reparameterized by the reflection vector, so we may simply take the first 9 coefficients of the reflected light field. The formula for the irradiance becomes more complicated (because of the reparameterization) but can still be expressed in terms of the first 9 lighting coefficients. Mathematically, we can rewrite equation 8 for our purposes as

$$\begin{aligned} B(\theta_R, \phi_R) &= K_d \sum_{l'=0}^{2} \sum_{m'=-l'}^{l'} \hat{A}_{l'} L_{l'm'} Y_{l'm'}(\alpha, \beta) \\ &+ K_s F(\mu, \theta_o) L_{slow}(\theta_R, \phi_R) \end{aligned} \qquad (20)$$

Here, we have simply parameterized the reflected light field by the reflected direction (θ_R, ϕ_R). Remember that for simplicity, we're assuming a single image, i.e. one value of (α, β) corresponding to each (θ_R, ϕ_R), with (α, β) a function of (θ_R, ϕ_R). With multiple images, we would have to weight contributions appropriately.

Now, it's a simple enough matter to compute coefficients obtaining

$$B_{lm} = K_d \sum_{l',m'} \hat{A}_{l'} L_{l'm'} \left\langle Y^*_{lm}(\theta_R, \phi_R), Y_{l'm'}(\alpha, \beta) \right\rangle + K_s F(\mu, \theta_o) L_{lm} \qquad (21)$$

Here, we have used the notation $< \cdot, \cdot >$ for the integral or inner product over the spherical domain of integration. This is what is computed discretely in equation 19. It can now be seen that equation 21 has the same form as equation 19. Note that in equation 21, we have multiplied out the denominators, and we use B_{lm} here instead of P_{lm}.

This inverse lighting method is difficult to extend to concave surfaces, since the 9 parameter diffuse model is no longer entirely valid. It is a subject of future work to see if it can be applied simply by increasing the number of parameters and the size of the matrix of simultaneous equations to be solved.

Positive regularization: So far, we have not explicitly tried to ensure positivity of the illumination. In practical applications, the methods above when applied naively will result in negative values, especially where the illumination is dark, and there is uncertainty about the precise value. Regularizing so the results are positive can also substantially improve the quality of the results by reducing high-frequency noise centered close to the zero point in dark regions.

We apply positive regularization to the unregularized solution from either of the previous two methods. For the first method (direct solution of linear system to determine lighting coefficients), we simply add another term to the RMS error which penalizes negative regions. While this method is a soft constraint and can still leave some negative regions, we have observed that it works well in practice. We use a conjugate gradient method to minimize

$$\nu' = \nu + \lambda \left(\frac{\nu_0}{\kappa_0} \right) \kappa \qquad (22)$$

Here, ν is the RMS error corresponding to equation 14 or 15. κ is a new penalty term added to penalize negative values of the lighting. ν_0 and κ_0 are initial values (for the unregularized solution), and λ weights the importance of the penalty term. λ is a dimensionless quantity, and we have found experimentally that $\lambda = 1$ works well. The penalty term κ is simply the sum of squares of all lighting pixels having negative values. Thus, negative values are penalized, but no penalty is imposed for positive values.

For our second method (using a dual angular and frequency-space method to estimate the lighting), regularization may be enforced (in step 1) simply by clamping L_{slow} to 0 if the right hand side in the first line of equation 16 is negative. This must be taken into account in the 9×9 simultaneous equations, and we solve the *positivity enforced* equations with a conjugate gradient method, using as a starting guess the solution without enforced positivity.

Comparison: Figure 4 compares the methods to each other, and to a reference solution from a gazing sphere. Both algorithms give reasonably accurate results. As predicted by the theory, high-frequency components are filtered by the roughness σ. In the first method, involving direct recovery of L_{lm}, there will still be some residual energy for $l > l^*$. Since we regularize by not considering higher frequencies—we could increase l^*, but this makes the result noisier—the recovered lighting is somewhat blurred compared to our dual angular and frequency-space algorithm (second method). As expected, positive regularization in algorithm 2 results in a smoother solution.

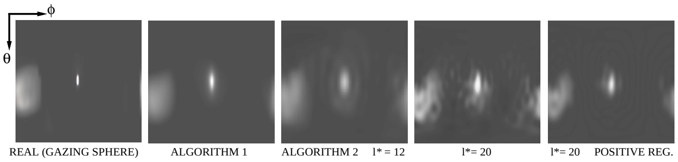


Figure 4: *Comparison of inverse lighting methods. From left to right, real lighting (from a gazing sphere), recovered illumination by direct estimation of spherical harmonic coefficients with* $l^* = 12$ *and* $l^* = 20$, *and estimation of dual angular and frequency-space lighting model. To make the artifacts more apparent, we have set* 0 *to gray. The results from the dual algorithm are sharper, but still somewhat blurred because of filtering by* σ. *A small amount of ringing occurs for direct coefficient recovery, and can be seen for* $l^* = 12$. *Using* $l^* = 20$ *makes the solution very noisy. Positive regularization (rightmost) gives a smoother solution.*

4.3 Factorization—Unknown Lighting and BRDF

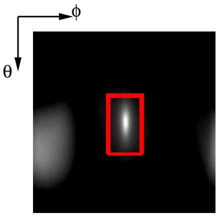
We can combine the inverse-BRDF and inverse-lighting methods to *factor* the reflected light field, simultaneously recovering the lighting and BRDF when both are unknown. Therefore, we are able to accurately recover BRDFs of curved surfaces under unknown complex illumination, something which has not previously been demonstrated. There is an unrecoverable global scale factor, so we set $K_d + K_s = 1$; we cannot find absolute reflectance. Also, the theory predicts that for low-frequency lighting, estimation of the surface roughness σ is ill-conditioned—blurring the lighting while sharpening the BRDF does not significantly change the reflected light field. However, for high-frequency lighting, this ambiguity can be removed. We will use a single manually specified directional source in the recovered lighting distribution to estimate σ.

Algorithm: The algorithm consists of nested procedures. In the outer loop, we effectively solve an inverse-BRDF problem—a nonlinear simplex algorithm adjusts the BRDF parameters to minimize error with respect to image pixels. Since $K_d + K_s = 1$, and σ will not be solved for till after the lighting and other BRDF parameters have been recovered, there are only 2 free parameters, K_s and μ. In the inner procedure, a linear problem is solved to estimate the lighting for a given set of BRDF parameters, using the methods of the previous subsection. Pseudocode is given below.

```
global $B_{input}$                              // Input images
global $K_d$,$K_s$,$\mu$,$\sigma$                 // BRDF parameters
global $L$                                      // Lighting
procedure Factor
    Minimize($K_s$,$\mu$,ObjFun)                // Simplex Method
    $\sigma$ = FindRoughness($L$)               // Figure 5, Equation 23
function ObjFun($K_s$,$\mu$)
    $K_d = 1 - K_s$                             // $K_d + K_s = 1$
    $L$ = Lighting($K_d$,$K_s$,$\mu$)           // Inverse Lighting
    $B_{pred}$ = Predict($L$,$K_d$,$K_s$,$\mu$) // Predicted Light Field
    return RMS($B_{input}$,$B_{pred}$)          // RMS Error
```

Finding σ using a directional source: If a directional source is present—and manually specified by us in the recovered lighting—we can estimate σ by equating specular components predicted by equations 5 and 7 for the center, i.e. brightest point, of the light source at normal exitance. An illustration is in figure 5.

$$L_{cen} \approx \frac{L_{total}}{4\pi\sigma^2} \qquad (23)$$



$L_{cen} = 1.0$

$L_{tot} = 0.14$

$\sigma = 0.11$

Figure 5: *We manually specify (red box) the region corresponding to the directional source in a visualization of the lighting. The algorithm then determines L_{cen}, the intensity at the center (brightest point), L_{tot}, the total energy integrated over the region specified by the red box, and computes σ using equation 23. The method does not depend on the size of the red box—provided it encloses the entire (filtered) source—nor the precise shape into which the source is filtered in the recovered lighting.*

Color: We have so far ignored issues of color, assuming the three color channels are considered separately. However, in the case of BRDF recovery under unknown lighting, there is a separate scale factor associated with each color channel. In order to obtain accurate colors for the BRDF and lighting components, we need some way to relate these 3 scale factors. For dielectrics, the specular component K_s is not spectrally sensitive, i.e. it is the same for red, green, and blue channels. The recovered BRDFs are scaled in order to make this hold. The issue is trickier for metals. There is a fundamental ambiguity between the color of the BRDF and the color of the lighting. We resolve this by considering the average color of the metallic surface as corresponding to white light. The use of more sophisticated color-space separation methods such as that of Klinker et al [6] might bring further benefits.

Results: We used the method of this subsection—with the dual angular and frequency-space algorithm for inverse lighting—to factor the light field for the spheres, simultaneously estimating the BRDF and lighting. The same setup and lighting were used for all the spheres so we could compare the recovered illumination distributions.

We see from figure 6 that the BRDF estimates under unknown lighting are accurate. Absolute errors are small, compared to parameters recovered under known lighting. The only significant anomalies are the slightly low values for the refractive index μ—caused because we cannot know the high-frequency lighting components, which are necessary for more accurately estimating the Fresnel term. We are also able to estimate a filtered version of the lighting. As shown in figure 7, the recovered lighting distributions from all the samples are largely consistent. As predicted by the theory, the directional source is spread out to different extents depending on how rough the surface is, i.e. the value of σ. Finally, figure 7 shows that rendered images using the estimated lighting and BRDF are almost indistinguishable from real photographs.

This section has been a detailed presentation of our practical algorithms for inverse rendering under complex illumination, using spheres of different materials to illustrate the concepts. As mentioned in the introduction to this section, our results using complex geometric models and textured objects are found in section 6.4 of our SIGGRAPH paper [13], to which we refer the reader for examples of images created by applying our algorithms to real-world objects under complex illumination conditions.

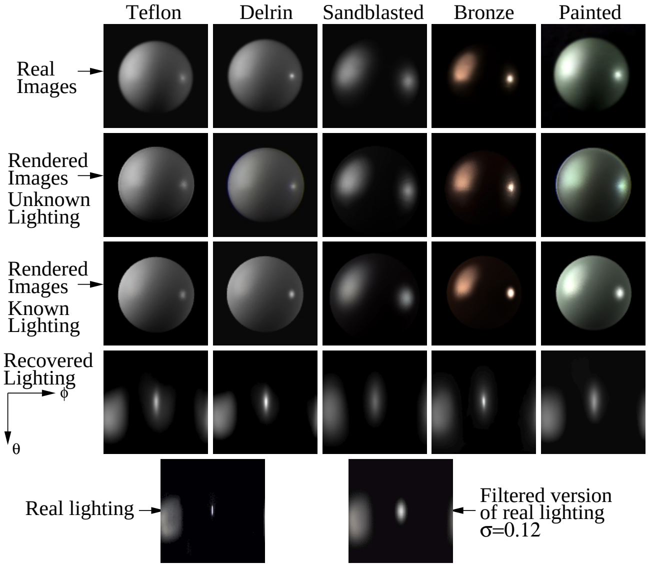

Teflon
Delrin
Sandblasted
Bronze
Painted
Real Images
Rendered Images Unknown Lighting
Rendered Images Known Lighting
Recovered Lighting
φ
θ
Real lighting
Filtered version of real lighting σ=0.12

Figure 7: *Spheres rendered using BRDFs estimated under known (section 4.1) and unknown (section 4.3) lighting. The algorithm in section 4.3 also recovers the lighting. Since there is an unknown global scale, we scale the recovered lighting distributions in order to compare them. The recovered illumination is largely consistent between all samples, and is similar to a filtered version of the real lighting. As predicted by the theory, the different roughnesses σ cause the directional source to be spread out to different extents. The filtered source is slightly elongated or asymmetric because the microfacet BRDF is not completely symmetric about the reflection vector.*

5 Conclusions and Future Work

We have considered the problem of complex illumination in inverse rendering. The use of realistic lighting and materials is likely to be of increasing importance in graphics and vision, and inverse rendering to acquire real-world material properties is likely to be a significant future direction. The use of natural lighting is becoming increasingly common in computer graphics and vision, and the techniques presented here allow for inverse rendering to be applied in arbitrary uncontrolled conditions rather than a laboratory setting. Furthermore, in certain cases, complex illumination may help in solving inverse problems, such as by allowing a much larger fraction of an image to exhibit specularity. Dror et al. [3] have also shown that people perceive reflectance properties much more easily under natural illumination.

However, this is only the first step in solving inverse problems under complex illumination. There are a number of open problems that remain subjects for future work. There has been relatively little work on BRDF representations in between low-parameter models and full measured representations. It is not clear what the best way to measure factored or separable BRDF representations is, or how to estimate these or higher-dimensional representations under complex illumination. We have not considered statistical properties of the illumination, that may allow us to simultaneously determine the lighting, BRDF and texture. All of our examples use a small number of images, and an interesting future direction is whether

Material	K_d		K_s		μ		σ	
	Known	Unknown	Known	Unknown	Known	Unknown	Known	Unknown
Teflon	0.89	0.87	0.11	0.13	1.78	1.48	0.12	0.14
Delrin	0.87	0.88	0.13	0.12	1.44	1.35	0.10	0.11
Neoprene Rubber	0.92	0.93	0.08	0.07	1.49	1.34	0.10	0.10
Sandblasted Steel	0.20	0.14	0.80	0.86			0.20	0.19
Bronze	(.15,.08,.05)	(.09,.07,.07)	(.85,.68,.59)	(.91,.69,.55)			0.12	0.10
Painted	(.62,.71,.62)	(.67,.75,.64)	0.29	0.25	1.38	1.15	0.15	0.15

Figure 6: *BRDFs of various spheres, recovered under known (section 4.1) and unknown (section 4.3) lighting. The reported values are normalized so* $K_d + K_s = 1$*. RGB values are reported for colored objects. We see that* K_s *is much higher for the more specular metallic spheres, and that* σ *is especially high for the rough sandblasted sphere. The Fresnel effect is very close to* 1 *for metals, so we do not consider the Fresnel term for these spheres.*

a single image suffices. We also believe there are more insights to be obtained from frequency space analysis and new frequency domain algorithms to be explored for inverse problems. Another interesting future direction is to preserve and extrapolate the original images to new conditions, using inverse rendering only as a guide in this extrapolation, but without explicitly computing lighting and reflectance parameters. This would have the potential to unify inverse and image-based (interpolatory) rendering methods.

References

[1] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH 96*, pages 303–312, 1996.

[2] P. Debevec, T. Hawkins, C. Tchou, H.P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *SIGGRAPH 00*, pages 145–156.

[3] R. Dror, E. Adelson, and A. Willsky. Estimating surface reflectance properties from images under unknown illumination. In *SPIE Photonics West: Human Vision and Electronic Imaging VI*, pages 231–242, 2001.

[4] N. Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics & Applications*, 6(11):21–29, 1986.

[5] Numerical Algorithms Group. *NAG C Library Manual, Mark 5*. 1999.

[6] G.J. Klinker, S.A. Shafer, and T. Kanade. The measurement of highlights in color images. *IJCV*, 2(1):7–32, 1988.

[7] J. J. Koenderink and A. J. van Doorn. Phenomenological description of bidirectional surface reflection. *JOSA A*, 15(11):2903–2912, 1998.

[8] R. Love. *Surface Reflection Model Estimation from Naturally Illuminated Image Sequences*. PhD thesis, Leeds, 1997.

[9] S.R. Marschner and D.P. Greenberg. Inverse lighting for photography. In *Fifth Color Imaging Conference*, pages 262–265, 1997.

[10] S.R. Marschner, S.H. Westin, E.P.F. Lafortune, and K.E. Torrance. Image-Based BRDF measurement. *Applied Optics*, 39(16):2592–2600, 2000.

[11] G. Miller and C. Hoffman. Illumination and reflection maps: Simulated objects in simulated and real environments. *SIGGRAPH 84 Advanced Computer Graphics Animation seminar notes*, 1984.

[12] R. Ramamoorthi and P. Hanrahan. On the relationship between radiance and irradiance: Determining the illumination from images of a convex lambertian object. *JOSA A*, 18(10):2448–2459, 2001.

[13] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. In *SIGGRAPH 01*, pages 117–128, 2001.

[14] I. Sato, Y. Sato, and K. Ikeuchi. Illumination distribution from brightness in shadows: adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. In *ICCV 99*, pages 875 – 882, 1999.

[15] Y. Sato and K. Ikeuchi. Reflectance analysis under solar illumination. Technical Report CMU-CS-94-221, CMU, 1994.

[16] Y. Sato, M. D. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. In *SIGGRAPH 97*, pages 379–388, 1997.

[17] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *JOSA*, 57(9):1105–1114, 1967.

[18] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *SIGGRAPH 99*, pages 215–224, 1999.

[19] Y. Yu and J. Malik. Recovering photometric properties of architectural scenes from photographs. In *SIGGRAPH 98*, pages 207–218, 1998.

[20] Z. Zhang. A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research, 1998.

A Signal-Processing Framework for Inverse Rendering

Ravi Ramamoorthi Pat Hanrahan

Stanford University [*]

Abstract

Realism in computer-generated images requires accurate input models for lighting, textures and BRDFs. One of the best ways of obtaining high-quality data is through measurements of scene attributes from real photographs by *inverse rendering*. However, inverse rendering methods have been largely limited to settings with highly controlled lighting. One of the reasons for this is the lack of a coherent mathematical framework for inverse rendering under general illumination conditions. Our main contribution is the introduction of a signal-processing framework which describes the reflected light field as a convolution of the lighting and BRDF, and expresses it mathematically as a product of spherical harmonic coefficients of the BRDF and the lighting. Inverse rendering can then be viewed as deconvolution. We apply this theory to a variety of problems in inverse rendering, explaining a number of previous empirical results. We will show why certain problems are ill-posed or numerically ill-conditioned, and why other problems are more amenable to solution. The theory developed here also leads to new practical representations and algorithms. For instance, we present a method to *factor* the lighting and BRDF from a small number of views, i.e. to estimate both simultaneously when neither is known.

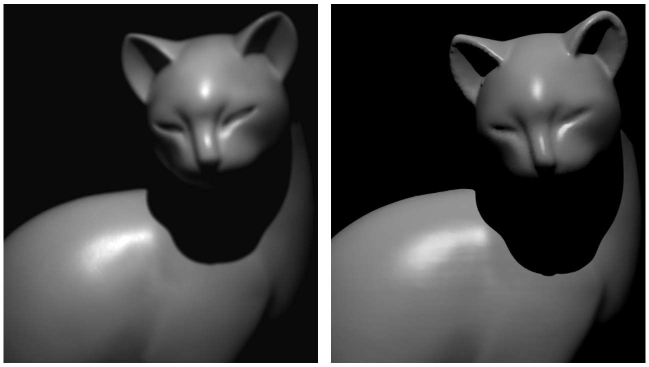
CR Categories: I.3.7 [*Computer Graphics*]: Realism; I.4.8 [*Computer Vision*]: Scene Analysis—Photometry

Keywords: Signal Processing, Spherical Harmonics, Inverse Rendering, Radiance, Light Field, Irradiance, Illumination, BRDF

1 Introduction

To create a realistic computer-generated image, we need both an accurate, physically-based rendering algorithm and a detailed model of the scene including light sources and objects specified by their geometry and material properties—texture and reflectance (BRDF). There has been substantial progress in the development of rendering algorithms, and nowadays, realism is often limited by the quality of input models. As a result, *image-based rendering* is becoming widespread. In its simplest form, image-based rendering uses view interpolation to construct new images from acquired images without constructing a conventional scene model.

The quality of view interpolation may be significantly improved if it is coupled with *inverse rendering*. Inverse rendering *measures* rendering attributes—lighting, textures, and BRDF—from photographs. Whether traditional or image-based rendering algorithms are used, rendered images use measurements from real objects, and therefore appear very similar to real scenes. Measuring scene attributes also introduces structure into the raw imagery, making it easier to manipulate the scene. For example, an artist can change independently the material properties or the lighting.



Real Photograph Rendered Image

Figure 1: **Left:** *Real Photograph* **Right:** *Rendered image. The BRDF used for the rendered image was estimated under complex unknown illumination from 3 photographs of a cat sculpture with known geometry. Our algorithm also recovered the lighting distribution, which consisted of two directional sources and an area source. The images above show a new view not used in BRDF recovery; the lighting is also new, being composed of a single directional source (with known direction) not used in BRDF estimation. These images show that the recovered BRDF accurately predicts appearance even under novel viewing and lighting conditions.*

Inverse rendering methods such as those of Debevec et al. [6], Marschner et al. [21], and Sato et al. [32], have produced high quality measurements. However, most previous work has been conducted in highly controlled lighting conditions, usually by careful active positioning of a single point source. Even methods that work in outdoor conditions, such as those of Yu and Malik [39], Sato and Ikeuchi [31] and Love [17], are designed specifically for natural illumination, and assume a simple parametric model for skylight. Previous methods have also usually tried to recover only one of the unknowns—texture, BRDF or lighting. The usefulness of inverse rendering would be greatly enhanced if it could be applied under general uncontrolled lighting, and if we could simultaneously estimate more than one unknown. For instance, if we could recover both the lighting and BRDF, we could determine BRDFs under unknown illumination. One reason there has been relatively little work in these areas is the lack of a common theoretical framework for determining under what conditions inverse problems can and cannot be solved, and for making principled approximations.

Our main contribution is a theoretical framework for analyzing the reflected light field from a curved convex homogeneous surface under distant illumination. We believe this framework provides a solid mathematical foundation for many areas of graphics. With respect to inverse rendering, we obtain the following results:

Reflection as Convolution: It has been observed qualitatively by Miller and Hoffman [23], Cabral et al. [3], Bastos et al. [2] and others that the reflection operator behaves like a convolution in the angular domain. We formalize these notions mathematically. The reflected light field can therefore be thought of in a precise quantitative way as obtained by convolving the lighting and BRDF, i.e. by filtering the illumination using the BRDF. We believe this is a useful way of analyzing many computer graphics problems. In particular, inverse rendering can be viewed as *deconvolution*.

Well-posedness and Conditioning of Inverse Problems: Inverse problems can be ill-posed—there may be no solutions or several solutions. They are also often numerically ill-conditioned, i.e. extremely sensitive to noisy input data. From our theory, we are able to analyze the well-posedness and conditioning of a number of inverse problems, explaining many previous empirical observations. This analysis can serve as a guideline for future research.

New Practical Representations and Algorithms: Insights from the theory lead to the derivation of a simple practical representation, which can be used to estimate BRDFs under complex lighting. The

[*] (ravir,hanrahan)@graphics.stanford.edu

theory also leads to novel frequency space and hybrid angular and frequency space methods for inverse problems, including two new algorithms for estimating the lighting, and an algorithm for simultaneously determining the lighting and BRDF. Therefore, we can recover BRDFs under general, unknown lighting conditions.

2 Previous Work

To describe previous work, we will introduce a taxonomy based on how many of the three quantities—lighting, BRDF and texture—are unknown. To motivate the taxonomy, we first write a simplified version of the reflection equation, omitting visibility.

$$B(\mathbf{x},\vec{\omega}_o) = \int_{\Omega_i} T(\mathbf{x})\rho(\vec{\omega}_i,\vec{\omega}_o)L(\mathbf{x},\vec{\omega}_i)(\vec{\omega}_i\cdot\vec{n})\,d\omega_i \qquad (1)$$

Here, B is the reflected light field, expressed as a function of the surface position $\mathbf{x}$ and outgoing direction $\vec{\omega}_o$. The normal vector is $\vec{n}$. For simplicity, we assume that a single texture T modulates the BRDF. In practice, we would use separate textures for the diffuse and specular components of the BRDF.

The integrand is a product of terms—the texture $T(\mathbf{x})$, the BRDF $\rho(\vec{\omega}_i,\vec{\omega}_o)$, and the lighting $L(\mathbf{x},\vec{\omega}_i)$. Inverse rendering, assuming known geometry, involves inverting the integral to recover one or more of ρ, L, or T. If two or more quantities are unknown, inverse rendering involves *factoring* the reflected light field.

One Unknown

1. Unknown Texture: Previous methods have recovered the diffuse texture of a surface using a single point light source by dividing by the irradiance in order to estimate the albedo at each point. Details are given by Marschner [34] and Levoy et al. [16].

2. Unknown BRDF: The BRDF [24] is a fundamental intrinsic surface property. Active measurement methods, known as gonioreflectometry, involving a single point source and a single observation at a time, have been developed. Improvements are suggested by Ward [37] and Karner et al. [12]. More recently, image-based BRDF measurement methods have been proposed by Lu et al. [18] and Marschner et al. [21]. If the entire BRDF is measured, it may be represented by tabulating its values. An alternative representation is by low-parameter models such as those of Ward [37] or Torrance and Sparrow [36]. The *parametric* BRDF will generally not be as accurate as a full *measured* BRDF. However, parametric models are often preferred in practice since they are compact, and are simpler to estimate. Love [17] estimates parametric BRDFs under natural illumination, assuming a low-parameter model for skylight and sunlight. Dror et al. [7] classify the surface reflectance as one of a small number of predetermined BRDFs, making use of assumed statistical characteristics of natural lighting. However, the *inverse BRDF* problem has not been solved for general illumination.

3. Unknown Lighting: A common solution is to use a mirrored ball, as done by Miller and Hoffman [23]. Marschner and Greenberg [20] find the lighting from a Lambertian surface. D'Zmura [8] proposes, but does not demonstrate, estimating spherical harmonic coefficients. For Lambertian objects, we [29] have shown how to recover the first 9 spherical harmonics. Previous work has not estimated the lighting from curved surfaces with general BRDFs.

Two Unknowns

4. Factorization—Unknown Lighting and BRDF: BRDF estimation methods have been proposed by Ikeuchi and Sato [10] and Tominaga and Tanaka [35] for the special case when the lighting consists of a single source of unknown direction. However, these methods cannot simultaneously recover a complex lighting distribution and the object BRDF. One of the main practical contributions of this paper is a solution to this problem for curved surfaces, allowing us to estimate BRDFs under general unknown illumination, while also determining the lighting. The closest previous work is that of Sato et al. [30] who use shadows to estimate the illumination distribution and the surface reflectance properties. We extend this work by not requiring shadow information, and presenting improved methods for estimating the illumination.

5. Factorization—Unknown Texture and BRDF: This corresponds to recovering *textured*, or spatially-varying BRDFs. Sato et al. [32] rotate an object on a turntable, using a single point source, to recover BRDF parameters and texture. Yu et al. [38] recover a texture only for the diffuse BRDF component, but account for interreflections. Using a large number of images obtained by moving a point source around a sphere surrounding the subject, Debevec et al. [6] acquire the reflectance field of a human face, and recover parameters of a microfacet BRDF model for each surface location. Sato and Ikeuchi [31] and Yu and Malik [39] recover BRDFs and diffuse textures under natural illumination, assuming a simple parametric model for skylight, and using a sequence of images acquired under different illumination conditions. Most of these methods recover only diffuse textures; constant values, or relatively low-resolution textures, are used for the specular parameters. A notable exception is the work of Dana et al. [5] who generalize BRDFs to a 6D bi-directional texture function (BTF).

6. Factorization—Unknown Lighting and Texture: We have shown [29] that a distant illumination field can cause only low frequency variation in the radiosity of a convex Lambertian surface. This implies that, for a diffuse object, high-frequency texture can be recovered independently of lighting. These observations are in agreement with the perception literature, such as Land's retinex theory [15], wherein high-frequency variation is usually attributed to texture, and low-frequency variation associated with illumination. However, note that there is a fundamental ambiguity between low-frequency texture and lighting effects. Therefore, lighting and texture cannot be factored without using active methods or making further assumptions regarding their expected characteristics.

General Case: Three Unknowns

7. Factorization—Unknown Lighting, Texture, BRDF: Ultimately, we wish to recover textured BRDFs under unknown lighting. We cannot solve this problem without further assumptions, because we must first resolve the lighting-texture ambiguity.

Our approach differs from previous work in that it is derived from a mathematical theory of inverse rendering. As such, it has similarities to inverse methods used in areas of radiative transfer and transport theory such as hydrologic optics [26] and neutron scattering. See McCormick [22] for a review.

In previous theoretical work, D'Zmura [8] has analyzed reflection as a linear operator in terms of spherical harmonics, and discussed some resulting perceptual ambiguities between reflectance and illumination. In computer graphics, Cabral et al. [3] first demonstrated the use of spherical harmonics to represent BRDFs. We extend these methods by explicitly deriving the frequency-space reflection equation (i.e. convolution formula), and by providing quantitative results for various special cases. We have earlier reported on theoretical results for planar or *flatland* light fields [27], and for determining the lighting from a Lambertian surface [29]. For the Lambertian case, similar results have been derived independently by Basri and Jacobs [1] in simultaneous work on face recognition. This paper extends these previous results to the general 3D case with arbitrary isotropic BRDFs, and applies the theory to developing new practical inverse-rendering algorithms.

3 Assumptions

The input to our algorithms consists of object geometry and photographs from a number of different locations, with known extrinsic and intrinsic camera parameters. We assume static scenes, i.e. that the object remains stationary and the lighting remains the same between views. Our method is a passive-vision approach; we do not actively disturb the environment. Our assumptions are:

B	Reflected radiance
B_{lmpq}	Coefficients of basis-function expansion of B
L	Incoming radiance
L_{lm}	Coefficients of spherical-harmonic expansion of L
ρ	Surface BRDF
$\hat{\rho}$	BRDF multiplied by cosine of incident angle
$\hat{\rho}_{lpq}$	Coefficients of spherical-harmonic expansion of $\hat{\rho}$
θ_i', θ_i	Incident elevation angle in *local*, *global* coordinates
ϕ_i', ϕ_i	Incident azimuthal angle in *local*, *global* coordinates
θ_o', θ_o	Outgoing elevation angle in *local*, *global* coordinates
ϕ_o', ϕ_o	Outgoing azimuthal angle in *local*, *global* coordinates
Ω_i', Ω_i	Hemisphere of integration in *local*,*global* coordinates
$\mathbf{x}$	Surface position
α	Surface normal parameterization—elevation angle
β	Surface normal parameterization—azimuthal angle
$R_{\alpha,\beta}$	Rotation operator for surface normal (α, β)
$D_{mm'}^l$	Matrix related to Rotation Group $SO(3)$
Y_{lm}	Spherical Harmonic basis function
Y_{lm}^*	Complex Conjugate of Spherical Harmonic
Λ_l	Normalization constant, $\sqrt{4\pi/(2l+1)}$
I	$\sqrt{-1}$

Figure 2: *Notation*

Known Geometry: We use a laser range scanner and a volumetric merging algorithm [4] to obtain object geometry. By assuming known geometry, we can focus on lighting and material properties.

Curved Objects: Our theoretical analysis requires curved surfaces, and assumes knowledge of the entire 4D reflected light field, corresponding to the hemisphere of outgoing directions for all surface orientations. However, our practical algorithms will require only a small number of photographs.

Distant Illumination: The illumination field will be assumed to be homogeneous, i.e. generated by distant sources, allowing us to use the same lighting function regardless of surface location. We treat the lighting as a general function of the incident angle.

Isotropic BRDFs: We will consider only surfaces having isotropic BRDFs. The BRDF will therefore be a function of only 3 variables, instead of 4, i.e. 3D instead of 4D.

No Interreflection: For concave surfaces, interreflection will be ignored. Also, shadowing is not considered in our theoretical analysis, which is limited to convex surfaces. However, we will account for shadowing in our practical algorithms, where necessary.

4 Theory of Reflection as Convolution

This section presents a signal-processing framework wherein reflection can be viewed as convolution, and inverse rendering as deconvolution. First, we introduce some preliminaries, defining the notation and deriving a version of the reflection equation. We then expand the lighting, BRDF and reflected light field in spherical harmonics to derive a simple equation in terms of spherical harmonic coefficients. The next section explores implications of this result.

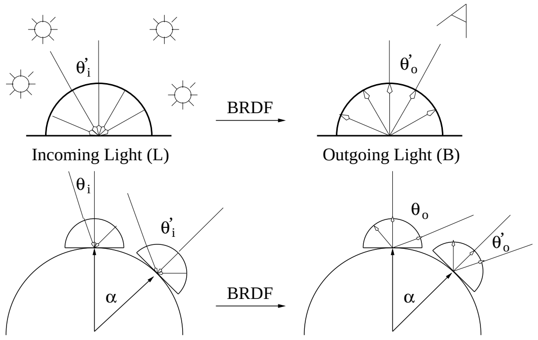


Figure 3: *Schematic of reflection. On top, we show the situation with respect to the local surface. The BRDF maps the incoming light distribution L to an outgoing light distribution B. The bottom figure shows how the rotation α affects the situation. Different orientations of the surface correspond to rotations of the upper hemisphere and BRDF, with global directions (θ_i,θ_o) corresponding to local directions (θ_i',θ_o').*

4.1 Preliminaries

For the purposes of theoretical analysis, we assume curved convex isotropic surfaces. We also assume homogeneous objects, i.e. untextured surfaces, with the same BRDF everywhere. We parameterize the surface by the spherical coordinates of the normal vector (α, β), using the standard convention that $(\alpha, \beta) = (0, 0)$ corresponds to the *north pole* or +Z axis. Notation used in this section is listed in figure 2, and a diagram is in figure 3. We will use two types of coordinates. Unprimed global coordinates denote angles with respect to a global reference frame. On the other hand, primed local coordinates denote angles with respect to the local reference frame, defined by the local surface normal. These two coordinate systems are related simply by a rotation, to be defined shortly.

Reflection Equation: We modify equation 1 based on our assumptions, dropping the texturing term, and using the surface normal (α, β) instead of the position $\mathbf{x}$ to parameterize B. Since $L(\mathbf{x}, \vec{\omega}_i)$ is assumed to be independent of $\mathbf{x}$, we write it as $L(\theta_i, \phi_i)$. Finally, $(\vec{\omega}_i \cdot \vec{n})$ can be written simply as $\cos\theta_i'$, the cosine of the incident angle in local coordinates.

$$B(\alpha, \beta, \theta_o', \phi_o') = \int_{\Omega_i'} L(\theta_i, \phi_i)\rho(\theta_i', \phi_i', \theta_o', \phi_o') \cos\theta_i' \, d\omega_i' \qquad (2)$$

We have mixed local (primed) and global (unprimed) coordinates. The lighting is a global function, and is naturally expressed in a global coordinate frame as a function of global angles. On the other hand, the BRDF is naturally expressed as a function of the local incident and reflected angles. When expressed in the local coordinate frame, the BRDF is the same everywhere for a homogeneous surface. Similarly, when expressed in the global coordinate frame, the lighting is the same everywhere, under the assumption of distant illumination. The reflected radiance B can be expressed conveniently in either local or global coordinates; we have used local coordinates to match the BRDF. Similarly, integration can be conveniently done over either local or global coordinates, but the upper hemisphere is easier to express in local coordinates.

We now define a transfer[1] function $\hat{\rho} = \rho\cos\theta_i'$ in order to absorb the cosine term. With this modification, equation 2 becomes

$$B(\alpha, \beta, \theta_o', \phi_o') = \int_{\Omega_i'} L(\theta_i, \phi_i)\hat{\rho}(\theta_i', \phi_i', \theta_o', \phi_o') \, d\omega_i' \qquad (3)$$

Rotations—Converting Local and Global coordinates: Local and global coordinates are related by a rotation corresponding to the surface normal (α, β). The *north pole* in local coordinates, $(0', 0')$ is the surface normal. The corresponding global coordinates are clearly (α, β). We define $R_{\alpha,\beta}$ as a rotation operator[2] on column vectors that rotates (θ_i', ϕ_i') into global coordinates, and is given by $R_{\alpha,\beta} = R_z(\beta)R_y(\alpha)$ where R_z is a rotation about the Z axis and R_y a rotation about the Y axis.

$$\begin{array}{rclcl}(\theta_i, \phi_i) & = & R_z(\beta)R_y(\alpha)(\theta_i', \phi_i') & = & R_{\alpha,\beta}(\theta_i', \phi_i') \\ (\theta_i', \phi_i') & = & R_y(-\alpha)R_z(-\beta)(\theta_i, \phi_i) & = & R_{\alpha,\beta}^{-1}(\theta_i, \phi_i)\end{array}$$

We can now write the dependence on incident angle in equation 3 entirely in global coordinates, or entirely in local coordinates.

$$\begin{array}{rcll}B(\alpha, \beta, \theta_o', \phi_o') & = & \int_{\Omega_i} L(\theta_i, \phi_i)\hat{\rho}\left(R_{\alpha,\beta}^{-1}(\theta_i, \phi_i), \theta_o', \phi_o'\right) \, d\,\omega_i & (4) \\ & = & \int_{\Omega_i'} L\left(R_{\alpha,\beta}(\theta_i', \phi_i')\right)\hat{\rho}(\theta_i', \phi_i', \theta_o', \phi_o') \, d\,\omega_i' & (5)\end{array}$$

[1] If we want the transfer function to be reciprocal, i.e. symmetric with respect to incident and outgoing angles, we may multiply both the transfer function and the reflected light field by $\cos\theta_o'$. See equation 13.

[2] For anisotropic surfaces, we need an initial rotation about Z to set the local tangent frame. We would then have rotations about Z, Y and Z—the familiar Euler-Angle parameterization. Since we are dealing with isotropic surfaces, we have ignored this initial Z rotation, which has no physical significance. It is not difficult to derive the theory for the more general anisotropic case.

Interpretation as Convolution: In the spatial domain, convolution is the result generated when a filter is *translated* over an input signal. However, we can generalize the notion of convolution to other transformations T_a, where T_a is a function of a, and write

$$(f \otimes g)(a) = \int_t f(t) g\left(T_a(t)\right)\, dt$$

When T_a is a translation by a, we obtain the standard expression for spatial convolution. When T_a is a rotation by the angle a, the above formula defines convolution in the angular domain.

Therefore, equations 4 and 5 represent rotational convolutions. Equation 4 in global coordinates states that the reflected light field at a given surface orientation corresponds to *rotating* the BRDF to that orientation, and then integrating over the upper hemisphere. The BRDF can be thought of as the filter, while the lighting is the input signal. Symmetrically, equation 5 in local coordinates states that the reflected light field at a given surface orientation may be computed by *rotating* the lighting into the local coordinate system of the BRDF, and then doing the hemispherical integration.

4.2 Spherical Harmonic Representation

For the translational case, the well-known frequency-space convolution formula is given in terms of Fourier transforms. For a general operator, an analogous formula can be obtained in terms of group representations and the associated basis functions. For translations, these basis functions are sines and cosines—the familiar Fourier basis. For rotations, the corresponding basis functions are spherical harmonics, and we now proceed to derive the frequency-space rotational convolution formula in terms of spherical harmonics.

Inui et al. [11] is a good reference for background on spherical harmonics and their relationship to rotations. Our use of spherical harmonics to represent the lighting is similar in some respects to previous methods [25] that use steerable linear basis functions. Spherical harmonics, as well as the closely related Zernike Polynomials, have been used before to represent BRDFs [3, 14, 33].

Spherical harmonics are the analog on the sphere to the Fourier basis on the line or circle. The spherical harmonic Y_{lm} is given by

$$\begin{aligned} N_{lm} &= \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}} \\ Y_{lm}(\theta,\phi) &= N_{lm} P_l^m(\cos\theta) e^{Im\phi} \end{aligned}$$

where N_{lm} is a normalization factor. In the above equation, the azimuthal dependence is expanded in terms of Fourier basis functions. The θ dependence is expanded in terms of the associated Legendre functions P_l^m. The indices obey $l \geq 0$ and $-l \leq m \leq l$.

The rotation formula for spherical harmonics is

$$Y_{lm}(R_{\alpha,\beta}(\theta_i',\phi_i')) = \sum_{m'=-l}^{l} D^l_{mm'}(\alpha) e^{Im\beta} Y_{lm'}(\theta_i',\phi_i') \tag{6}$$

The important thing to note here is that the m indices are *mixed*—a spherical harmonic after rotation must be expressed as a combination of other spherical harmonics with different m indices. However, the l indices are not mixed; rotations of spherical harmonics with order l are composed entirely of other spherical harmonics with order l. For given order l, D^l is a matrix that tells us how a spherical harmonic transforms under rotation about the y-axis, i.e. how to rewrite a rotated spherical harmonic as a linear combination of all the spherical harmonics of the same order.

We begin by expanding the lighting in global coordinates.

$$L(\theta_i,\phi_i) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} L_{lm} Y_{lm}(\theta_i,\phi_i) \tag{7}$$

Here, the coefficients L_{lm} can be computed in the standard way by integrating against the complex conjugate Y_{lm}^*

$$L_{lm} = \int_{\theta_i=0}^{\pi}\int_{\phi_i=0}^{2\pi} L(\theta_i,\phi_i) Y_{lm}^*(\theta_i,\phi_i)\, \sin\theta_i\, d\theta_i d\phi_i$$

We now represent the transfer function $\hat{\rho} = \rho\cos(\theta_i')$ in terms of spherical harmonics. Note that $\hat{\rho}$ is nonzero only over the upper hemisphere, i.e. when $\cos\theta_i' > 0$ and $\cos\theta_o' > 0$.

$$\hat{\rho}(\theta_i',\phi_i',\theta_o',\phi_o') = \sum_{l,m,p,q} \hat{\rho}_{lm,pq} Y_{lm}^*(\theta_i',\phi_i') Y_{pq}(\theta_o',\phi_o')$$

We are interested in isotropic BRDFs, which depend only on $\mid \phi_o' - \phi_i' \mid$. This implies that the BRDF is invariant with respect to adding a constant angle $\triangle\phi'$ to both incident and outgoing azimuthal angles. It can be shown from the form of the spherical harmonics that this condition forces all terms to vanish unless $m = q$. The use of the complex conjugate for Y_{lm}^* in the expansion above is to make $m = q$ instead of $m = -q$. We now write

$$\hat{\rho}(\theta_i',\phi_i',\theta_o',\phi_o') = \sum_{l,p,q} \hat{\rho}_{lq,pq} Y_{lq}^*(\theta_i',\phi_i') Y_{pq}(\theta_o',\phi_o') \tag{8}$$

Furthermore, invariance of the BRDF with respect to negating both incident and outgoing azimuthal angles requires that $\hat{\rho}_{lq,pq} = \hat{\rho}_{l(-q),p(-q)}$. Finally, we use only three indices for the BRDF.

$$\hat{\rho}_{lpq} = \hat{\rho}_{lq,pq} = \hat{\rho}_{l(-q),p(-q)}$$

To represent the reflected light field, we define a new set of orthonormal basis functions. The normalization and form of these functions are derived in the appendix. In particular, the matrix D comes from the rotation formula for spherical harmonics, equation 6. It will be convenient to first define a normalization constant.

$$\Lambda_l = \sqrt{\frac{4\pi}{2l+1}} \qquad \Lambda_l^{-1} = \sqrt{\frac{2l+1}{4\pi}}$$

The new basis functions can then be written

$$C_{lmpq}(\alpha,\beta,\theta_o',\phi_o') = \Lambda_l^{-1} D^l_{mq}(\alpha) e^{Im\beta} Y_{pq}(\theta_o',\phi_o') \tag{9}$$

The expansion of the reflected light field is now

$$B(\alpha,\beta,\theta_o',\phi_o') = \sum_{l,m,p,q} B_{lmpq} C_{lmpq}(\alpha,\beta,\theta_o',\phi_o')$$

The translational convolution theorem expresses convolution in frequency-space as a product of Fourier coefficients. For the rotational case, an analogous result is derived in the appendix, using spherical harmonics instead of complex exponentials. The *frequency-space* reflection equation (or rotational convolution formula) is a similar product of basis-function coefficients.

$$\boxed{B_{lmpq} = \Lambda_l L_{lm} \hat{\rho}_{lpq}} \tag{10}$$

5 Implications

This section explores the implications of our results for problems in inverse rendering, and works out some special cases in detail. Our theory indicates which inverse problems are tractable, as opposed to being ill-posed or ill-conditioned. Finally, we will use the insights gained to develop a new practical representation.

5.1 General Observations

Inverse BRDF: Equation 10 can be manipulated to yield

$$\hat{\rho}_{lpq} = \Lambda_l^{-1} \frac{B_{lmpq}}{L_{lm}} \tag{11}$$

We may use any index m in inverse BRDF computation. Therefore, BRDF recovery is well-posed unless the denominator vanishes for all m, i.e. all terms for some order l in the spherical harmonic expansion of the lighting vanish. In signal processing terms, if the input signal (lighting) has no amplitude along certain modes of the filter (BRDF), those modes cannot be estimated. BRDF recovery is well conditioned when the lighting contains high frequencies like directional sources, and is ill-conditioned for soft lighting.

Inverse Lighting: Equation 10 can also be manipulated to yield

$$L_{lm} = \Lambda_l^{-1} \frac{B_{lmpq}}{\hat{\rho}_{lpq}} \qquad (12)$$

Similarly as for BRDF recovery, any p, q can be used for inverse lighting. The problem is well-posed unless the denominator $\hat{\rho}_{lpq}$ vanishes for all p, q for some l. In signal processing terms, when the BRDF filter truncates certain frequencies in the input lighting signal (for instance, if it were a low-pass filter), we cannot determine those frequencies from the output signal. Inverse lighting is well-conditioned when the BRDF has high-frequency components like sharp specularities, and is ill-conditioned for diffuse surfaces.

Light Field Factorization—Lighting and BRDF: We now consider the problem of *factorizing* the light field, i.e simultaneously recovering the lighting and BRDF when both are unknown. The reflected light field is defined on a four-dimensional domain while the lighting is a function of two dimensions and the isotropic BRDF is defined on a three-dimensional domain. This seems to indicate that we have more knowns (in terms of coefficients of the reflected light field) than unknowns (lighting and BRDF coefficients).

For fixed order l, we can use known lighting coefficients L_{lm} to find unknown BRDF coefficients $\hat{\rho}_{lpq}$ and vice-versa. In fact, we need only one known nonzero lighting or BRDF coefficient to bootstrap this process. It would appear from equation 10, however, that there is an unrecoverable scale factor for each order l, corresponding to the known coefficient we require. But, we can also use reciprocity of the BRDF. To make the transfer function symmetric, we multiply it, as well as the reflected light field B, by $\cos\theta'_o$.

$$\begin{aligned} \tilde{\rho} &= \hat{\rho}\cos\theta'_o = \rho\cos\theta'_i\cos\theta'_o \\ \tilde{B} &= B\cos\theta'_o \\ \tilde{B}_{lmpq} &= \Lambda_l L_{lm}\tilde{\rho}_{lpq} \qquad (13) \end{aligned}$$

The new transfer function $\tilde{\rho}$ is symmetric with respect to incident and outgoing directions, and corresponding indices: $\tilde{\rho}_{lpq} = \tilde{\rho}_{plq}$.

There is a global scale factor we cannot recover, since $\tilde{B}$ is not affected if we multiply the lighting and divide the BRDF by the same amount. Therefore, we scale the lighting so the DC term $L_{00} = \Lambda_0^{-1} = \sqrt{1/(4\pi)}$. Now, using equations 11, 12, and 13,

$$\begin{aligned} L_{00} &= \Lambda_0^{-1} \\ \tilde{\rho}_{0p0} &= \tilde{B}_{00p0} \\ L_{lm} &= \Lambda_l^{-1}\left(\frac{\tilde{B}_{lmpq}}{\tilde{\rho}_{lpq}} = \frac{\tilde{B}_{lm00}}{\tilde{\rho}_{l00}} = \frac{\tilde{B}_{lm00}}{\tilde{\rho}_{0l0}}\right) \\ &= \Lambda_l^{-1}\frac{\tilde{B}_{lm00}}{\tilde{B}_{00l0}} \\ \tilde{\rho}_{lpq} &= \Lambda_l^{-1}\frac{\tilde{B}_{lmpq}}{L_{lm}} \\ &= \frac{\tilde{B}_{lmpq}\tilde{B}_{00l0}}{\tilde{B}_{lm00}} \end{aligned}$$

In the last line, we can use any value of m. This gives an explicit formula for the lighting and BRDF in terms of coefficients of the output light field. Therefore, up to global scale, **the reflected light field can be factored into the lighting and the BRDF**, provided the appropriate coefficients of the reflected light field do not vanish.

5.2 Special Cases

Mirror BRDF: The mirror BRDF corresponds to a gazing sphere. Just as the inverse lighting problem is easily solved in angular space in this case, we will show that it is well-posed and easily solved in frequency space. The BRDF involves a delta function,

$$\hat{\rho}(\theta'_i, \phi'_i, \theta'_o, \phi'_o) = \delta(\cos\theta'_i - \cos\theta'_o)\delta(\phi'_i - \phi'_o \pm \pi)$$

Note that the BRDF is nonzero only when $\theta'_i \leq \pi/2$ and $\theta'_o \leq \pi/2$. The coefficients for the BRDF, reflected light field, and lighting are

$$\begin{aligned} \hat{\rho}_{lpq} &= (-1)^q\delta_{lp} \\ B_{lmpq} &= \Lambda_l(-1)^q\delta_{lp}L_{lm} \\ \forall q: \quad L_{lm} &= \Lambda_l^{-1}(-1)^q B_{lmlq} \qquad (14) \end{aligned}$$

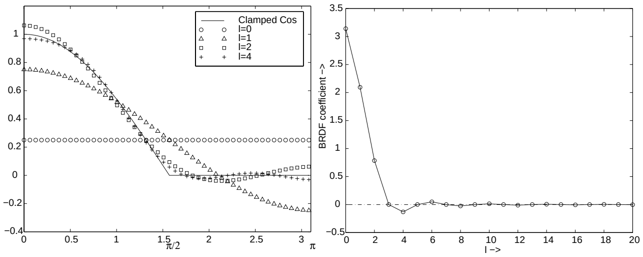


Figure 4: *Left: Successive approximations to the* clamped cosine *function by adding more spherical harmonic terms. For $l = 2$, we get a very good approximation. Right: The solid line is a plot of spherical harmonic coefficients $A_l = \Lambda_l\hat{\rho}_l$. For $l > 1$, odd terms vanish, and even terms decay rapidly.*

The factor of $(-1)^q$ is because the azimuthal angle changes by π upon reflection. We see that the lighting coefficients correspond in a very direct way to the coefficients of the reflected light field. In signal processing terminology, the inverse lighting problem is well conditioned because the frequency spectrum of a delta function remains constant with increasing order l, and does not decay.

Single Directional Source: For convenience, we position the coordinate axes so that the source is located at $+Z$, i.e. at $(0, 0)$. Because the directional source is described by a delta function, the spherical harmonic expansion coefficients are given simply by $L_{lm} = Y^*_{lm}(0)$, which vanishes for $m \neq 0$. Thus,

$$\begin{aligned} L_{lm} &= \delta_{m0}Y^*_{l0}(0) = \delta_{m0}\Lambda_l^{-1} \\ B_{lmpq} &= \delta_{m0}\hat{\rho}_{lpq} \\ \hat{\rho}_{lpq} &= B_{l0pq} \end{aligned}$$

In angular space, a single observation corresponds to a single BRDF measurement. This property is used in image-based BRDF measurement [18, 21]. We see that in frequency space, there is a similar straightforward relation between BRDF coefficients and reflected light field coefficients. BRDF recovery is well-conditioned since we are estimating the BRDF filter from its impulse response.

Lambertian BRDF: For a Lambertian object, the transfer function is a scaled *clamped cosine* function, since it is proportional to the cosine of the incident angle over the upper hemisphere when $\cos\theta'_i \geq 0$ and is equal to 0 over the lower hemisphere. Plots of spherical-harmonic fits to the clamped cosine function and the magnitude of the coefficients are shown in figure 4. Because there is no dependence on outgoing angle, we can drop the indices p and q. Further, the reflected light field is now effectively the surface radiosity function, and can be expanded[3] in spherical harmonics.

$$B(\alpha, \beta) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} B_{lm}Y_{lm}(\alpha, \beta)$$

We [29] have shown that with the definitions,

$$\begin{aligned} \hat{\rho}(\theta'_i) = \max\left[\cos\theta'_i, 0\right] &= \sum_{l=0}^{\infty}\hat{\rho}_l Y_{l0}(\theta'_i) \\ \hat{\rho}_l &= 2\pi\int_0^{\pi/2}\cos\theta'_i\, Y_{l0}(\theta'_i)\,\sin\theta'_i\, d\theta'_i \end{aligned}$$

one can derive

$$\begin{aligned} B_{lm} &= \Lambda_l\hat{\rho}_l L_{lm} \\ L_{lm} &= \Lambda_l^{-1}\frac{B_{lm}}{\hat{\rho}_l} \qquad (15) \end{aligned}$$

We define $\hat{A}_l = \Lambda_l\hat{\rho}_l$. An analytic formula for A_l may be derived [29]. It can be shown that $\hat{A}_l$ vanishes for odd values of $l > 1$,

[3] The basis functions C_{lmpq} in equation 9 become $\Lambda_l^{-1}D^l_{m0}(\alpha)e^{Im\beta}$ if we ignore output dependence, and set $q = 0$ (the BRDF is azimuthally symmetric). It can be shown that this is simply $Y_{lm}(\alpha, \beta)$. Equation 15 now follows naturally from equation 10 upon dropping indices p and q. Our previous derivation [29] was specialized to the Lambertian case, and ignored the output dependence from the onset.

and even terms fall off very rapidly as $l^{-5/2}$. More than 99% of the energy of the BRDF filter is captured by $l \leq 2$. Numerically,

$$\hat{A}_0 = 3.14 \quad \hat{A}_1 = 2.09 \quad \hat{A}_2 = 0.79 \quad \hat{A}_3 = 0 \quad \hat{A}_4 = -0.13 \quad (16)$$

Thus, the Lambertian BRDF acts like a low-pass filter, truncating or severely attenuating frequencies with $l > 2$. Therefore, from observations of a Lambertian surface, estimation of the illumination is formally ill-posed, and is well-conditioned only for the lighting coefficients with $l \leq 2$, corresponding to 9 parameters—1 for order 0 ($(l, m) = (0, 0)$), 3 for order 1 ($l = 1, -1 \leq m \leq 1$), and 5 for order 2 ($l = 2, -2 \leq m \leq 2$). This explains the ill-conditioning observed by Marschner and Greenberg [20] in trying to solve the *inverse lighting* problem from a Lambertian surface. Furthermore, for practical applications, including forward rendering [28], **the reflected light field from a Lambertian surface can be characterized using only its first 9 spherical harmonic coefficients**; lighting effects cannot produce high-frequency variation in intensity with respect to surface curvature.

Phong BRDF: The normalized Phong transfer function is

$$\hat{\rho} = \frac{s+1}{2\pi} \left(\vec{R} \cdot \vec{L} \right)^s$$

where $\vec{R}$ is the reflection of the outgoing (viewing) direction about the surface normal, $\vec{L}$ is the direction to the light source, and s is the *shininess*, or Phong exponent. The normalization ensures the Phong lobe has unit energy. Technically, we must also zero the BRDF when the light is not in the upper hemisphere. However, the Phong BRDF is not physically based, so others have often ignored this boundary effect, and we will do the same.

We now reparameterize by the reflection vector $\vec{R}$, transforming the integral over the upper hemisphere centered on the surface normal to an integral centered on $\vec{R}$. The reflection vector takes the place of the normal in the analysis, with (α, β) referring to $\vec{R}$, and $\vec{R} \cdot \vec{L} = \cos \theta_i'$. The Phong BRDF after reparameterization is mathematically analogous to the Lambertian BRDF just discussed. In fact, the properties of convolution can be used to show that **for the Phong BRDF, blurring the lighting and using a mirror BRDF is equivalent to using the real lighting and real BRDF.** This formalizes the transformation often made in rendering with environment maps [23]. Specifically, equation 15 can be written as

$$\begin{aligned} L'_{lm} &= \Lambda_l \hat{\rho}_l L_{lm} \\ \Lambda_l \hat{\rho}'_l &= 1 \\ B_{lm} &= \Lambda_l \hat{\rho}'_l L'_{lm} = L'_{lm} \end{aligned} \qquad (17)$$

Here, L'_{lm} is the blurred illumination and $\hat{\rho}'_l$ is the mirror BRDF[4].

The BRDF coefficients depend on s, and are given by

$$\hat{\rho}_l = (s+1) \int_0^{\pi/2} \left[\cos \theta_i' \right]^s Y_{l0}(\theta_i') \sin \theta_i' \, d\theta_i' \qquad (18)$$

This integral may be solved analytically. Formulae are in the appendix, and numerical plots are in figure 5.

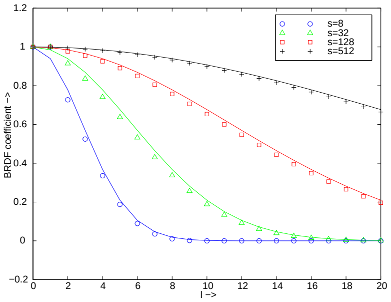


Figure 5: *Numerical plots of the Phong coefficients $\Lambda_l \hat{\rho}_l$, as defined by equation 18. The solid lines are the approximations in equation 19.*

For large s and $l \ll s$, a good approximation is

$$\Lambda_l \hat{\rho}_l \approx \exp \left[-\frac{l^2}{2s} \right] \qquad (19)$$

The coefficients fall off as a gaussian with width of order $\sqrt{s}$. The Phong BRDF behaves in the frequency domain like a gaussian filter, with the filter width controlled by the shininess. Therefore, inverse lighting calculations will be well-conditioned only up to order $\sqrt{s}$. As s approaches infinity, $\Lambda_l \hat{\rho}_l = 1$, and the frequency spectrum becomes constant, corresponding to a perfect mirror.

Microfacet BRDF: We now consider a simplified 4-parameter Torrance-Sparrow [36] model, with parameters K_d, K_s, μ and σ. This microfacet model is widely used in computer graphics.

$$\begin{aligned} \rho(\vec{\omega}_i', \vec{\omega}_o{}') &= K_d + K_s \frac{FS}{4 \cos \theta_i' \cos \theta_o'} \\ \vec{\omega}_h' &= \frac{\vec{\omega}_i' + \vec{\omega}_o'}{\| \vec{\omega}_i' + \vec{\omega}_o' \|} \\ F &= \frac{F(\mu, \theta_o')}{F(\mu, 0)} \\ S &= \frac{1}{\pi \sigma^2} \exp \left[-\left(\theta_h' / \sigma \right)^2 \right] \end{aligned}$$

The subscript h stands for the half-way vector. $F(\mu, \theta_o')$ is the Fresnel term for refractive index μ; we normalize it to be 1 at normal exitance. Actually, F depends on the angle with respect to the half-way vector; in practice, this angle is usually very close to θ_o'. For simplicity in the analysis, we have omitted the geometric attenuation factor G. In practice, this omission is not very significant except for observations made at grazing angles, which are usually assigned low confidence anyway in practical applications.

We focus on the specular component, reparameterizing by the reflection vector, as for the Phong BRDF. It will also simplify matters to fix the exitant direction, and focus on the frequency-space representation of the incident-angle dependence. Precise analytic formulae are difficult to derive, but we can make a good approximation, as shown in the appendix. For normal exitance,

$$\Lambda_l \hat{\rho}_l \approx \exp \left[-(\sigma l)^2 \right] \qquad (20)$$

For normal exitance, the specular part of the BRDF is a gaussian, so equation 20 simply states that even in the spherical-harmonic basis, the frequency spectrum of a gaussian is also gaussian, with the frequency width related to the reciprocal of the angular width.

For non-normal exitance, microfacet BRDFs are not symmetric about the reflection vector. Unlike for the Phong BRDF, there is a preferred direction, determined by the exitant angle. However, the BRDF filter is essentially symmetric about the reflected direction for small viewing angles, as well as for low frequencies l. Hence, it can be shown by Taylor-series expansions and verified numerically, that the corrections to equation 20 are small under these conditions. Finally, we approximate the effects of the Fresnel factor at non-normal exitance by multiplying our expressions by $F(\mu, \theta_o')$.

With respect to the conditioning of inverse problems, equation 20 indicates that **inverse lighting from a microfacet BRDF is well-conditioned only for frequencies up to order $l \sim \sigma^{-1}$.** Equation 20 also indicates that BRDF estimation is ill-conditioned under low-frequency lighting. For low-frequency lighting, we may apply the properties of convolution as we did for Phong BRDFs, filtering the lighting using equations 17 and 20, while using a mirror BRDF. Note that for frequencies $l << \sigma^{-1}$, the effects of this filtering are insignificant. The BRDF passes through virtually all the low-frequency energy. Therefore, **if the lighting contains only low frequencies, the reflected light field from a microfacet BRDF is essentially independent of the BRDF filter width σ^{-1}; this makes estimation of the surface roughness σ ill-conditioned.**

5.3 Practical Representation

Thus far, we have presented the theoretical foundation for, and some implications of, a frequency-space view of reflection. A sig-

[4] The formula $\Lambda_l \hat{\rho}'_l = 1$ is not identical to equation 14 since we have now reparameterized by the reflection vector. This accounts for the slightly different normalization.

nal processing approach has been used before in some other areas of computer graphics, notably the theory of aliasing. Just as a frequency-space analysis of aliasing gives many insights difficult to obtain by other means, the last two sections lead to new ways of analyzing inverse rendering problems. However, the Fourier-space theory of aliasing is not generally used directly for antialiasing. The ideal Fourier-space bandpass filter in the spatial domain, the sinc function, is usually modified for practical purposes because it has infinite extent and leads to ringing. Similarly, representing BRDFs purely as a linear combination of spherical harmonics leads to ringing. Moreover, it is difficult to compute Fourier spectra from sparse irregularly sampled data. Similarly, it is difficult to compute the reflected light field coefficients B_{lmpq} from a few photographs; we would require a very large number of input images, densely sampling the entire sphere of possible directions.

For these reasons, the frequency-space ideas must be put into practice carefully. Here, we first discuss two useful practical techniques—dual angular and frequency-space representations, and the separation of the lighting into slow and fast-varying components. Finally, we use these ideas, and the insights gained from the previous subsection, to derive a simple practical model of the reflected light field for the microfacet BRDF. This representation will be used extensively in the practical algorithms of section 6.

Dual Angular and Frequency-Space Representations: Quantities local in angular space have broad frequency spectra and vice-versa. By developing a frequency-space view of reflection, we ensure that we can use either the angular-space or frequency-space representation, or even a combination of the two. The diffuse BRDF component is slowly varying in angular-space, but is local in frequency-space, while the specular BRDF component is local in the angular domain. For representing the lighting, the frequency-space view is appropriate for the diffuse BRDF component, while the angular-space view is appropriate for the specular component.

Separation of slow and fast-varying lighting: For the angular-space description of the lighting, used in computing the reflected light field from the specular BRDF component, we separate the lighting into a *slow* varying component corresponding to low frequencies or area sources—for which we filter the lighting and use a mirror BRDF—and a *fast* varying component corresponding to high frequencies or directional sources. For the frequency-space lighting description, used for the diffuse BRDF component, this distinction need not be made since the formulae for the Lambertian BRDF are the same for both slow and fast varying components.

Model for Reflected Light Field: Our model for the reflected light field from the microfacet BRDF includes three components.

$$B = B_d + B_{s,slow} + B_{s,fast}$$

B_d is from the diffuse component of the BRDF. $B_{s,slow}$ represents specularities from the slowly-varying lighting, and $B_{s,fast}$ specular highlights from the fast varying lighting component.

To write B_d, corresponding to the Lambertian BRDF component, we use the 9 parameter frequency-space representation of the lighting. Explicitly noting $l \leq 2$, and with E being the irradiance,

$$\begin{aligned} B_d &= K_d E(\alpha, \beta) \\ E(\alpha, \beta) &= \sum_{l=0}^{2} \left(\Lambda_l \hat{\rho}_l \sum_{m=-l}^{+l} L_{lm} Y_{lm}(\alpha, \beta) \right) \end{aligned} \qquad (21)$$

The numerical values of $\Lambda_l \hat{\rho}_l$ are given in equation 16.

For $B_{s,slow}$, we filter the lighting, using equations 17 and 20, and treat the BRDF as a mirror. With $\vec{R}$ denoting the reflected direction, and L_{slow} the filtered version of the lighting, we obtain

$$B_{s,slow} = K_s F(\mu, \theta'_o) L_{slow}(\vec{R}) \qquad (22)$$

For the *fast varying* portion of the lighting—corresponding to sources of angular width $\ll \sigma$—we treat the total energy of the source, given by an integral over the (small) solid angle subtended, as located at its center, so the lighting is a delta function. $B_{s,fast}$ is given by the standard equation for the specular highlight from a directional source. The extra factor of $4\cos\theta'_o$ in the denominator as compared to equation 22 comes from the relation between differential microfacet and global solid angles.

$$\begin{aligned} B_{s,fast} &= \frac{K_s F(\mu, \theta'_o)}{4\cos\theta'_o} \sum_j T_j \\ T_j &= \exp\left[-\left(\theta'_h / \sigma\right)^2 \right] \left(\frac{L_{j,fast}}{\pi\sigma^2} \right) \end{aligned} \qquad (23)$$

The subscript j denotes a particular directional source; there could be several. Note that $L_{j,fast}$ is now the *total energy* of the source.

For BRDF estimation, it is convenient to expand out these equations, making dependence on the BRDF parameters explicit.

$$B = K_d E + K_s F(\mu, \theta'_o) \left[L_{slow}(\vec{R}) + \frac{1}{4\cos\theta'_o} \sum_j T_j(\sigma) \right] \qquad (24)$$

6 Algorithms and Results

This section shows how the theory, and in particular the model just derived in section 5.3, can be applied to a broad range of practical inverse rendering problems. We present two types of methods—algorithms that recover coefficients of a purely frequency-space description of the lighting or BRDF by representing these quantities as a sum of spherical harmonic terms, and algorithms that estimate parameters corresponding to our model of section 5.3. Section 6.1 on BRDF estimation demonstrates direct recovery of spherical harmonic BRDF coefficients, as well as estimation of parametric microfacet BRDFs using equation 24. Similarly, section 6.2 demonstrates direct recovery of spherical harmonic lighting coefficients, as well as estimation of a dual angular and frequency-space lighting description as per the model of section 5.3. Section 6.3 shows how to combine BRDF and lighting estimation techniques to simultaneously recover the lighting and BRDF parameters, when both are unknown. In this case, we do not show direct recovery of spherical harmonic coefficients, as we have thus far found this to be impractical. Finally, section 6.4 demonstrates our algorithms on geometrically complex objects, showing how it is straightforward to extend our model to handle textures and shadowing.

To test our methods, we first used homogeneous spheres[5] of different materials. Spheres are naturally parameterized with spherical coordinates, and therefore correspond directly to our theory. Later, we also used complex objects—a white cat sculpture, and a textured wooden doll—to show the generality of our algorithms.

Data Acquisition: We used a mechanical gantry to position an inward-looking Toshiba IK-TU40A CCD(x3) camera on an arc of radius 60cm. Calibration of intrinsics was done by the method of Zhang [40]. Since the camera position was computer-controlled, extrinsics were known. The mapping between pixel and radiance values was also calibrated. We acquired 60 images of the target sphere, taken at 3 degree intervals. To map from image pixels to angular coordinates $(\alpha, \beta, \theta'_o, \phi'_o)$, we used image silhouettes to find the geometric location of the center of the sphere and its radius.

Our gantry also positioned a 150W white point source along an arc. Since this arc radius (90 cm) was much larger than the sphere radii (between 1.25 and 2cm), we treated the point source as a directional light. A large area source, with 99% of its energy in low-frequency modes of order $l \leq 6$, was obtained by projecting white light on a projection screen. The lighting distribution was determined using a gazing sphere. This information was used directly for experiments assuming known illumination, and as a reference solution for experiments assuming unknown illumination.

We also used the same experimental setup, but with only the point source, to measure the BRDF of a white teflon sphere using the image-based method of Marschner et al. [21]. This independent measurement was used to verify the accuracy of our BRDF estimation algorithms under complex illumination.

[5] Ordered from the McMaster-Carr catalog http://www.mcmaster.com

6.1 Inverse BRDF with Known Lighting

Estimation of Spherical Harmonic BRDF coefficients: Spherical harmonics and Zernike polynomials have been fit [14] to measured BRDF data, but previous work has not tried to estimate coefficients directly. Since the BRDF is linear in the coefficients $\hat{\rho}_{lpq}$, we simply solve a linear system to determine $\hat{\rho}_{lpq}$.

Figure 6 compares the parametric BRDFs estimated under complex lighting to BRDFs measured using a single point source with the method of Marschner et al. [21]. As expected [14], the recovered BRDFs exhibit ringing. One way to reduce ringing is to attenuate high-frequency coefficients. According to our theory, this is equivalent to using low frequency lighting. Therefore, as seen in figure 6, images rendered with low-frequency lighting do not exhibit ringing and closely match real photographs, since only the low-frequency components of the BRDF are important. However, images rendered using directional sources show significant ringing.

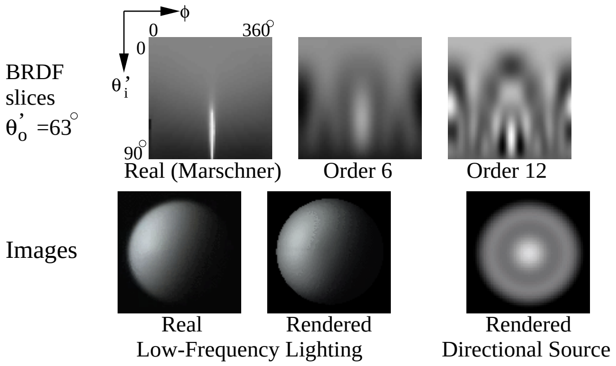


Figure 6: **Top:** *Slices of the BRDF transfer function of a teflon sphere for fixed exitant angle of* 63°. θ_i' *varies linearly from* 0° *to* 90° *from top to bottom, and* $\mid \phi_o' - \phi_i' \mid$ *linearly from* 0° *to* 360° *from left to right. The central bright feature is the specular highlight. Left is the BRDF slice independently measured using the approach of Marschner et al. [21], middle is the recovered value using a maximum order 6, and right is the recovered version for order 12. Ringing is apparent in both recovered BRDFs. The right version is sharper, but exhibits more pronounced ringing.* **Bottom:** *Left is an actual photograph; the lighting is low-frequency from a large area source. Middle is a rendering using the recovered BRDF for order 6 and the same lighting. Since the lighting is low-frequency, only low-frequency components of the BRDF are important, and the rendering appears very similar to the photograph even though the recovered BRDF does not include frequencies higher than order 6. Right is a rendering with a directional source at the viewpoint, and exhibits ringing.*

For practical applications, it is usually more convenient to recover low-parameter BRDF models since these are compact, can be estimated from fewer observations, and do not exhibit ringing. In the rest of this section, we will derive improved inverse rendering algorithms, assuming our parametric BRDF model.

Estimation of Parametric BRDF Model: We estimate BRDF parameters under general known lighting distributions using equation 24. The inputs are images that sample the reflected light field B. We perform the estimation using nested procedures. In the outer procedure, a simplex algorithm adjusts the nonlinear parameters μ and σ to minimize error with respect to image pixels. In the inner procedure, a linear problem is solved for K_d and K_s. For numerical work, we use the simplex method `e04ccc` and linear solvers `f01qcc and f01qdc` in the NAG [9] C libraries. The main difference from previous work is that equation 24 provides a principled way of accounting for all components of the lighting and BRDF, allowing for the use of general illumination conditions.

We tested our algorithm on the spheres. Since the lighting includes high and low-frequency components (a directional source and an area source), the theory predicts that parameter estimation is well-conditioned. To validate our algorithm, we compared parameters recovered under complex lighting for one of the samples, a white teflon sphere, to those obtained by fitting to the full BRDF separately measured by us using the method of Marschner et al. [21]. Unlike most previous work on BRDF estimation, we consider the Fresnel term. It should be noted that accurate estimates for the refractive index μ require correct noise-free measurements at grazing angles. Since these measurements tend to be the most error-prone, there will be small errors in the estimated values of μ for some materials. Nevertheless, we find the Fresnel term important for reproducing accurate specular highlights at oblique angles.

Parameter	Our Method	Fit to Data
Reflectance	0.86	0.87
$K_d/(K_d+K_s)$	0.89	0.91
$K_s/(K_d+K_s)$	0.11	0.09
μ	1.78	1.85
σ	0.12	0.13
RMS	9.3%	8.5%

Figure 7: *Comparison of BRDF parameters recovered by our algorithm under complex lighting to those fit to measurements made by the method of Marschner et al. [21].*

The results in figure 7 show that the estimates of BRDF parameters from our method are quite accurate, and there is only a small increase in the error-of-fit when using parameters recovered by our algorithm to fit the measured BRDF. We also determined percentage RMS errors between images rendered using recovered BRDFs and real photographs to be between 5 and 10%. A visual comparison is shown in the first and third rows of figure 12. All these results indicate that, as expected theoretically, we can accurately estimate BRDFs even under complex lighting.

6.2 Inverse Lighting with Known BRDF

Previous methods for estimating the lighting have been developed only for the special cases of mirror BRDFs (a gazing sphere), Lambertian BRDFs (Marschner and Greenberg [20]), and when shadows are present (Sato et al. [30]). Previous methods [20, 30] have also required regularization using penalty terms with user-specified weights, and have been limited by the computational complexity of their formulations to a coarse discretization of the sphere. We present two new algorithms for curved surfaces with general BRDFs. The first method directly recovers spherical harmonic lighting coefficients L_{lm}. The second algorithm estimates parameters of the dual angular and frequency-space lighting model of section 5.3. This method requires no explicit regularization, and yields high-resolution results that are sharper than those from the first algorithm, but is more difficult to extend to concave surfaces.

The theory tells us that inverse lighting is ill-conditioned for high-frequencies. Therefore, we will recover only low-frequency continuous lighting distributions, and will not explicitly account for directional sources, i.e. we assume that $B_{s,fast} = 0$. The reflected light field is essentially independent of the surface roughness σ under these conditions, so our algorithms do not explicitly use σ. The theory predicts that the recovered illumination will be a filtered version of the real lighting. Directional sources will appear as continuous distributions of angular width approximately σ.

Estimation of Spherical Harmonic Lighting coefficients: We represent the lighting by coefficients L_{lm} with $l \leq l^*$, and solve a linear least-squares system for L_{lm}. The first term in parentheses below corresponds to B_d, and the second to $B_{s,slow}$. The cutoff l^* is used for regularization, and should be of order $l^* \sim \sigma^{-1}$.

$$B = \sum_{l=0}^{l^*} \sum_{m=-l}^{l} L_{lm} \left(K_d \Lambda_l \hat{\rho}_l Y_{lm}(\alpha, \beta) + K_s F Y_{lm}(\theta_R, \phi_R) \right) \quad (25)$$

Estimation of Parametric Dual Lighting Model: Another approach is to estimate the dual angular and frequency-space lighting model of section 5.3. Our algorithm is based on subtracting out the diffuse component B_d of the reflected light field. After this, we treat the object as a mirror sphere, recovering a high-resolution angular-space version of the illumination from the specular component alone. To determine B_d, we need only the 9 lowest frequency-space coefficients L_{lm} with $l \leq 2$. Our algorithm uses the following methods to convert between angular and frequency-space:

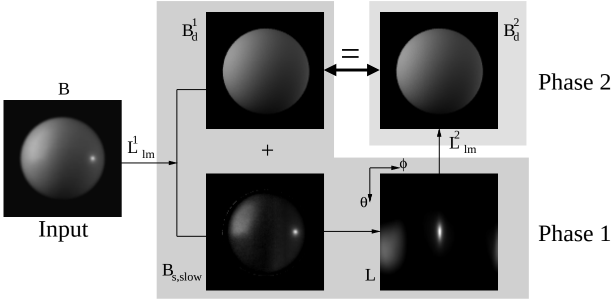


Figure 8: *Estimation of dual lighting representation. In phase 1, we use frequency-space parameters L^1_{lm} to compute diffuse component B^1_d. This is subtracted from the input image, leaving the specular component, from which the angular-space lighting is found. In phase 2, we compute coefficients L^2_{lm}, which can be used to determine B^2_d. The consistency condition is that $B^1_d = B^2_d$ or $L^1_{lm} = L^2_{lm}$. In this and all subsequent figures, the lighting is visualized by unwrapping the sphere so θ ranges in equal increments from 0 to π from top to bottom, and ϕ ranges in equal increments from 0 to 2π from left to right (so the image wraps around in the horizontal direction).*

1. **9 parameters to High-Resolution Lighting:** The inputs to phase 1 are the coefficients L^1_{lm}. These suffice to find B^1_d by equation 21. Since we assumed that $B_{s,fast} = 0$,

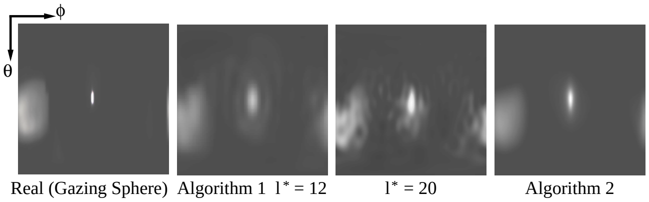
$$\begin{aligned} B_{s,slow} &= K_s F(\mu, \theta_o') L_{slow}(\vec{R}) \; = B - B^1_d(L^1_{lm}) \\ L_{slow}(\vec{R}) &= \frac{B - B^1_d(L^1_{lm})}{K_s F(\mu, \theta_o')} \end{aligned}$$

   We assume the BRDF parameters are known, and B is the input to the algorithm, so the right-hand side can be evaluated.

2. **High-Resolution Lighting to 9 parameters:** Using the angular space values L found from the first phase, we can easily find the 9 frequency-space parameters of the lighting L^2_{lm}.

Now, assume we run phase 1 (with inputs L^1_{lm}) and phase 2 (with outputs L^2_{lm}) sequentially. The *consistency condition* is that $L^1_{lm} = L^2_{lm}$—converting from frequency to angular to frequency space must not change the result. Equivalently, the computed diffuse components must match, i.e. $B^1_d(L^1_{lm}) = B^2_d(L^2_{lm})$. This is illustrated in figure 8. Since everything is linear in terms of the lighting coefficients, the consistency condition reduces to a system of 9 simultaneous equations. After solving for L_{lm}, we run phase 1 to determine the high-resolution lighting in angular space.

Figure 9 compares the methods to each other, and to a reference solution from a gazing sphere. Both algorithms give reasonably accurate results. As predicted by the theory, high-frequency components are filtered by the roughness σ. In the first method, involving direct recovery of L_{lm}, there will still be some residual energy for $l > l^*$. Since we regularize by not considering higher frequencies—we could increase l^*, but this makes the result noisier—the recovered lighting is somewhat blurred compared to our dual angular and frequency-space algorithm (second method).



Real (Gazing Sphere) Algorithm 1 $l^* = 12$ $l^* = 20$ Algorithm 2

Figure 9: *Comparison of inverse lighting methods. From left to right, real lighting (from a gazing sphere), recovered illumination by direct estimation of spherical harmonic coefficients with $l^* = 12$ and $l^* = 20$, and estimation of dual angular and frequency-space lighting model. To make the artifacts more apparent, we have set 0 to gray. The results from the dual algorithm are sharper, but still somewhat blurred because of filtering by σ. A small amount of ringing occurs for direct coefficient recovery, and can be seen for $l^* = 12$. Using $l^* = 20$ makes the solution very noisy.*

6.3 Factorization—Unknown Lighting and BRDF

We can combine the inverse-BRDF and inverse-lighting methods to *factor* the reflected light field, simultaneously recovering the lighting and BRDF when both are unknown. Therefore, we are able to recover BRDFs of curved surfaces under unknown complex illumination, something which has not previously been demonstrated. There is an unrecoverable global scale factor, so we set $K_d + K_s = 1$; we cannot find absolute reflectance. Also, the theory predicts that for low-frequency lighting, estimation of the surface roughness σ is ill-conditioned—blurring the lighting while sharpening the BRDF does not significantly change the reflected light field. However, for high-frequency lighting, this ambiguity can be removed. We will use a single manually specified directional source in the recovered lighting distribution to estimate σ.

Algorithm: The algorithm consists of nested procedures. In the outer loop, we effectively solve an inverse-BRDF problem—a nonlinear simplex algorithm adjusts the BRDF parameters to minimize error with respect to image pixels. Since $K_d + K_s = 1$, and σ will not be solved for till after the lighting and other BRDF parameters have been recovered, there are only 2 free parameters, K_s and μ. In the inner procedure, a linear problem is solved to estimate the lighting for a given set of BRDF parameters, using the methods of the previous subsection. Pseudocode is given below.

```
global $B_{input}$                              // Input images
global $K_d$,$K_s$,$\mu$,$\sigma$                    // BRDF parameters
global $L$                                      // Lighting
procedure Factor
    Minimize($K_s$,$\mu$,ObjFun)                  // Simplex Method
    $\sigma$ = FindRoughness($L$)                     // Figure 10, Equation 26
function ObjFun($K_s$,$\mu$)
    $K_d = 1 - K_s$                                // $K_d + K_s = 1$
    $L$ = Lighting($K_d$,$K_s$,$\mu$)                 // Inverse Lighting
    $B_{pred}$ = Predict($L$,$K_d$,$K_s$,$\mu$)           // Predicted Light Field
    return RMS($B_{input}$,$B_{pred}$)               // RMS Error
```

Finding σ using a directional source: If a directional source is present—and manually specified by us in the recovered lighting—we can estimate σ by equating specular components predicted by equations 22 and 23 for the center, i.e. brightest point, of the light source at normal exitance. An illustration is in figure 10.

$$L_{cen} \approx \frac{L_{total}}{4\pi\sigma^2} \tag{26}$$

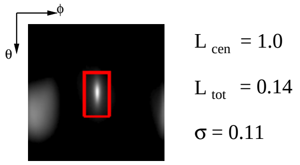


Figure 10: *We manually specify (red box) the region corresponding to the directional source in a visualization of the lighting. The algorithm then determines L_{cen}, the intensity at the center (brightest point), L_{tot}, the total energy integrated over the region specified by the red box, and computes σ using equation 26. The method does not depend on the size of the red box—provided it encloses the entire (filtered) source—nor the precise shape into which the source is filtered in the recovered lighting.*

Results: We used the method of this subsection—with the dual angular and frequency-space algorithm for inverse lighting—to factor the light field for the spheres, simultaneously estimating the BRDF and lighting. The same setup and lighting were used for all the spheres so we could compare the recovered illumination.

We see from figure 11 that the BRDF estimates under unknown lighting are accurate. Absolute errors are small, compared to parameters recovered under known lighting. The only significant anomalies are the slightly low values for the refractive index μ—caused because we cannot know the high-frequency lighting components, which are necessary for more accurately estimating the Fresnel term. We are also able to estimate a filtered version of the lighting. As shown in figure 12, the recovered lighting distributions from all the samples are largely consistent. As predicted by the theory, the directional source is spread out to different extents depending on how rough the surface is, i.e. the value of σ. Finally, figure 12 shows that rendered images using the estimated lighting and BRDF are almost indistinguishable from real photographs.

Material	K_d		K_s		μ		σ	
	Known	Unknown	Known	Unknown	Known	Unknown	Known	Unknown
Teflon	0.89	0.87	0.11	0.13	1.78	1.48	0.12	0.14
Delrin	0.87	0.88	0.13	0.12	1.44	1.35	0.10	0.11
Neoprene Rubber	0.92	0.93	0.08	0.07	1.49	1.34	0.10	0.10
Sandblasted Steel	0.20	0.14	0.80	0.86			0.20	0.19
Bronze	(.15,.08,.05)	(.09,.07,.07)	(.85,.68,.59)	(.91,.69,.55)			0.12	0.10
Painted	(.62,.71,.62)	(.67,.75,.64)	0.29	0.25	1.38	1.15	0.15	0.15

Figure 11: *BRDFs of various spheres, recovered under known (section 6.1) and unknown (section 6.3) lighting. The reported values are normalized so $K_d + K_s = 1$. RGB values are reported for colored objects. We see that K_s is much higher for the more specular metallic spheres, and that σ is especially high for the rough sandblasted sphere. The Fresnel effect is very close to 1 for metals, so we do not consider the Fresnel term for these spheres.*

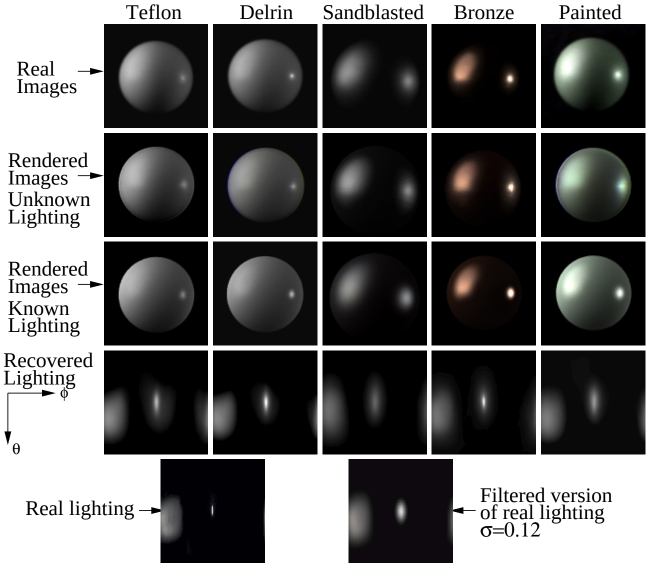


Figure 12: *Spheres rendered using BRDFs estimated under known (section 6.1) and unknown (section 6.3) lighting. The algorithm in section 6.3 also recovers the lighting. Since there is an unknown global scale, we scale the recovered lighting distributions in order to compare them. The recovered illumination is largely consistent between all samples, and is similar to a filtered version of the real lighting. As predicted by the theory, the different roughnesses σ cause the directional source to be spread out to different extents. The filtered source is slightly elongated or asymmetric because the microfacet BRDF is not completely symmetric about the reflection vector.*

6.4 Complex Objects—Texture and Shadowing

We now demonstrate our algorithms on objects with complex geometry, and discuss extensions to handle concave surfaces and textured objects. Although the theory is developed for homogeneous surfaces, our algorithms can be extended to textured objects simply by letting the BRDF parameters be functions of surface position. It would appear that concave regions, where one part of the surface may shadow another, are a more serious problem since our theory is developed for convex objects and assumes no self-shadowing. However, using our new practical model of section 5.3, we will see that the extensions necessary mainly just involve checking for shadowing of the reflected ray and directional sources, which are routine operations in a raytracer.

Shadowing—Concave Surfaces: In our practical model, the reflected light field consists of 3 parts—B_d, $B_{s,slow}$, and $B_{s,fast}$. $B_{s,slow}$ depends on $L_{slow}(\vec{R})$, the slowly-varying component of the lighting evaluated at the reflection vector. Our model allows us to approximate the effects of shadowing simply by checking if the reflected ray is shadowed. The other components are handled in the standard manner. To consider shadowing when computing $B_{s,fast}$, corresponding to specularities from directional sources, we check if these sources are shadowed. B_d depends on the irradiance E, which should now be computed in the more conventional angular-space way by integrating the scene lighting while considering visibility, instead of using the 9-parameter lighting approximation of equation 21. It should be emphasized that in all cases, the corrections for visibility depend only on object geometry, and can be precomputed for each point on the object using a ray tracer.

For parametric BRDF estimation, we modify each component of equation 24 to consider visibility, as discussed above. Our first inverse lighting method, that directly recovers the coefficients L_{lm}, is modified similarly. In equation 25, we check if the reflected ray is shadowed, and consider shadowing when computing the irradiance due to each Y_{lm}. Note that B is still a linear combination of the lighting coefficients, so we will still solve a linear system for L_{lm}. However, it is difficult to extend our dual angular and frequency-space method for inverse lighting to handle concave surfaces because B_d no longer depends only on the 9 lighting coefficients L_{lm} with $l \leq 2$. For light field factorization, we simply extend both the inverse-BRDF and inverse-lighting methods as discussed.

A white cat sculpture was used to test our algorithms on complex geometric objects that include concavities. Geometry was acquired using a Cyberware range scanner and aligned to the images by manually specifying correspondences. The lighting was slightly more complex than that for the *spheres* experiment; we used a second directional source in addition to the area source.

To show that we can recover BRDFs using a small number of images, we used only 3 input photographs. We recovered BRDFs under both known lighting, using the method of section 6.1, and unknown lighting—using the factorization method of section 6.3, with the inverse lighting component being direct recovery of spherical harmonic coefficients using $l^* = 12$. Comparisons of photographs and renderings are in figures 1 and 13. BRDF and lighting parameters are tabulated in figure 14. This experiment indicates that our methods for BRDF recovery under known and unknown lighting are consistent, and are accurate even for complex lighting and geometry. The rendered images are very close to the original photographs, even under viewing and lighting conditions not used for BRDF recovery. The most prominent artifacts are because of imprecise geometric alignment and insufficient geometric resolution. For instance, since our geometric model does not include the eyelids of the cat, that feature is missing from the rendered images.

Textured BRDFs: Since the theory shows that factorization of lighting and texture is ambiguous, we consider only recovery of textured BRDFs under known lighting. It is fairly straightforward to allow $K_d(\mathbf{x})$ and $K_s(\mathbf{x})$ to be described by textures that depend on surface position $\mathbf{x}$. In the inner procedure of the parametric BRDF estimation algorithm of section 6.1, we simply solve a separate linear problem for each point $\mathbf{x}$ to estimate $K_d(\mathbf{x})$ and $K_s(\mathbf{x})$.

As an experimental test, we used a wooden doll. We compared the real input photographs with images rendered using the recovered textured BRDF. We also took a photograph of the same object under a single directional source and compared this to a rendering using the textured BRDF recovered under complex illumination. The results in figure 15 show that our renderings closely resemble real photographs. The main artifact is blurring of texture because of geometry-image misregistration.

7 Conclusions and Future Work

This paper has developed a signal-processing framework for inverse rendering. The qualitative observation that the reflected light field is a convolution of the lighting and BRDF has been formalized mathematically. We have shown in frequency-space why a gazing sphere is well-suited for recovering the lighting—the frequency spectrum of the mirror BRDF (a delta function) is constant—and

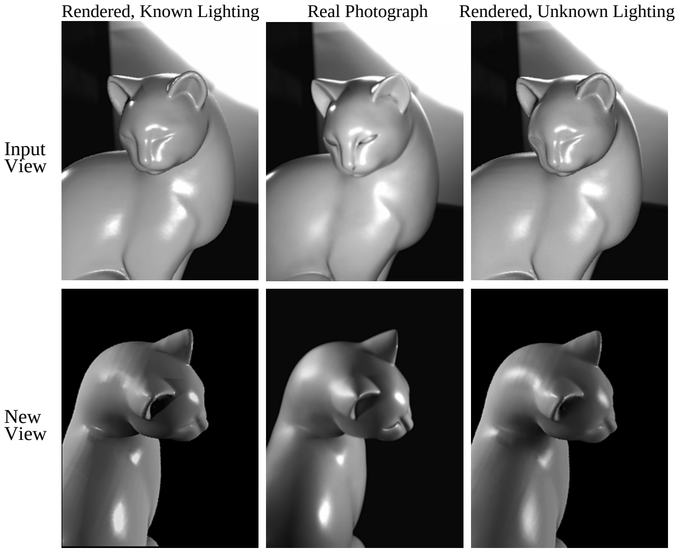


Figure 13: *Comparison of real photographs (middle column) to images rendered using BRDFs recovered under known lighting (left column), and using BRDFs (and lighting) estimated under unknown lighting (right column). The top row is one of the 3 input views. The bottom row is a new view, not used for BRDF estimation. Note that in the top row, we have composited the left and right renderings over the same background as the middle photograph in order to make a meaningful comparison.*

Parameter	Known Lighting	Unknown Lighting
BRDF Parameters		
K_d	0.88	0.90
K_s	0.12	0.10
μ	1.68	1.47
σ	0.12	0.14
Lighting Coefficients (l,m)		
(l,m) = (0,0)	0.68	0.68
(1,-1)	-0.06	-0.02
(1,0)	-0.17	-0.15
(1,1)	-0.02	-0.06
(2,-2)	0.10	0.04
(2,-1)	0.03	0.09
(2,0)	-0.55	-0.51
(2,1)	0.30	0.28
(2,2)	0.84	0.88

Figure 14: *BRDF and lighting parameters for the cat sculpture. We see good agreement between BRDF parameter values recovered with known and unknown lighting, showing our methods are consistent. Note that we normalize so $K_d + K_s = 1$. We may also check the accuracy of the recovered lighting. Since there is an unknown global scale for the recovered values, we report normalized lighting coefficient values for the first 9 spherical harmonic coefficients (in real form), which are the most important, because they significantly affect the diffuse component of the BRDF.*

why a directional source is well-suited for recovering the BRDF—we are estimating the BRDF filter by considering its impulse response. The conditioning properties and well-posedness of BRDF and lighting estimation under various conditions have been derived, as well as an explicit formula for *factoring* the reflected light field into the lighting and BRDF. The ill-conditioning observed by Marschner and Greenberg [20] in estimating the lighting from a Lambertian surface has been explained, and we have shown that factorization of lighting effects and low-frequency texture is ambiguous. All these results indicate that the theory provides a useful analytical tool for studying the properties of inverse problems.

The insights gained from the theory also lead to a new practical representation. We can numerically represent quantities in angular or frequency space, depending on where they are more local. This leads to new algorithms which are often expressed in a combination of angular and frequency-space. We can determine which BRDF and lighting parameters are important, and can handle the various components appropriately. For BRDF estimation, the parametric recovery algorithms of Yu and Malik [39], Sato and Ikeuchi [31], and Love [17]—which are designed specifically for natural lighting—can be seen as special cases of this general approach; they treat sunlight (high-frequency) and skylight (low-frequency) separately. We provide a general framework for arbitrary illumination, and also determine conditions under which parameter recovery is robust. For instance, our theory predicts that estimation of σ is ill-conditioned on a cloudy day, with only low-frequency lighting. Our framework can also be applied to developing new frequency-space algorithms to estimate the lighting from objects with general BRDFs. The use of frequency-space naturally handles continuous lighting distributions. Our dual angular and frequency-space algorithm effectively reduces the problem for general BRDFs to that for a gazing sphere, requires no explicit regularization, and allows much higher resolutions to be obtained than with previous purely angular-space methods [20, 30]. Finally, we demonstrate a method for *factoring* the light field to simultaneously estimate the lighting and BRDF. This allows us to estimate BRDFs of geometrically complex objects under unknown general lighting, which has not previously been demonstrated.

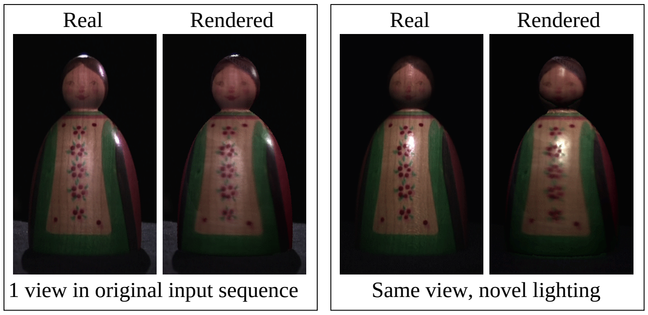


Figure 15: *Recovering textured BRDFs under complex lighting. The rendered images closely resemble the real photographs, even under novel lighting.*

We have only scratched the surface of possible applications. In the future, it is likely that many more algorithms can be derived using the basic approaches outlined here. Possible algorithmic improvements include extending the consistency condition for inverse lighting so we can use color-space methods [13] to help separate diffuse and specular components for colored objects. Finally, while we have discussed only inverse rendering applications, we believe the convolution-based approach is of theoretical and practical importance in many other areas of computer graphics. We have already shown [28] how to use the 9 term irradiance approximation for efficient forward rendering of diffuse objects with environment maps, and we believe there are many further applications.

Acknowledgements: We are grateful to Marc Levoy for many helpful initial discussions regarding both the interpretation of reflection as convolution and the practical issues in inverse rendering, as well as for reading a draft of the paper. We thank Szymon Rusinkiewicz for many suggestions regarding this research, for reading early versions of the paper, and for being ever willing to assist us in debugging problems with the gantry. Steve Marschner also deserves thanks for detailed comments on many early drafts. Jean Gleason at Bal-tec assisted in supplying us with our mirror sphere. Finally, we thank John Parissenti at Polytec Products for much useful advice on obtaining uniform spheres, and for getting one of our steel spheres sandblasted. The cat is a range scan of the sculpture *Serenity* by Sue Dawes. The work described in this paper was supported in part by a Hodgson-Reed Stanford graduate fellowship and NSF ITR grant #0085864 "Interacting with the Visual World."

References

[1] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. In *International Conference on Computer Vision*, 2001.

[2] R. Bastos, K. Hoff, W. Wynn, and A. Lastra. Increased photorealism for interactive architectural walkthroughs. In *I3D 99*, pages 183–190, 1999.

[3] B. Cabral, N. Max, and R. Springmeyer. Bidirectional reflection functions from surface bump maps. In *SIGGRAPH 87*, pages 273–281, 1987.

[4] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH 96*, pages 303–312, 1996.

[5] K. Dana, B. Ginneken, S. Nayar, and J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, January 1999.

[6] P. Debevec, T. Hawkins, C. Tchou, H.P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *SIGGRAPH 00*, pages 145–156.

[7] R. Dror, E. Adelson, and A. Willsky. Estimating surface reflectance properties from images under unknown illumination. In *SPIE Photonics West: Human Vision and Electronic Imaging VI*, 2001.

[8] M. D'Zmura. *Computational Models of Visual Processing*, chapter Shading Ambiguity: Reflectance and Illumination, pages 187–207. MIT Press, 1991.

[9] Numerical Algorithms Group. *NAG C Library Manual, Mark 5*. 1999.

[10] K. Ikeuchi and K. Sato. Determining reflectance properties of an object using range and brightness images. *PAMI*, 13(11):1139–1153, 1991.

[11] T. Inui, Y. Tanabe, and Y. Onodera. *Group theory and its applications in physics*. Springer Verlag, 1990.

[12] K. F. Karner, H. Mayer, and M. Gervautz. An image based measurement system for anisotropic reflection. *Computer Graphics Forum*, 15(3):119–128, 1996.

[13] G.J. Klinker, S.A. Shafer, and T. Kanade. The measurement of highlights in color images. *IJCV*, 2(1):7–32, 1988.

[14] J. J. Koenderink and A. J. van Doorn. Phenomenological description of bidirectional surface reflection. *JOSA A*, 15(11):2903–2912, 1998.

[15] E. Land and J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61(1):1–11, 1971.

[16] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3D scanning of large statues. In *SIGGRAPH 00*, pages 131–144.

[17] R.C. Love. *Surface Reflection Model Estimation from Naturally Illuminated Image Sequences*. PhD thesis, Leeds, 1997.

[18] R. Lu, J. Koenderink, and A. Kappers. Optical properties (bidirectional reflection distribution functions) of velvet. *Applied Optics*, 37(25):5974–5984, 1998.

[19] T.M. MacRobert. *Spherical harmonics; an elementary treatise on harmonic functions, with applications*. Dover Publications, 1948.

[20] S.R. Marschner and D.P. Greenberg. Inverse lighting for photography. In *Fifth Color Imaging Conference*, pages 262–265, 1997.

[21] S.R. Marschner, S.H. Westin, E.P.F. Lafortune, and K.E. Torrance. Image-Based BRDF measurement. *Applied Optics*, 39(16):2592–2600, 2000.

[22] N. McCormick. Inverse radiative transfer problems: a review. *Nuclear Science and Engineering*, 112:185–198, 1992.

[23] G. Miller and C. Hoffman. Illumination and reflection maps: Simulated objects in simulated and real environments. *SIGGRAPH 84 Advanced Computer Graphics Animation seminar notes*, 1984.

[24] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. *Geometric Considerations and Nomenclature for Reflectance*. National Bureau of Standards (US), 1977.

[25] J. Nimeroff, E. Simoncelli, and J. Dorsey. Efficient re-rendering of naturally illuminated environments. In *EGWR 94*, pages 359–373.

[26] R.W. Preisendorfer. *Hydrologic Optics*. US Dept Commerce, 1976.

[27] R. Ramamoorthi and P. Hanrahan. Analysis of planar light fields from homogeneous convex curved surfaces under distant illumination. In *SPIE Photonics West: Human Vision and Electronic Imaging VI*, 2001.

[28] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *SIGGRAPH 01*, 2001.

[29] R. Ramamoorthi and P. Hanrahan. On the relationship between radiance and irradiance: Determining the illumination from images of a convex lambertian object. *To appear, Journal of the Optical Society of America A*, 2001.

[30] I. Sato, Y. Sato, and K. Ikeuchi. Illumination distribution from brightness in shadows: adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. In *ICCV 99*, pages 875 – 882, 1999.

[31] Y. Sato and K. Ikeuchi. Reflectance analysis under solar illumination. Technical Report CMU-CS-94-221, Carnegie Mellon University, 1994.

[32] Y. Sato, M. D. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. In *SIGGRAPH 97*, pages 379–388, 1997.

[33] F. X. Sillion, J. Arvo, S. H. Westin, and D. Greenberg. A global illumination solution for general reflectance distributions. In *SIGGRAPH 91*, pages 187–196.

[34] S.R.Marschner. *Inverse Rendering for Computer Graphics*. PhD thesis, Cornell, 1998.

[35] S. Tominaga and N. Tanaka. Estimating reflection parameters from a single color image. *IEEE Computer Graphics & Applications*, 20(5):58–66, 2000.

[36] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *JOSA*, 57(9):1105–1114, 1967.

[37] G. J. Ward. Measuring and modeling anisotropic reflection. In *SIGGRAPH 92*, pages 265–272, 1992.

[38] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *SIGGRAPH 99*, pages 215–224, 1999.

[39] Y. Yu and J. Malik. Recovering photometric properties of architectural scenes from photographs. In *SIGGRAPH 98*, pages 207–218, 1998.

[40] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

Appendix: Mathematical Derivations

Frequency-space Reflection Equation: We rewrite the spherical harmonic lighting expansion in equation 7, applying the rotation formula in equation 6.

$$
\begin{aligned}
L(\theta_i, \phi_i) &= L\left(R_{\alpha,\beta}(\theta_i', \phi_i')\right) = \sum_{l,m} L_{lm} Y_{lm}\left(R_{\alpha,\beta}(\theta_i', \phi_i')\right) \\
&= \sum_{l,m,m'} L_{lm} D^l_{mm'}(\alpha) e^{Im\beta} Y_{lm'}(\theta_i', \phi_i')
\end{aligned}
$$

Next, we reproduce the BRDF expansion in equation 8, using l' instead of l, in order not to conflict with the index l, already in use for the lighting.

$$
\hat{\rho}(\theta_i', \phi_i', \theta_o', \phi_o') = \sum_{l',p,q} \hat{\rho}_{l'pq} Y^*_{l'q}(\theta_i', \phi_i') Y_{pq}(\theta_o', \phi_o')
$$

We can now write out the reflection integral, equation 5. The integrand is simply the product of the above two results for the lighting and BRDF. After taking terms not depending on (θ_i', ϕ_i') outside the integral, we obtain

$$
\begin{aligned}
&B(\alpha, \beta, \theta_o', \phi_o') = \\
&\sum_{l,l',m,m',p,q} [L_{lm} \hat{\rho}_{l'pq} D^l_{mm'}(\alpha) e^{Im\beta} Y_{pq}(\theta_o', \phi_o') \cdot \\
&\int_{\Omega_i'} Y_{lm'}(\theta_i', \phi_i') Y^*_{l'q}(\theta_i', \phi_i') \, d\,\omega_i']
\end{aligned}
$$

By orthonormality of the spherical harmonics, the value of the integral is $\delta_{ll'}\delta_{m'q}$. We can therefore set $m' = q$ and $l' = l$ and eliminate l' and m'.

$$
B(\alpha, \beta, \theta_o', \phi_o') = \sum_{l,m,p,q} \left(L_{lm} \hat{\rho}_{lpq}\right) \left(D^l_{mq}(\alpha) e^{Im\beta} Y_{pq}(\theta_o', \phi_o')\right)
$$

From this, we derive the form of C_{lmpq} in equation 9, and the frequency-space reflection formula, equation 10. The prefactor Λ_l^{-1} in equation 9, and the resulting factor Λ_l in equation 10, is for normalization. The orthonormality of the basis functions C_{lmpq}, as well as the value of the normalization constant Λ_l, can be derived from the orthogonality of the group representation matrices D^l (equation 7.73 of [11]).

Phong BRDF: We substitute $u = \cos\theta_i'$ in equation 18, and note that $Y_{l0}(\theta_i') = \Lambda_l^{-1} P_l(\cos\theta_i')$, where P_l is the legendre polynomial of order l.

$$
\hat{\rho}_l = \Lambda_l^{-1}(s+1) \int_0^1 u^s P_l(u) \, d\,u
$$

An analytic formula is given by MacRobert [19] in equations 19 and 20 of chapter 5.

$$
\begin{aligned}
\text{ODD } l \quad \Lambda_l \hat{\rho}_l &= \frac{(s+1)(s-1)(s-3)\ldots(s-l+2)}{(s+l+1)(s+l-1)\ldots(s+2)} \\
\text{EVEN } l \quad \Lambda_l \hat{\rho}_l &= \frac{s(s-2)\ldots(s-l+2)}{(s+l+1)(s+l-1)\ldots(s+3)}
\end{aligned}
$$

This can be expressed using Euler's Gamma function, which for positive integers is simply the factorial function, $\Gamma(n) = (n-1)!$. Neglecting constant terms, we obtain for large s and $s > l - 1$,

$$
\Lambda_l \hat{\rho}_l = \frac{\left[\Gamma\left(\frac{s}{2}\right)\right]^2}{\Gamma\left(\frac{s}{2} - \frac{l}{2}\right) \Gamma\left(\frac{s}{2} + \frac{l}{2}\right)}
$$

If $l \ll s$, we can expand the logarithm of this function in a Taylor series about $l = 0$.

$$
\log(\Lambda_l \hat{\rho}_l) = -l^2 \left(\frac{1}{2s} - \frac{1}{2s^2}\right) + O\left(\frac{l^4}{s^2}\right)
$$

For large s, $1/s \gg 1/s^2$, and we derive equation 19.

Microfacet BRDF: Consider normal exitance, and ignore Fresnel variation with incident angle. There is no azimuthal dependence, and $\theta_h' = \theta_i'/2$.

$$
\hat{\rho}_l = 2\pi \int_0^{\pi/2} \frac{\exp\left[-\theta_i'^2/4\sigma^2\right]}{4\pi\sigma^2} Y_{l0}(\theta_i') \, \sin\theta_i' \, d\,\theta_i' \tag{27}
$$

The expansion of $Y_{l0}(t)$ near $t = 0$ for small l is

$$
Y_{l0}(t) = \Lambda_l^{-1}\left(1 - \frac{l(l+1)}{4} t^2 + O(t^4)\right) \tag{28}
$$

The asymptotic form of $Y_{l0}(t)$ near $t = 0$ for large l is

$$
Y_{l0}(t) \sim \Lambda_l^{-1}\left(\frac{1}{\sqrt{t}} \cos[(l + 1/2)t - \pi/4]\right) \tag{29}
$$

To integrate equation 27, we substitute $\theta_i' = 2\sigma u$. Assuming $\sigma \ll 1$, the upper limit of the integral becomes infinite, and $\sin\theta_i' \, d\,\theta_i' = \theta_i' \, d\,\theta_i' = 4\sigma^2 u \, d\,u$.

$$
\hat{\rho}_l = \int_0^\infty 2e^{-u^2} Y_{l0}(2\sigma u) u \, d\,u
$$

We therefore set $t = 2\sigma u$ in equations 28 and 29. When $\sigma l \ll 1$, we use equation 28.

$$
\begin{aligned}
\Lambda_l \hat{\rho}_l &= \left(\int_0^\infty 2ue^{-u^2} \, d\,u - (\sigma l)^2 \int_0^\infty 2u^3 e^{-u^2} \, d\,u\right) \\
&= 1 - (\sigma l)^2 + O\left([\sigma l]^4\right)
\end{aligned}
$$

Note that these are the first terms in the Taylor series of $\exp[-(\sigma l)^2]$. When $\sigma l \gg 1$, we use equation 29 to obtain (Φ is a phase that encapsulates the lower-order terms):

$$
\Lambda_l \hat{\rho}_l \sim \int_0^\infty e^{-u^2} \sqrt{u} \cos[(2\sigma l)u + \Phi] \, d\,u
$$

The dominant term can be shown to be $\exp[-(2\sigma l)^2/4] = \exp[-(\sigma l)^2]$. Therefore, we can simply use $\exp[-(\sigma l)^2]$ as a valid approximation in both domains, giving rise to equation 20. We have also verified this result numerically.

SIGGRAPH 2002 Course:
Acquiring Material Models Using Inverse Rendering

Measuring Spatial Variation with Complex BRDFs

Hendrik P. A. Lensch

March 28, 2002

These course notes are closely related to the attached paper "Image-Based Reconstruction of Spatially Varying Materials" [7] which has been published at the Eurograhpics Workshop on Rendering 2001.

The overall pipeline for the image-based measurement of spatially varying materials for one object consists of the following steps:

- acquisition of 3D geometry
- capturing of high dynamic range images with different illumination
- resampling
- clustering of surface points into clusters of similar materials
- fitting a BRDF model to the material clusters
- representing the BRDF of each point as a weighted sum of cluster BRDFs.

The course notes will emphasize the basic idea of image-based measurement of spatially varying materials. The notes will also give some insight into practical issues on the acquisition process complementing the techniques described in the paper.

1 Spatially Varying Materials

Real-world objects are usually composed of a number of different materials that often show subtle changes even within a single material. Photorealistic rendering of such objects requires accurate measurements of the reflection properties of each material, as well as the spatially varying effects.

The process of measuring the BRDF of a homogeneous material is already quite demanding. There are several methods to measure the BRDF directly.

1.1 BRDF Measurement for Homogeneous Materials

Given a flat sample surface, a gonioreflectometer samples the BRDF by varying the position of a light source and a sensor over the hemisphere. At each position the reflectivity is measured for exactly one pair of incident and exitant directions. A dense sampling of the BRDF thus requires moving the light source and the sensor to a large number of different positions which is very time consuming. In Figure 1.1a the principle setup of a gonioreflectometer is depicted. The same effect can be obtained by keeping either the light source or the sensor fix while tilting the surface and changing its orientation instead (see Figure 1.1b).

A more efficient technique to acquire BRDFs of homogeneous materials has been proposed by Marschner et al. [8]. Here, the sensor is replaced by a digital camera which acts as a sensor array. Instead of a planar sample the technique uses a curved surface. Since the surface normal for a curved object is different at each point a single image of the object lit by a point light source captures a large number of different incident and exitant light direc-

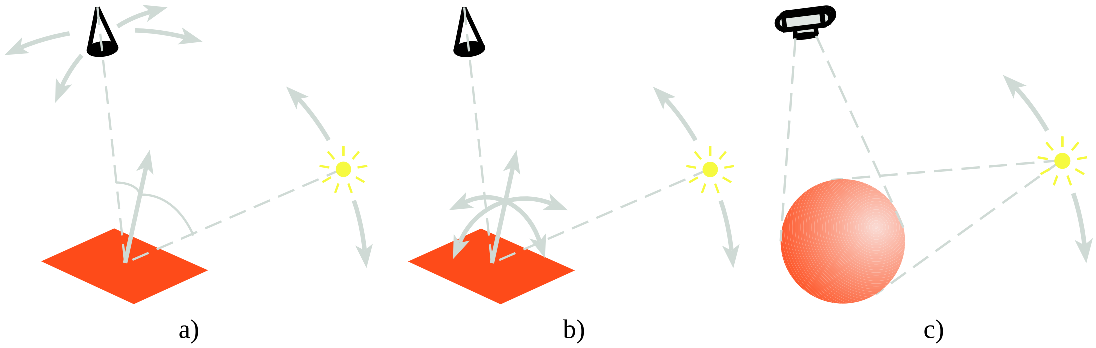


Figure 1: *a) Principle setup of a gonioreflectometer: The BRDF of a flat, homogeneous surface is sampled by varying the position of the light source and the detecting sensor. b) Instead of moving the sensor, the surface sample is tilted, varying the surface normal. c) In image-based BRDF measurement the sensor is replaced by a camera which captures many samples of the BRDF at one shot since the surface is curved, resulting in a different normal at each pixel.*

tions with respect to the surface. Only a small number of images is needed for a dense sampling of the BRDF.

1.2 Inhomogeneous Materials

For a real world object it is typically not adequate to assume that it consists of only a single homogeneous material. The material and thus the reflection properties may be different for each point on the surface.

In simple cases, this variation is modeled simply as an albedo map or texture which only represents the view independent diffuse part of the BRDF [9, 10]. The specular part is assumed to be constant or to vary only smoothly over the surface [11, 15, 14]. An albedo map can be obtained from a couple of images taken of the object.

One could also think of using a gonioreflectometer to exactly measure spatially varying reflection properties. Using a gonioreflectometer for an arbitrarily shaped object, one would need to measure the BRDF at each position on the surface separately which clearly would be very tedious.

A more suitable approach has been taken by Debevec et al. [1]. For the purpose of acquiring the reflection field they constructed the *light stage* where a point light source spins around the object while a video camera takes several hundreds of images for a fixed view. For one view the incident light directions are densely sampled. The images of several camera positions can be combined to collect multiple samples of the viewing direction. Using multiple cameras at the same time allows rendering from arbitrary views, however the amount of data to be acquired increases drastically.

1.3 Image-Based Measurement of Material Clusters

In many cases actually far less images are necessary to measure the spatially varying properties. Based on the image-based BRDF measurement technique we have developed a method that captures the spatial variation of reflection properties across the surface just from a small number of images (around 30).

The goal is to determine the BRDF of each single surface point or texel. However, instead of trying to measure the BRDF for each texel separately we first measure the BRDF of whole groups or clusters of surface points.

The approach is motivated by the fact that many man-made objects and even quite a lot of naturally grown objects are composed of only a few visually distinguishable materials or mixtures of them. The partitioning of surface points into clusters of similar material is the first step of the algorithm. A complete description of how the clustering is performed can be found in the attached paper.

Given a partitioning of the surface points into a finite number of clusters, a single image of a curved object already yields a more or less dense sampling of the clusters'

BRDFs, like in the image-based measurement approach for homogeneous materials. Multiple images have to be taken to increase the sampling rate concerning the incident and outgoing direction, and at the same time to ensure that each point on the surface is visible in at least one image. The number of images necessary for the sampling rate in many cases matches the number of images required to fulfill the visibility constraint.

1.4 Parameterization

Knowing the BRDF of the clusters allows us to synthesize images of the object from arbitrary views with arbitrary illumination. Directly using the sampled BRDF data to render the object is not so easy though, because no regular sampling pattern is present. Instead, samples are scattered over the four dimensional domain of the BRDF. Reconstructing the correct reflection for a given incident and exitant direction would require scattered data interpolation.

To avoid this reconstruction for every synthesized image we chose to represent the BRDF by a mathematical model. Using a BRDF model has the further advantage that it also shows a smooth behavior for regions of the BRDF's 4D domain where actually no samples have been collected.

Several different BRDF models have been proposed (e.g. [5, 12, 13]) with different parameters describing the reflection properties. These parameters have to be determined from our input data for every cluster by a nonlinear optimization. Reconstructing the reflection for a given incident and outgoing direction then boils down to evaluating the BRDF model.

1.5 Adding Spatial Variation

Unfortunately, we are not done by just measuring and representing the BRDFs of material clusters. Assuming that all texels within one cluster show the same reflection properties does not result in realistic images since real surfaces exhibit subtle changes like dust or impurities even within the same material. Additionally, smooth transitions between materials should be modeled.

Again we have to determine the BRDF of each surface point separately. However, the small number of images yields only a very small number of samples of the BRDF of single texels (3-10 samples on average). These are clearly too sparse to represent the BRDF directly using scattered data interpolation. Even the fitting of parameters of a BRDF model will not work reliably for so few samples.

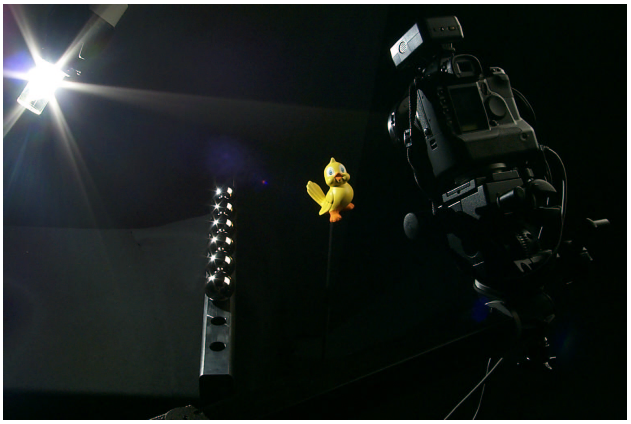


Figure 2: *Picture of the acquisition setup in a photo studio covered with dark felt (from left to right): point light source, metal spheres for light source tracking, object to be measured, digital still camera.*

In order to model the spatial detail we represent the BRDF at a single texel as a mixture of the BRDFs that were measured for the clusters. As the cluster BRDFs are blended by a weighted sum, we now have to determine the optimal weights for each texel. Most often a very small set of samples is enough to compute these weights reliably.

Even if there are too few samples for a texel a plausible BRDF can be obtained since the sum is formed by reasonable BRDFs determined for whole groups of texels with a large number of samples. Further details on the representation can be found in the attached paper.

2 Acquisition

Before fitting any parameters to samples, the samples have to be acquired. In this section we give a brief overview over the necessary steps, consisting of 3D scanning, calibration, high-dynamic range imaging, and so on.

We obtain the 3D models with a structured light 3D scanner and a computer tomography scanner both gen-

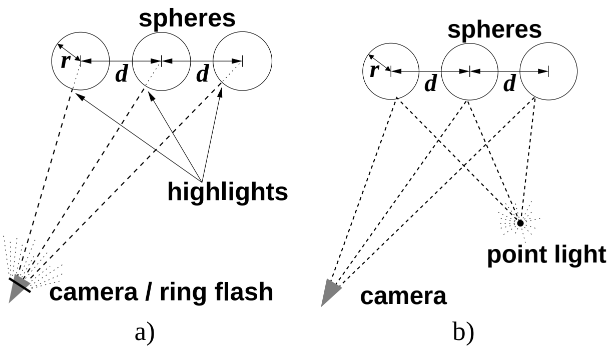


Figure 3: *a) the ring flash mounted on the camera yields a highlight in the center of the spheres. b) rays from the camera to the light source highlights will be reflected to the point light.*

erating dense triangle meshes. The triangle meshes are smoothed [3, 4], manually cleaned, and decimated.

All images are acquired in a measurement lab (see Figure 2) using a professional digital camera. An HMI metal halide bulb serves as point light source for the BRDF measurements. The interior of the photo studio is covered with dark and diffusely reflecting felt to minimize the influence of the environment on the measurements.

Several views of each object are captured with different camera and light source positions. Light source and camera are positioned manually, which, however, is easily possible since only a few different views are required. For each view we acquire three sets of images: one image of the object's silhouette to register the 3D model with the images, and two images to recover the light source position (see Section 2.1). We then acquire a high dynamic range image of the object lit by the point light source by taking a series of photographs with varying exposure time [2]. One high dynamic range image of a gray card with known camera and light position is taken in order to allow for an absolute calibration of the reflectance.

In addition, a series of calibration images of a checkerboard pattern is taken whenever the lens settings are changed. The calibration method proposed by Zhang [16] is used to recover the intrinsic camera parameters.

To register the images with the 3D model we use a silhouette-based method [6] that yields the camera position relative to the object.

2.1 Recovering the Light Source Position

In order to recover the position of the point light source a geometric approach was used which requires no user interaction. Six steel spheres of known, equal diameter are used (see Figure 2). A metal fixture manufactured using a CNC milling machine ensures that the spheres are lying exactly on a straight line, and are separated by the same known distance.

For each view two images of the spheres are acquired. One view shows only the reflection of the point light source in the spheres. For the second view a ring flash mounted on the camera lens produces a highlight on the center of each sphere. The exact centers of these reflections in the images are determined by automatically fitting ellipses to the highlights.

Given the intrinsic parameters of the camera, the pixel coordinates of the reflections of the ring flash define rays in space on which the centers of the spheres are located (see Figure 3a). Knowing the distance between the spheres we can triangulate their positions relative to the camera.

Now we send rays from the camera to the positions of the light source highlights and reflect the rays off the spheres. The light source position is located at the intersection of the rays (see Figure 3b). To increase the stability of this method more than the three necessary spheres are used and a least squares approximation is computed.

After measuring the light source position the metal spheres are covered with black cloth in order to avoid disturbing the acquisition of the HDR images.

3 Resampling

After acquisition of the geometric model, high dynamic range image recovery, and registration, it is necessary to merge the acquired data for further processing. For each point on the model's surface we collect the available information from all the images where the point is visible and lit.

How the points for which the data is collected are determined in order to best match the resolution of the input images and what data actually is stored per texel is described in the attached paper (Section 4).

Here, we want to point out some important issues that have to be considered when resampling the data of the input images and combining it with the geometry.

3.1 Discarding Data at Depth Discontinuities

In order to increase the quality of the resampled data it is sometimes necessary to discard some of the input data. Especially near depth discontinuities the resampled data is prone to registration errors and imprecise 3D geometry. If the 3D model is not perfectly aligned with the 2D image the part of the surface that is visible in one pixel may not correspond to the surface part predicted by the 3D model. In the case of depth discontinuities the visible part and the predicted part will not even be adjacent (see Figure 4a). Radiance samples would then be assigned to completely wrong surface points or lumitexels.

Furthermore, since a sensor element of the camera always integrates over a finite area, the radiance values reported at depth discontinuities are never reliable even if the registration with the 3D model were perfect. Thus, it is necessary to discard the image data at depth discontinuities as depicted in Figure 4b.

The depth discontinuities are detected using the following approach: A depth map of the registered 3D model is rendered and subsequently blurred using an averaging filter. This changes the depth values of pixels near depth discontinuities while pixels showing a flat surface will not be affected. Regions where the filtered depth map deviates more than a small threshold from the original one will not be considered for further processing. The threshold can be computed given the filter size and the difference of two adjacent depth values that should be detected as a discontinuity.

The same approach also applies to the shadowing problem. Here, depth discontinuities result in shadow boundaries whose position can only be determined up to some uncertainty. Hence, also pixels near shadow boundaries have to be discarded (Figure 4c and d). They can be determined by a filtered shadow map.

The results of removing samples at depth discontinuities are displayed in Figure 5. Note that the dark stripes on the dress and across the hand have been removed by this step.

4 Conclusion

Putting it all together, the complete pipeline for the BRDF measurement of spatially varying materials consists of the following steps: After the acquisition of geometry and images, and the resampling, the surface points are clustered into groups showing similar reflection properties. The parameters of a BRDF model are fit to the acquired samples of each cluster. Finally, spatial variation is obtained by representing the BRDF of each single point as a weighted sum of BRDFs of the cluster. Please refer to the attached paper for presentations of results.

References

[1] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the Reflectance Field of a Human Face. In *Proc. SIGGRAPH*, pages 145–156, July 2000. ISBN 1-58113-208-5.

[2] P. Debevec and J. Malik. Recovering High Dynamic Range Radiance Maps from Photographs. In *Proc. SIGGRAPH*, pages 369–378, August 1997.

[3] M. Garland and P. Heckbert. Surface Simplification Using Quadric Error Metrics. In *Proc. SIGGRAPH*, pages 209–216, August 1997.

[4] L. Kobbelt. Discrete fairing. In *Proc. of the 7th IMA Conf. on the Mathematics of Surfaces*, pages 101–131, 1996.

[5] E. Lafortune, S. Foo, K. Torrance, and D. Greenberg. Non-Linear Approximation of Reflectance Functions. In *Proc. SIGGRAPH*, pages 117–126, August 1997.

[6] H. Lensch, W. Heidrich, and H.-P. Seidel. Automated Texture Registration and Stitching for Real World Models. In *Pacific Graphics '00*, pages 317–326, October 2000.

[7] H.P.A. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel. Image-Based Reconstruction of Spatially Varying Materials. In Steven Gortler and Karol Myszkowski, editors, *Proceedings of the*

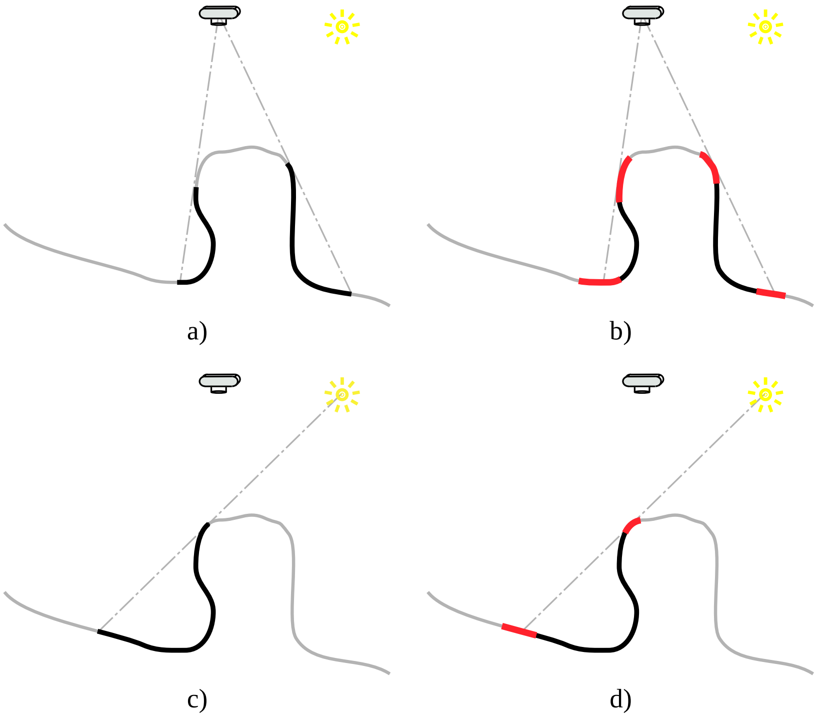


Figure 4: *Due to registration errors and imprecise 3D geometry depth discontinuities may cause problems. Data near depth discontinuities have to be discarded because it is not clear whether the front or the back part of the surface is visible at a point close to depth discontinuities (a and b). The same also applies to shadow boundaries where it is not clear where the shadowed area ends.*

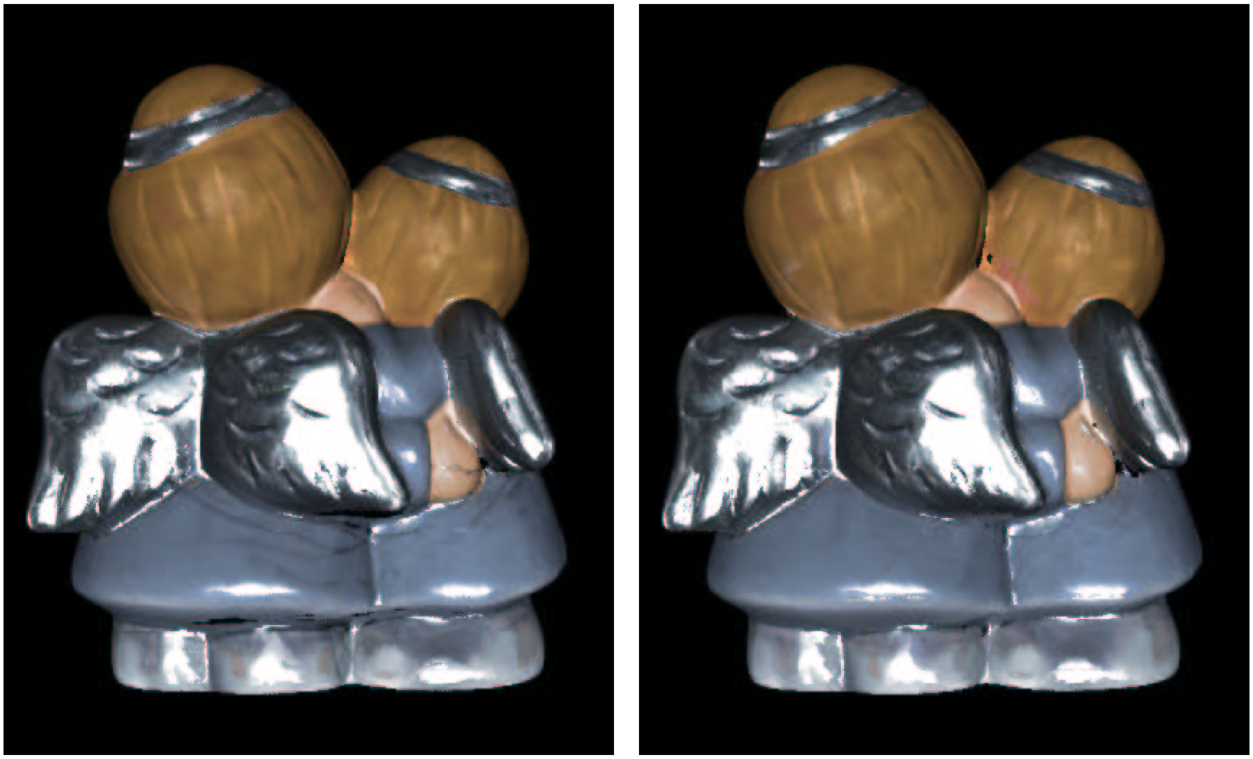


Figure 5: *Left: Dark stripes on the dress and on the hand are due to depth discontinuities. Right: By discarding samples at depth discontinuities and shadow boundaries these effects have been removed.*

12th Eurographics Workshop on Rendering, London, Great Britain, 2001. Springer.

[8] S. Marschner, S. Westin, E. Lafortune, K. Torrance, and D. Greenberg. Image-based BRDF Measurement Including Human Skin. In *10th Eurographics Workshop on Rendering*, pages 131–144, June 1999.

[9] K. Matsushita and T. Kaneko. Efficient and Handy Texture Mapping on 3D Surfaces. *Computer Graphics Forum*, 18(3):349–358, September 1999.

[10] P. Neugebauer and K. Klein. Texturing 3D Models of Real World Objects from Multiple Unregistered Photographic Views. *Computer Graphics Forum*, 18(3):245–256, September 1999.

[11] Y. Sato, M. Wheeler, and K. Ikeuchi. Object Shape and Reflectance Modeling from Observation. In *Proc. SIGGRAPH*, pages 379–388, August 1997.

[12] K. Torrance and E. Sparrow. Theory for Off-Specular Reflection from Roughened Surfaces. *Journal of Optical Society of America*, 57(9), 1967.

[13] G. Ward Larson. Measuring and Modeling Anisotropic Reflection. In *Proc. SIGGRAPH*, pages 265–272, July 1992.

[14] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse Global Illumination: Recovering Reflectance Models of Real Scenes From Photographs. In *Proc. SIGGRAPH*, pages 215–224, August 1999.

[15] Y. Yu and J. Malik. Recovering Photometric Properties of Architectural Scenes from Photographs. In *Proc. SIGGRAPH*, pages 207–218, July 1998.

[16] Z. Zhang. Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. In *Int. Conf. on Computer Vision*, pages 666–673, September 1999.

SAN ANTONIO

SIGGRAPH

2002

SAN ANTONIO

SIGGRAPH

2002

Course: Acquiring Material Models Using Inverse Rendering

Measuring Spatial Variation with Complex BRDFs

Hendrik P.A. Lensch

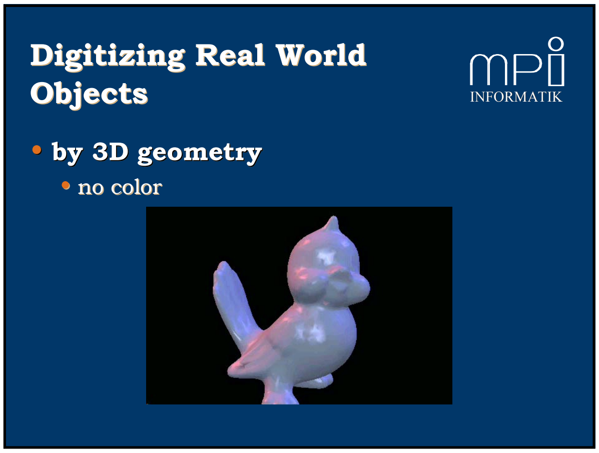
Max-Planck-Institut für Informatik

Saarbrücken (Germany)

Digitizing Real World Objects

mpi INFORMATIK

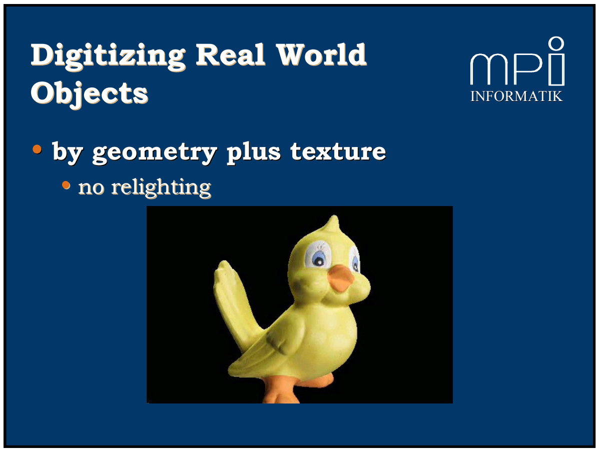
- by images
  - no interaction



Digitizing Real World Objects

mpi INFORMATIK

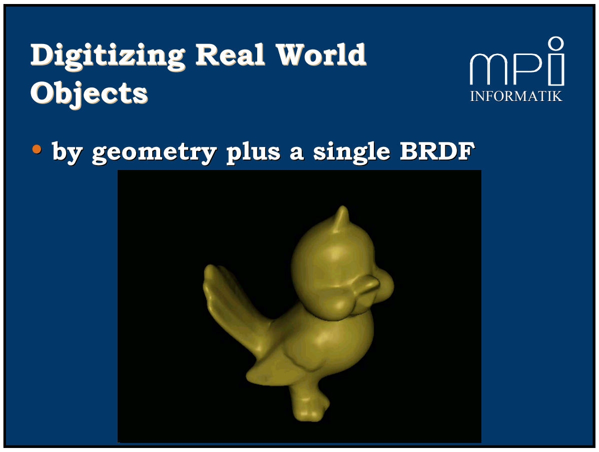
- by 3D geometry
  - no color



Digitizing Real World Objects

mpi INFORMATIK

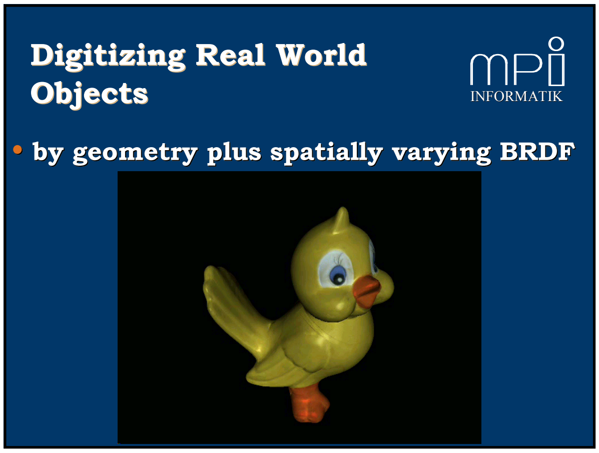
- by geometry plus texture
  - no relighting



Digitizing Real World Objects

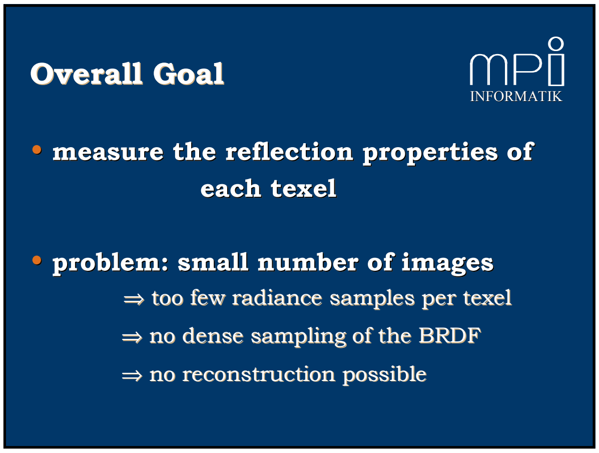
mpi INFORMATIK

- by geometry plus a single BRDF



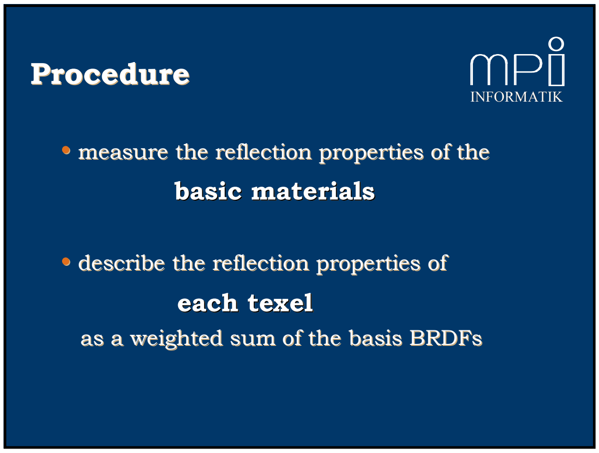
Digitizing Real World Objects

- by geometry plus spatially varying BRDF



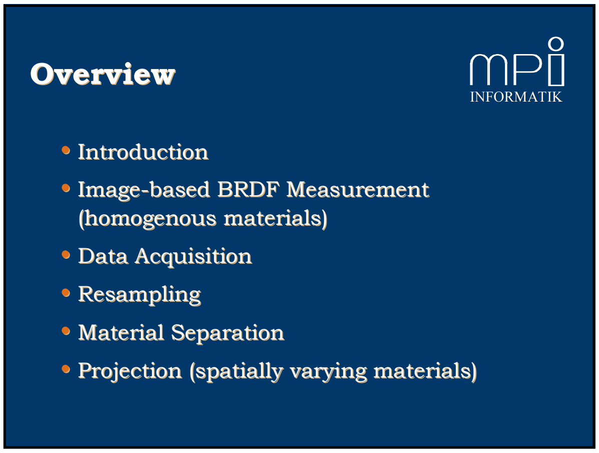
Overall Goal

- measure the reflection properties of each texel
- problem: small number of images
  - ⇒ too few radiance samples per texel
  - ⇒ no dense sampling of the BRDF
  - ⇒ no reconstruction possible



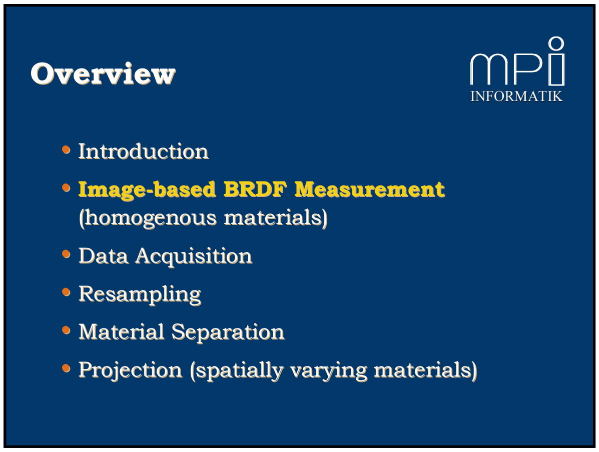
Procedure

- measure the reflection properties of the **basic materials**
- describe the reflection properties of **each texel** as a weighted sum of the basis BRDFs



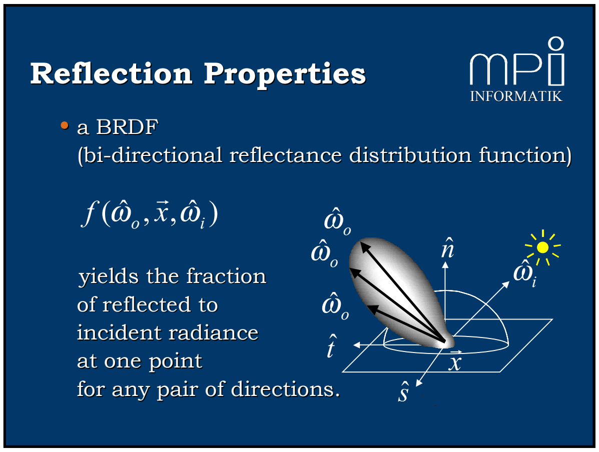
Overview

- Introduction
- Image-based BRDF Measurement (homogenous materials)
- Data Acquisition
- Resampling
- Material Separation
- Projection (spatially varying materials)



Overview

- Introduction
- **Image-based BRDF Measurement** (homogenous materials)
- Data Acquisition
- Resampling
- Material Separation
- Projection (spatially varying materials)

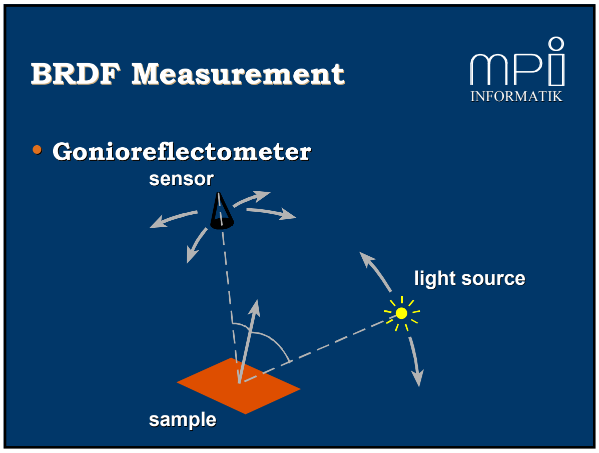


Reflection Properties

- a BRDF (bi-directional reflectance distribution function)

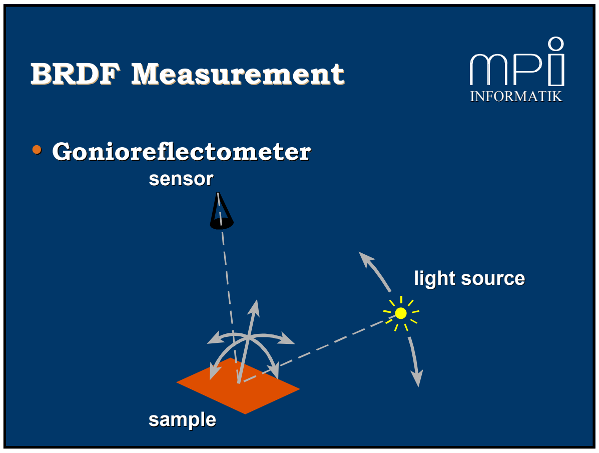
$f(\hat{\omega}_o, \vec{x}, \hat{\omega}_i)$

yields the fraction of reflected to incident radiance at one point for any pair of directions.



BRDF Measurement

- Gonioreflectometer



BRDF Measurement

- Gonioreflectometer

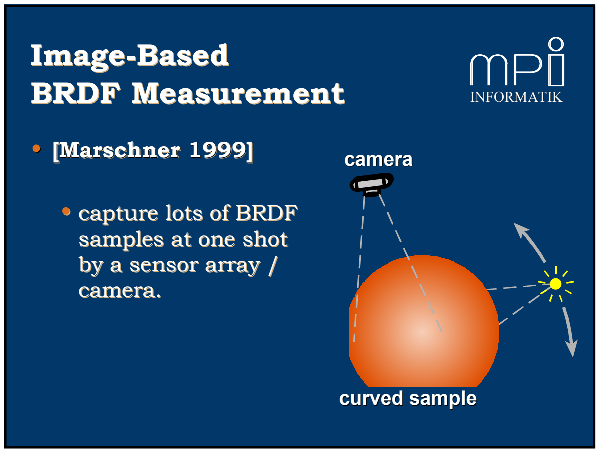
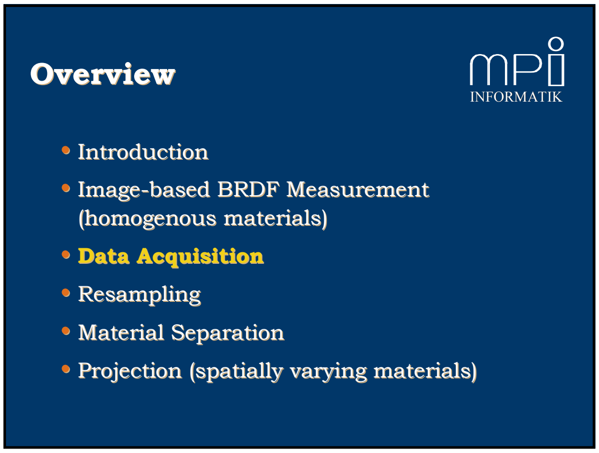


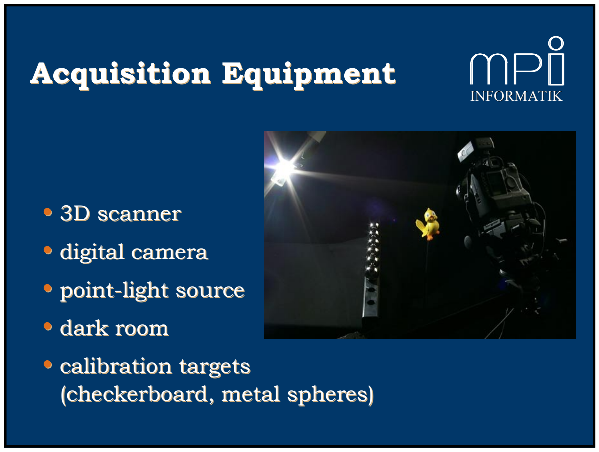
Image-Based BRDF Measurement

- [Marschner 1999]
  - capture lots of BRDF samples at one shot by a sensor array / camera.



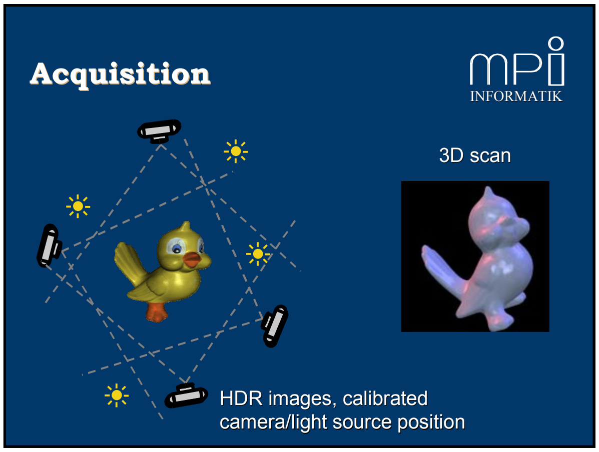
Overview

- Introduction
- Image-based BRDF Measurement (homogenous materials)
- **Data Acquisition**
- Resampling
- Material Separation
- Projection (spatially varying materials)

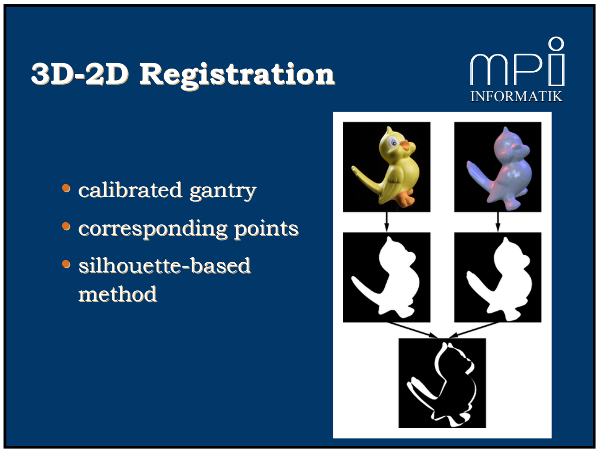


Acquisition Equipment

- 3D scanner
- digital camera
- point-light source
- dark room
- calibration targets (checkerboard, metal spheres)

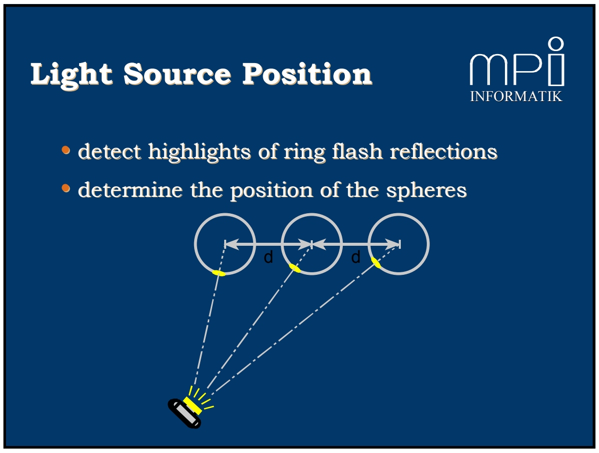


Acquisition



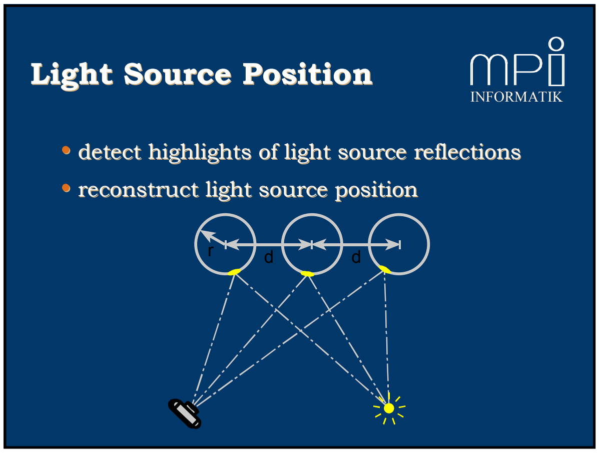
3D-2D Registration

- calibrated gantry
- corresponding points
- silhouette-based method



Light Source Position

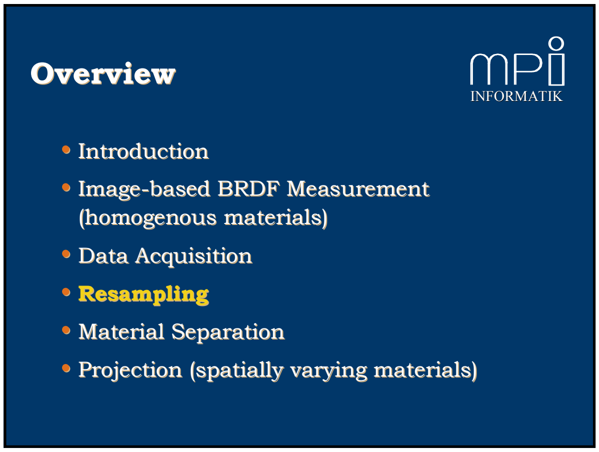
- detect highlights of ring flash reflections
- determine the position of the spheres



Light Source Position

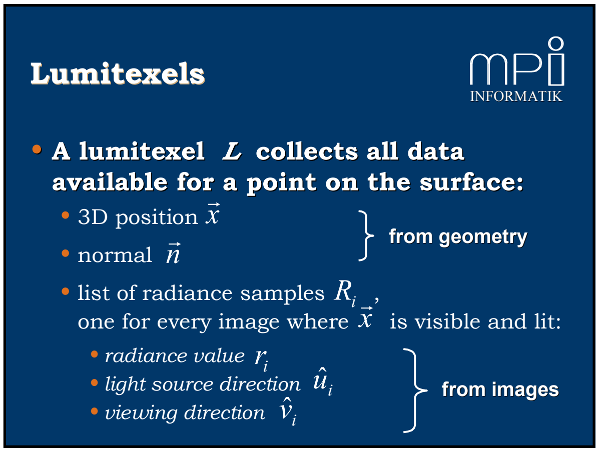
- detect highlights of light source reflections
- reconstruct light source position

Light Source Position



Overview

- Introduction
- Image-based BRDF Measurement (homogenous materials)
- Data Acquisition
- **Resampling**
- Material Separation
- Projection (spatially varying materials)

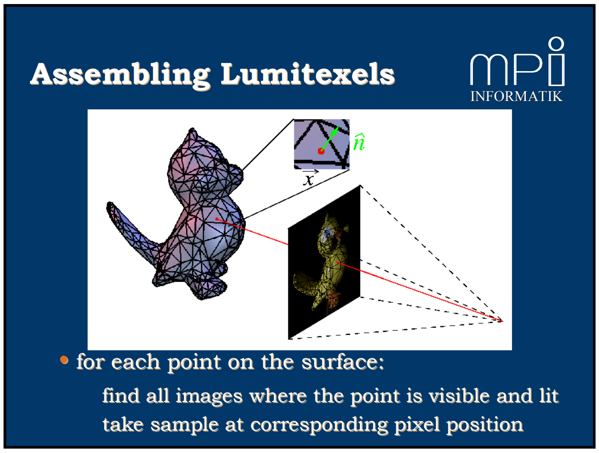


Lumitexels

- **A lumitexel L collects all data available for a point on the surface:**
  - 3D position $\vec{x}$
  - normal $\vec{n}$

  } **from geometry**
  - list of radiance samples $R_{i,\vec{x}}$, one for every image where $\vec{x}$ is visible and lit:
    - *radiance value* r_i
    - *light source direction* $\hat{u}_i$
    - *viewing direction* $\hat{v}_i$

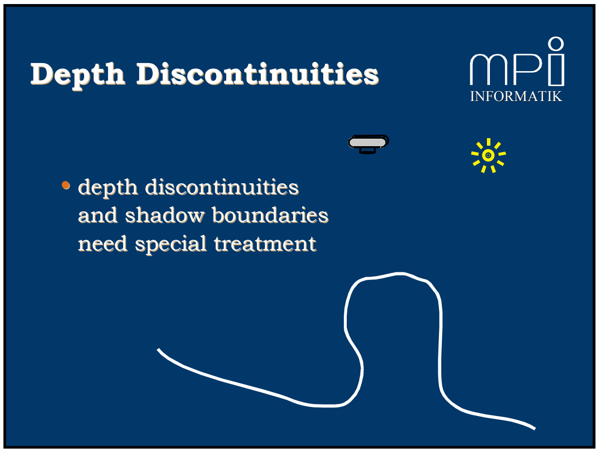
    } **from images**



Assembling Lumitexels

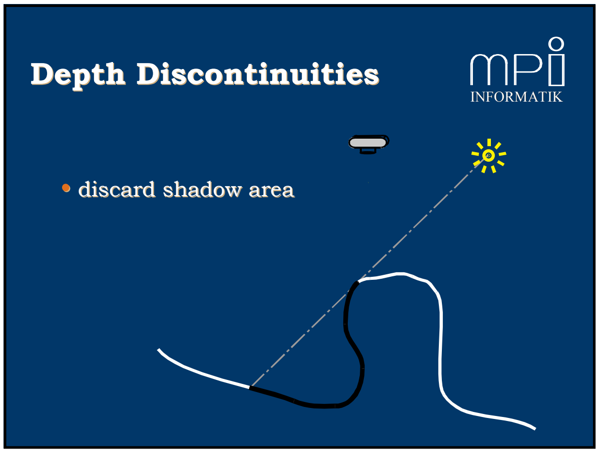
- for each point on the surface:

  find all images where the point is visible and lit
  take sample at corresponding pixel position



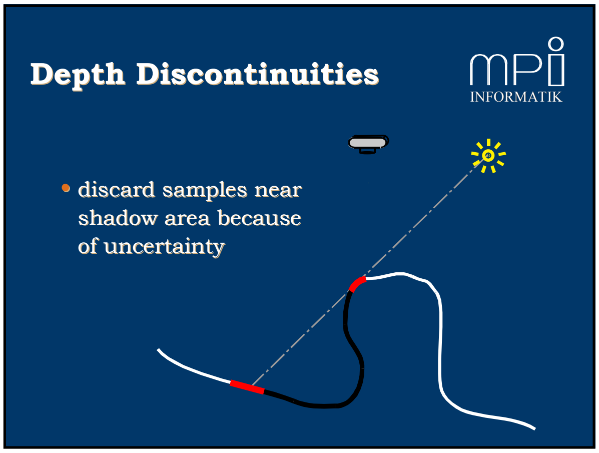
Depth Discontinuities

- depth discontinuities and shadow boundaries need special treatment



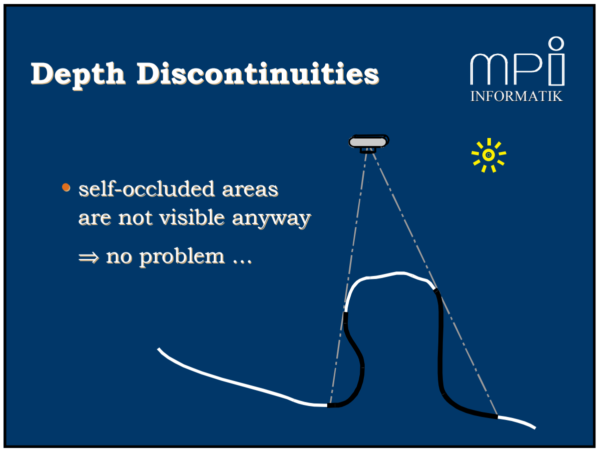
Depth Discontinuities

- discard shadow area



Depth Discontinuities

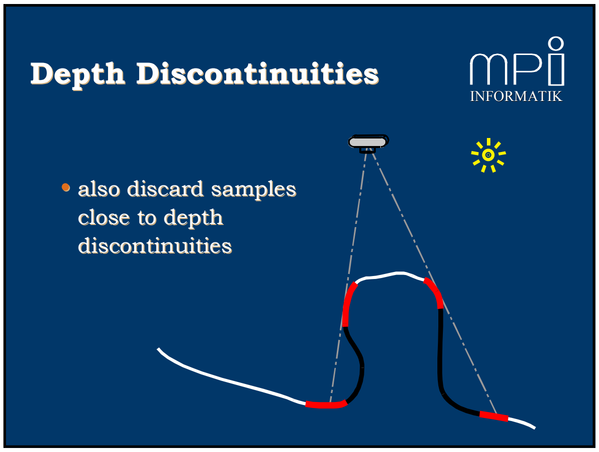
- discard samples near shadow area because of uncertainty



Depth Discontinuities

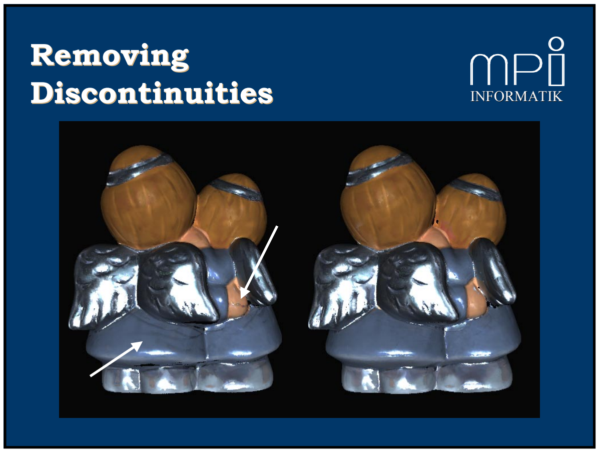
- self-occluded areas are not visible anyway

  ⇒ no problem …

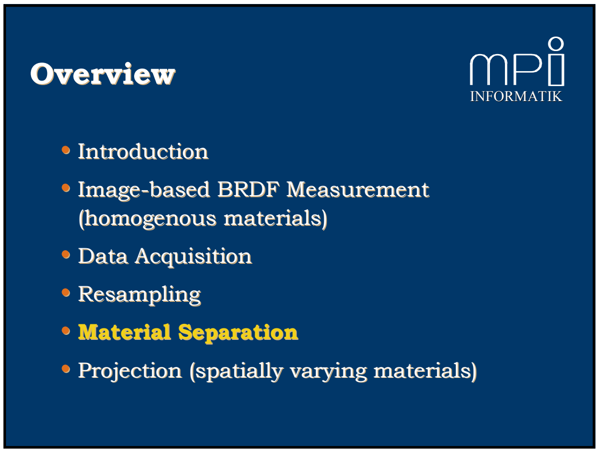


Depth Discontinuities

- also discard samples close to depth discontinuities



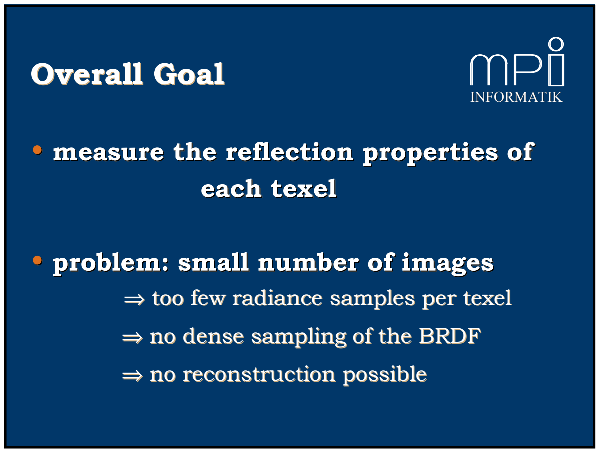
Removing Discontinuities



Overview

MPI
INFORMATIK

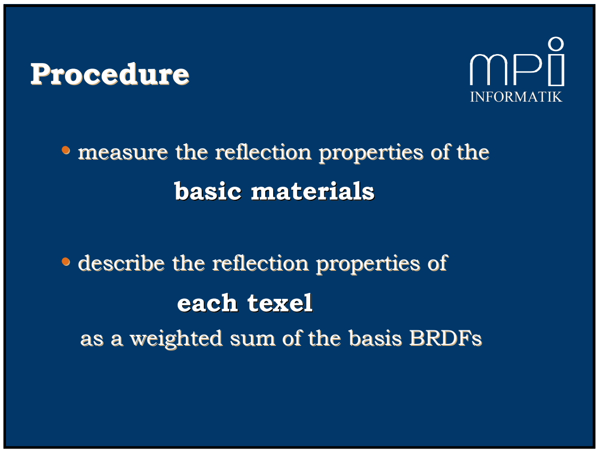
- Introduction
- Image-based BRDF Measurement (homogenous materials)
- Data Acquisition
- Resampling
- **Material Separation**
- Projection (spatially varying materials)



Overall Goal

MPI
INFORMATIK

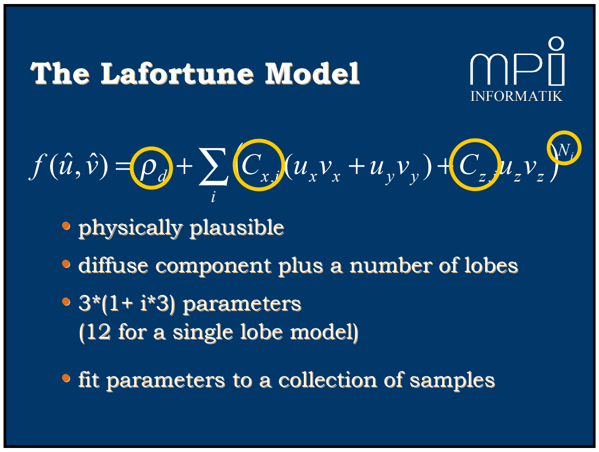
- **measure the reflection properties of each texel**
- **problem: small number of images**
  - ⇒ too few radiance samples per texel
  - ⇒ no dense sampling of the BRDF
  - ⇒ no reconstruction possible



Procedure

MPI
INFORMATIK

- measure the reflection properties of the **basic materials**
- describe the reflection properties of **each texel** as a weighted sum of the basis BRDFs

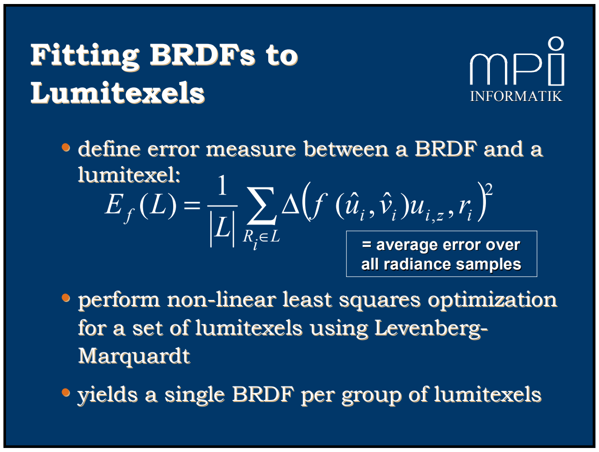


The Lafortune Model

MPI
INFORMATIK

$$f(\hat{u},\hat{v}) = \rho_d + \sum_i \left(C_{x,i}(u_x v_x + u_y v_y) + C_{z,i} u_z v_z \right)^{N_i}$$

- physically plausible
- diffuse component plus a number of lobes
- 3*(1+ i*3) parameters
  (12 for a single lobe model)
- fit parameters to a collection of samples



Fitting BRDFs to Lumitexels

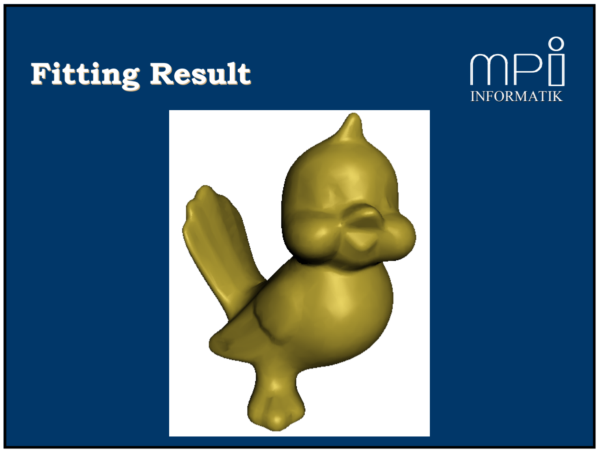
MPI
INFORMATIK

- define error measure between a BRDF and a lumitexel:

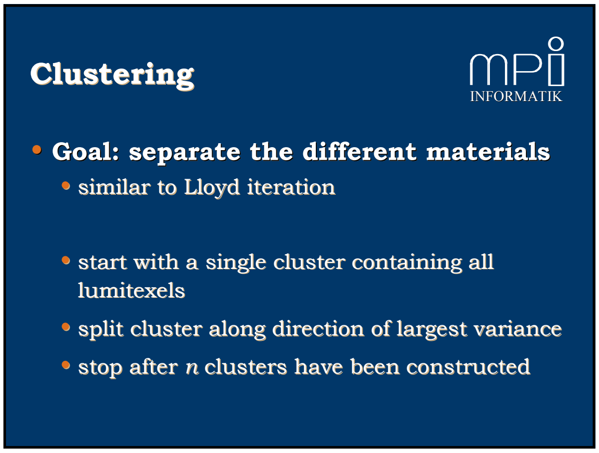
$$E_f(L) = \frac{1}{|L|} \sum_{R_i \in L} \Delta\left(f\,(\hat{u}_i,\hat{v}_i) u_{i,z}, r_i \right)^2$$

= average error over all radiance samples

- perform non-linear least squares optimization for a set of lumitexels using Levenberg-Marquardt
- yields a single BRDF per group of lumitexels

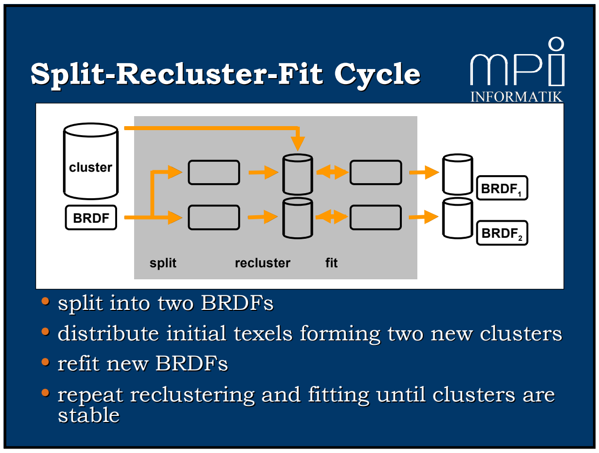


Fitting Result



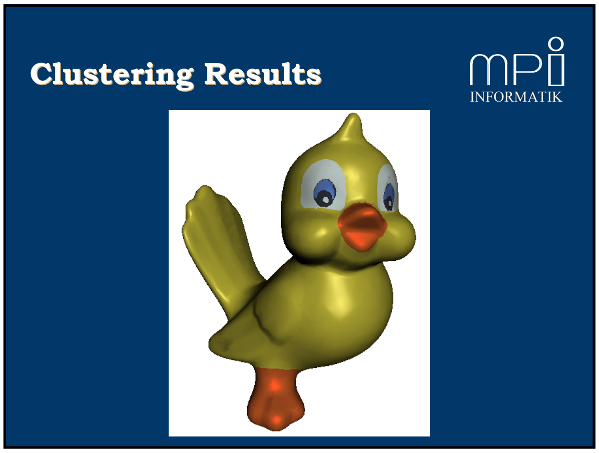
Clustering

- **Goal: separate the different materials**
  - similar to Lloyd iteration
  - start with a single cluster containing all lumitexels
  - split cluster along direction of largest variance
  - stop after *n* clusters have been constructed

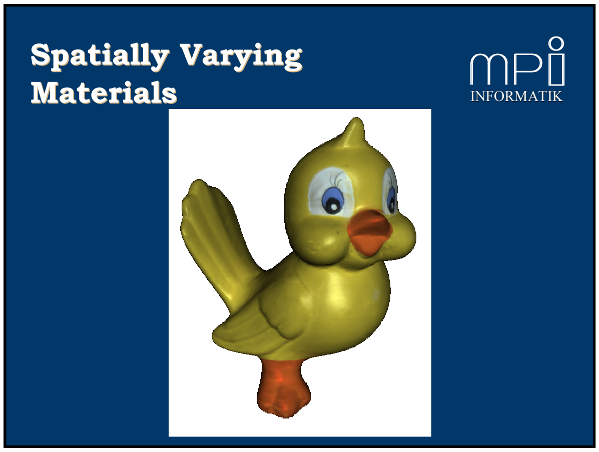


Split-Recluster-Fit Cycle

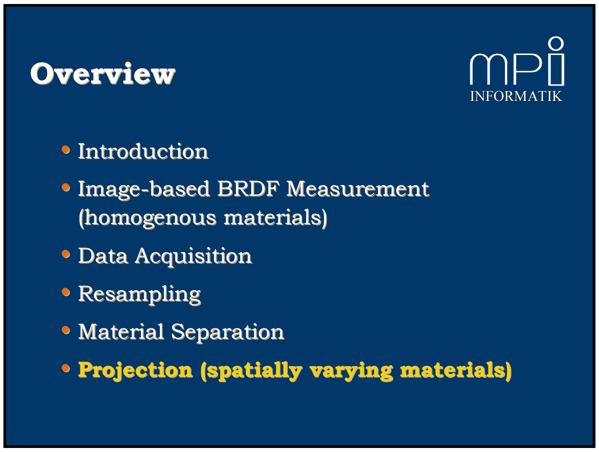
- split into two BRDFs
- distribute initial texels forming two new clusters
- refit new BRDFs
- repeat reclustering and fitting until clusters are stable



Clustering Results

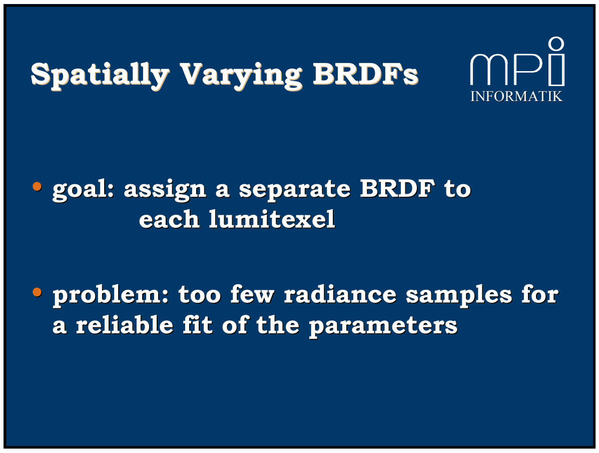


Spatially Varying Materials



Overview

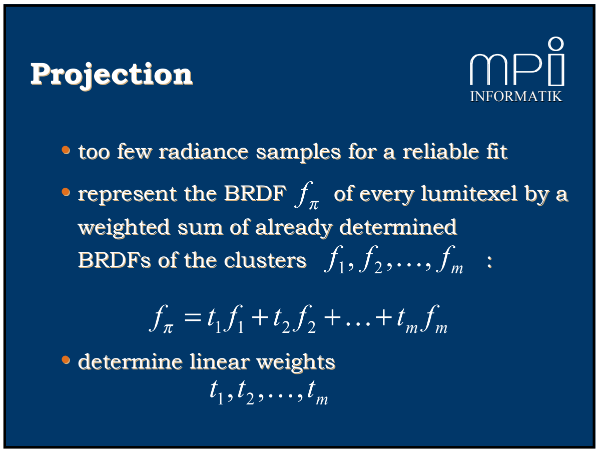
- Introduction
- Image-based BRDF Measurement (homogenous materials)
- Data Acquisition
- Resampling
- Material Separation
- **Projection (spatially varying materials)**



Spatially Varying BRDFs

MPI INFORMATIK

- **goal: assign a separate BRDF to each lumitexel**
- **problem: too few radiance samples for a reliable fit of the parameters**



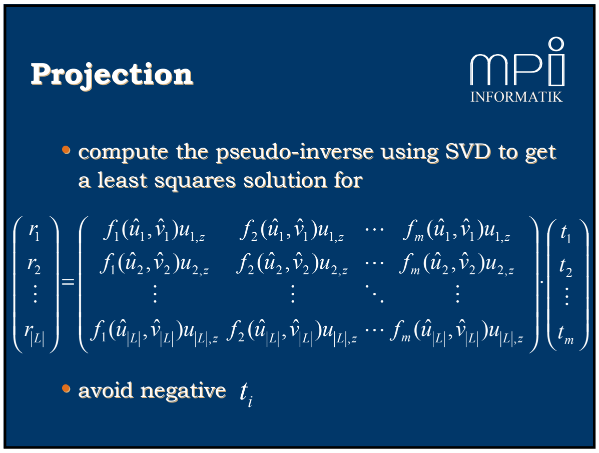
Projection

MPI INFORMATIK

- too few radiance samples for a reliable fit
- represent the BRDF f_π of every lumitexel by a weighted sum of already determined BRDFs of the clusters $f_1, f_2, \ldots, f_m$:

$$f_\pi = t_1 f_1 + t_2 f_2 + \ldots + t_m f_m$$

- determine linear weights $t_1, t_2, \ldots, t_m$



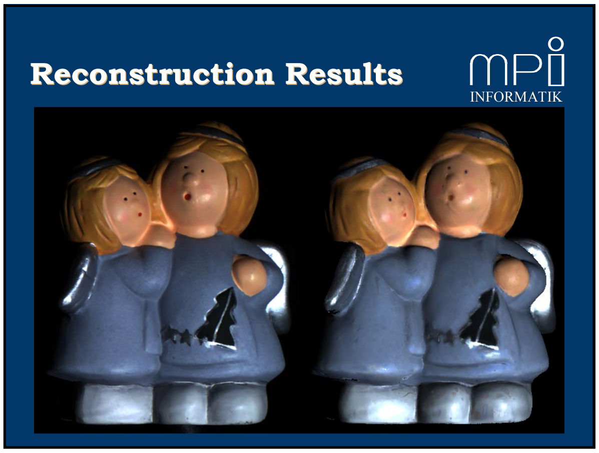
Projection

MPI INFORMATIK

- compute the pseudo-inverse using SVD to get a least squares solution for

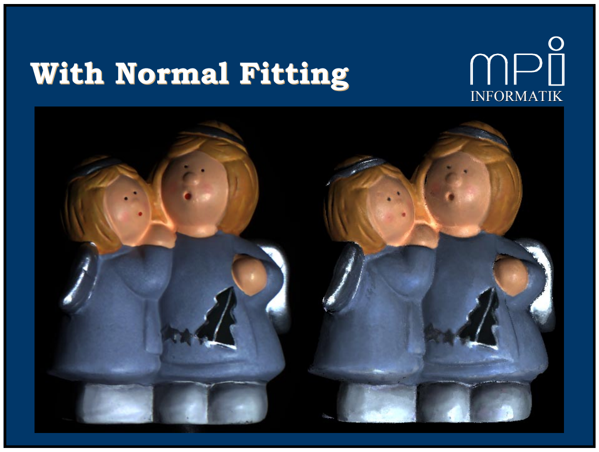
$$\begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{|L|} \end{pmatrix} = \begin{pmatrix} f_1(\hat{u}_1, \hat{v}_1) u_{1,z} & f_2(\hat{u}_1, \hat{v}_1) u_{1,z} & \cdots & f_m(\hat{u}_1, \hat{v}_1) u_{1,z} \\ f_1(\hat{u}_2, \hat{v}_2) u_{2,z} & f_2(\hat{u}_2, \hat{v}_2) u_{2,z} & \cdots & f_m(\hat{u}_2, \hat{v}_2) u_{2,z} \\ \vdots & \vdots & \ddots & \vdots \\ f_1(\hat{u}_{|L|}, \hat{v}_{|L|}) u_{|L|,z} & f_2(\hat{u}_{|L|}, \hat{v}_{|L|}) u_{|L|,z} & \cdots & f_m(\hat{u}_{|L|}, \hat{v}_{|L|}) u_{|L|,z} \end{pmatrix} \cdot \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_m \end{pmatrix}$$

- avoid negative t_i



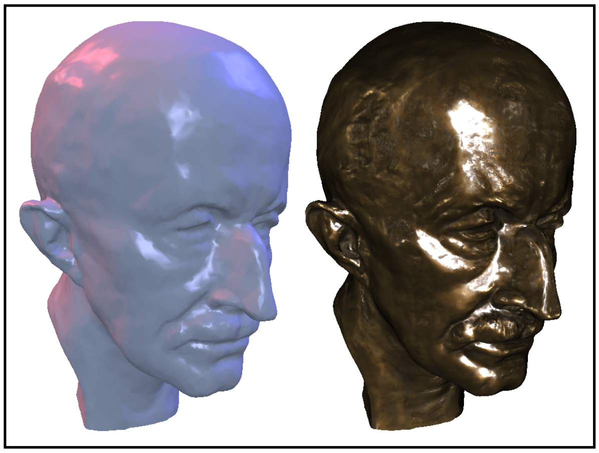
Reconstruction Results

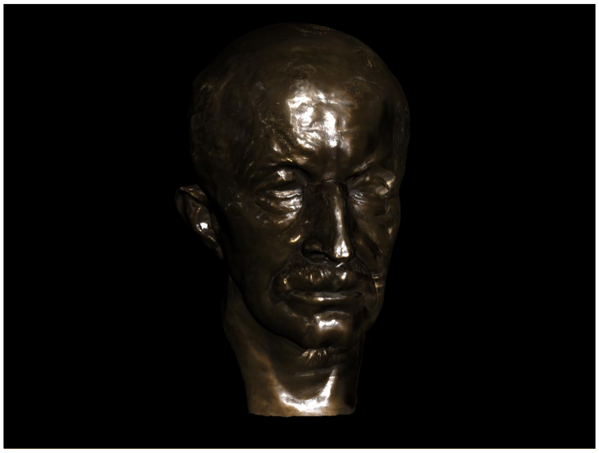
MPI INFORMATIK

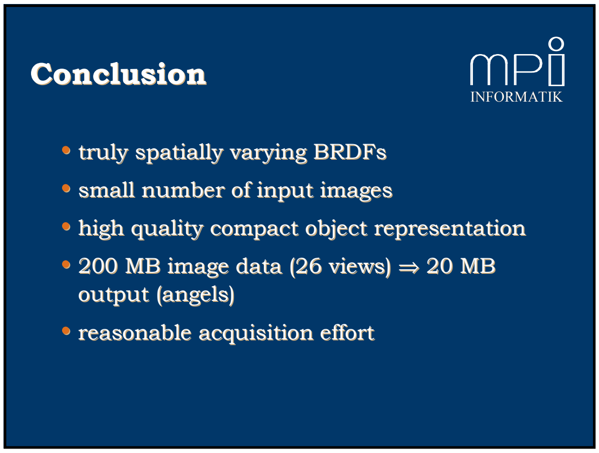


With Normal Fitting

MPI INFORMATIK

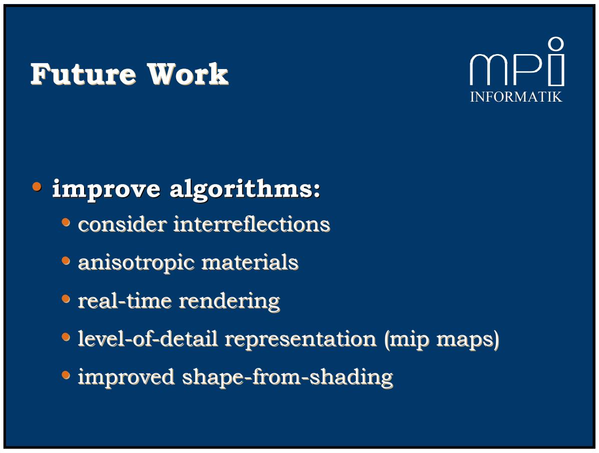






Conclusion

- truly spatially varying BRDFs
- small number of input images
- high quality compact object representation
- 200 MB image data (26 views) ⇒ 20 MB output (angels)
- reasonable acquisition effort

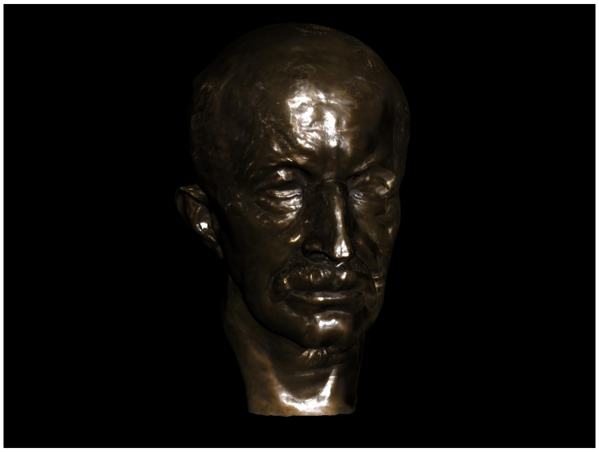


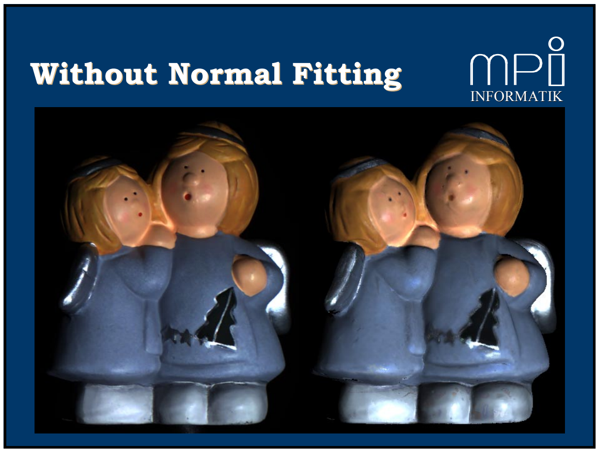
Future Work

- **improve algorithms:**
  - consider interreflections
  - anisotropic materials
  - real-time rendering
  - level-of-detail representation (mip maps)
  - improved shape-from-shading

Thank You

- **Questions?**
- **www.mpi-sb.mpg.de**





Without Normal Fitting

Image-Based Reconstruction of Spatially Varying Materials

Hendrik P. A. Lensch[1] Jan Kautz[1] Michael Goesele[1]
Wolfgang Heidrich[2] Hans-Peter Seidel[1]

[1] Max-Planck-Institut für Informatik
Saarbrücken, Germany

[2] The University of British Columbia
Vancouver, Canada

Abstract. The measurement of accurate material properties is an important step towards photorealistic rendering. Many real-world objects are composed of a number of materials that often show subtle changes even within a single material. Thus, for photorealistic rendering both the general surface properties as well as the spatially varying effects of the object are needed.

We present an image-based measuring method that robustly detects the different materials of real objects and fits an average bidirectional reflectance distribution function (BRDF) to each of them. In order to model the local changes as well, we project the measured data for each surface point into a basis formed by the recovered BRDFs leading to a truly spatially varying BRDF representation.

A high quality model of a real object can be generated with relatively few input data. The generated model allows for rendering under arbitrary viewing and lighting conditions and realistically reproduces the appearance of the original object.

1 Introduction

The use of realistic models for all components of image synthesis is a fundamental prerequisite for photorealistic rendering. This includes models for the geometry, light sources, and cameras, as well as materials. As more and more visual complexity is demanded, it is more and more often infeasible to generate these models manually. Automatic and semi-automatic methods for model acquisition are therefore becoming increasingly important.

In this paper we concentrate on the acquisition of realistic materials. In particular, we describe an acquisition process for spatially varying BRDFs that is efficient, reliable, and requires little manual intervention. Other methods described in the literature (see Section 2 for an overview) are either focusing on homogeneous materials, or make assumptions on the type of material to be measured (e.g. human faces). In our work, we measure spatially varying BRDFs without making any additional assumptions. In particular, our contributions are

- a robust and efficient BRDF fitting process that clusters the acquired samples into groups of similar materials and fits a Lafortune model [11] to each group,
- a method that projects every sample texel into a basis of BRDFs obtained from the clustering procedure. This projection accurately represents the material at that point and results in a compact representation of a truly spatially varying BRDF.

We require only a relatively small number of high-dynamic range photographs (about 20-25 images for one object), thereby speeding up the acquisition phase.

As a result of the fitting, clustering, and projection process, we obtain a compact representation of spatially varying materials that is well suited for rendering purposes (see Figure 5 for an example). The method works both for objects consisting of a mixture of distinct materials (e.g. paint and silver, see Figure 7), or for smooth transitions between material properties.

In the following we first review some of the previous work in this area, before we discuss the details of our own method. We start by describing the acquisition of the measurement data (Section 3), explain the resampling of this data into our data structures (Section 4), the BRDF fitting and material clustering steps (Sections 5 and 6), and finally present a method for projecting the materials into a basis of BRDFs (Section 7). Section 8 briefly describes our rendering method. In Section 9 we present our results and then we conclude in Section 10.

2 Related Work

The representation of real-world materials has recently received a lot of attention in the computer graphics community. The approaches can be grouped into three different categories: light field and image database methods with static illumination, dense sampling of the light and viewing directions to generate a tabular representation of the BRDF, and finally the fitting of reflection models, often based on a sparser set of samples. This last approach is the one we take and extend to spatially varying BRDFs.

In the first category, there has been a number of approaches ranging from a relatively sparse set of images with a geometric model [4] over the Lumigraph [7] with more images and a coarser model to the light field [13] with no geometry and a dense image database. Recently surface light fields [27, 18] have become popular, which feature both a dense sampling of the directional information and a detailed geometry. In contrast to these approaches, bidirectional texture functions [1] also work for changes in the lighting conditions, although at very high storage costs. In our work we use an algorithm similar to the function quantization approach proposed by Wood et al. [27] to resample the image data into a compact representation.

The traditional approach for dense sampling of reflectance properties is to use specialized devices (gonioreflectometers), that position both a light source and a sensor relative to the material. These devices can only obtain one sample for each pair of light and sensor position and are therefore relatively slow.

More recently, image-based approaches have been proposed. These methods are able to acquire a large number of samples at once. For example, Ward Larson [25] uses a hemispherical mirror to sample the exitant hemisphere of light with a single image. Instead of using curved mirrors, it is also possible to use curved geometry to obtain a large number of samples with a single image. This approach is taken by Lu et al [15], who assume a cylindrical surface, and Marschner et al. [17] who obtain the geometry using a range scanner. Our method is similar in spirit to the method of Marschner et al., but we are also dealing with spatially varying BRDFs and we are fitting a reflection model rather than using a tabular form in order to achieve a compact representation.

A number of researchers have also described the fitting of reflection models to the acquired sample data [2, 11, 22, 25, 28]. Of these methods, the ones by Ward Larson [25] and Lafortune et al. [11] do not consider spatial variations. Sato et al. [22] fit a Torrance-Sparrow model [24] to the data, and consider high-frequency variations for the diffuse part but only per-triangle variations for the specular part. This is also the case for the work by Yu et al. [28], which also takes indirect illumination into account. In our work, we perform the measurements in a darkened, black room, so that there is no indirect light coming from the outside of the object. Indirect light within the object is assumed to be negligible, which excludes the use of objects with extreme concavities.

Debevec et al. [2] describe a method for acquiring the reflectance field of human faces. In one part of their work they fit a specialized reflection model for human skin to the measured data (consisting of about 200 images). Both specular and diffuse parameters of the reflection model can vary rapidly across the surface, but other parameters like the de-saturation of the diffuse component at grazing angles are constant and only apply to human skin. In our work we try to avoid making assumptions on the kind of material we are measuring.

Several different representation have been used for fitting BRDF data. In addition to the models used for measured data (e.g. Koenderink et al. [10], Lafortune [11], Torrance-Sparrow [22, 28], Ward [25]), Westin et al. [26] have used spherical harmonics for projecting simulated BRDF data. In our work we use the Lafortune model because it is compact, well suited for optimization algorithms, and capable of representing interesting BRDF properties such as off-specular peaks and retro-reflection.

3 Acquisition

We obtain the 3D models with a structured light 3D scanner and a computer tomography scanner both generating dense triangle meshes. The triangle meshes are smoothed [5, 9], manually cleaned, and decimated.

All images are acquired in a measurement lab using a professional digital camera. An HMI metal halide bulb serves as point light source for the BRDF measurements. The interior of the photo studio is covered with dark and diffusely reflecting felt to reduce the influence of the environment on the measurements.

Several views of each object are captured with different camera and light source positions. For each view we acquire three sets of images: two images to recover the light source position, one image of the object's silhouette to register the 3D model with the images. We then acquire a high dynamic range image [3] of the object lit by the point light source by taking a series of photographs with varying exposure time.

In addition, a series of calibration images of a checkerboard pattern is taken whenever the lens settings are changed. The calibration method proposed by Zhang [29] is used to recover the intrinsic camera parameters. Another high dynamic range image of a gray card with known camera and light position is taken in order to compute the brightness of the light source.

To register the images with the 3D model we use a silhouette-based method [12] that yields the camera position relative to the object. The light source position is triangulated based on the reflections in a number of mirroring steel balls. The details of that approach will be described elsewhere.

4 Resampling of Radiance Values

After acquisition of the geometric model, high-dynamic range image recovery, and registration, it is necessary to merge the acquired data for further processing. For each point on the model's surface we collect all available information using two data structures.

The first one is a so called *lumitexel* denoted by $\mathcal{L}$, which is generated for every visible surface point. Each lumitexel stores the geometric and photometric data of one point, i.e. its position $\vec{x}$ and the normal $\hat{n}$ in world coordinates[1]. Linked to the lumitexel

[1] hats denote unit vectors and arrows denote vectors of arbitrary length.

is a list of radiance samples $\mathcal{R}_i$, each representing the outgoing radiance r of the surface point captured by one image plus the direction of the light $\hat{u}$ and the viewing direction $\hat{v}$. $\hat{u}$ and $\hat{v}$ are both given in the local coordinate frame of the surface point spanned by $\hat{n}$ and a deterministically constructed tangent and bi-normal.

A lumitexel can be seen as a very sparsely sampled BRDF. We define the error between a given BRDF f_r and a lumitexel $\mathcal{L}$ as:

$$E_{f_r}(\mathcal{L}) = \frac{1}{|\mathcal{L}|} \sum_{\mathcal{R}_i \in \mathcal{L}} s \cdot I(f_r(\hat{u}_i, \hat{v}_i) u_{i,z}, r_i) + D(f_r(\hat{u}_i, \hat{v}_i) u_{i,z}, r_i), \tag{1}$$

where $|\mathcal{L}|$ stands for the number of radiance samples linked to the lumitexel, $I(r_1, r_2)$ is a function measuring the intensity difference, and $D(r_1, r_2)$ measures the color-difference. We introduce the weight s, to be able to compensate for noisy data (e.g. a slightly wrong normal resulting in a wrong highlight). We always set $s \leq 1$. Please note that since r represents radiance and not reflectance, the BRDF has to be multiplied by the cosine between the normal and the local light direction, which is u_z.

4.1 Assembling Lumitexels

Collecting all radiance samples for a lumitexel requires a resampling of the input images for the particular point on the surface. At first, one has to determine the set of surface points for which a lumitexel should be generated. In order to obtain the highest quality with respect to the input images, the sampling density of the surface points must match that of the images.

Every triangle of the 3D model is projected into each image using the previously determined camera parameters. The area of the projected triangle is measured in pixels and the triangle is assigned to the image I_{best} in which its projected area is largest. For every pixel within the triangle in I_{best} a lumitexel is generated.

The position $\vec{x}$ of the surface point for the lumitexel is given by the intersection of the ray from the camera through the pixel with the mesh (see Figure 1). The normal $\hat{n}$ is interpolated using the triangle's vertex normals.

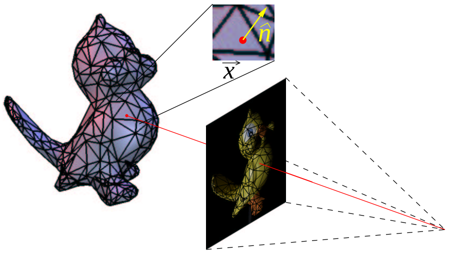


Fig. 1. *The correspondence between pixel position and point position $\vec{x}$ on the object is computed by tracing a ray through the image onto the object. At every $\vec{x}$ a local normal $\hat{n}$ can be computed from the triangle's vertex normals.*

A radiance sample $\mathcal{R}_j$ is now constructed for each image I_j in which $\vec{x}$ is visible from the camera position and the surface point is lit by the point light source. The vectors $\hat{u}_j$ and $\hat{v}_j$ can be directly calculated. The associated radiance is found by projecting $\vec{x}$ onto the image plane and retrieving the color c_j at that point using bilinear interpolation. Note, that for I_{best} no bilinear interpolation is necessary and c_{best} can be obtained without resampling since $\vec{x}$ exactly maps to the original pixel by construction. The radiance r_j of the radiance sample $\mathcal{R}_j$ is obtained by scaling c_j according to the brightness of the light source and the squared distance from the light source to $\vec{x}$.

5 BRDF Fitting

In this section we will first detail the Lafortune BRDF model [11] that we use to fit our given lumitexels. Then we will explain how this fit is performed using Levenberg-Marquardt optimization.

5.1 Lafortune Model

BRDFs are four-dimensional functions that depend on the local viewing and light direction. The dependence on wavelength is often neglected or simply three different BRDFs are used for the red, green, and blue channel. We use the latter approach.

Instead of representing a measured BRDF as a 4D table the measured samples are in our case approximated with a parameterized BRDF. This has two advantages. Firstly, the BRDF requires much less storage since only the parameters are stored and secondly, we only require a sparse set of samples that would not be sufficient to faithfully represent a complete tabular BRDF.

Many different BRDF models have been proposed (e.g. [24, 25]) with different strengths and weaknesses. Our method may be used together with any parameterized BRDF model. We have chosen the computationally simple but general and physically plausible Lafortune model [11] in its isotropic form:

$$f_r(\hat{u}, \hat{v}) = \rho_d + \sum_i [C_{x,i}(u_x v_x + u_y v_y) + C_{z,i} u_z v_z]^{N_i}, \tag{2}$$

This model uses only a handful of parameters. $\hat{u}$ and $\hat{v}$ are the local light and viewing directions, ρ_d is the diffuse component, N_i is the specular exponent, the ratio between $C_{x,i}$ and $C_{z,i}$ indicates the off-specularity of lobe i of the BRDF. The sign of $C_{x,i}$ makes the lobe i either retro-reflective (positive $C_{x,i}$) or forward-reflective (negative $C_{x,i}$). The albedo of the lobe i is given by the magnitude of the parameters $C_{x,i}$ and $C_{z,i}$. From now on we will denote the BRDF with $f_r(\vec{a}; \hat{u}, \hat{v})$, where $\vec{a}$ subsumes all the parameters of the model, i.e. ρ_d, $C_{x,i}$, $C_{z,i}$, and N_i. In the case of only one lobe $\vec{a}$ is 12-dimensional (4 parameters for each color channel).

5.2 Non-Linear Fitting

The Lafortune BRDF is non-linear in its parameters, which means that we have to use a non-linear optimization method to fit the parameters to the given data. As in the original work by Lafortune et al. [11], we use the Levenberg-Marquardt optimization [20] to determine the parameters of the Lafortune model from our measured data. This method has proven to be well-suited for fitting non-linear BRDFs.

Instead of BRDF samples we use radiance samples as our input data, which means we are not directly fitting the BRDF $f_r(\vec{a}; \hat{u}, \hat{v})$ but the radiance values $f_r(\vec{a}; \hat{u}, \hat{v}) u_z$ to the radiance samples $\mathcal{R}_i$ in order to avoid the numerically problematic division by u_z.

We also ensure that the fitting process works well and does not get stuck in undesired local minima by initializing the fitting routine with parameters that correspond to an average BRDF.

The Levenberg-Marquardt optimization outputs not only the best-fit parameter vector $\vec{a}$, but also a covariance matrix of the parameters, which provides a rough idea of the parameters that could not be fit well. This information is used in our splitting and clustering algorithm, as explained in the next section.

6 Clustering

In this section we will explain how we cluster the given lumitexels so that each cluster C_i corresponds to one material of the object. Given a set of BRDFs $\{f_i\}$, each cluster C_i consists of a list of all the lumitexels $\mathcal{L}_i$ for which f_i provides the best approximation. Determining these clusters is a problem closely related to vector quantization [6] and k-means clustering [14, 16], both of which work in affine spaces. Unfortunately, we do not have an affine space when clustering BRDF samples, and we are therefore employing a modified Lloyd [14] iteration method.

The general idea is to first fit a BRDF f_r to an initial cluster containing all the data. Then we generate two new BRDF models f_1 and f_2 using the covariance matrix from the fit (explained in more detail below) representing two new clusters. The lumitexels $\mathcal{L}_i$ from the original cluster are then distributed according to the distance $E_{f_1}(\mathcal{L}_i)$ and $E_{f_2}(\mathcal{L}_i)$ into the new clusters. We then recursively choose another cluster, split it, and redistribute the lumitexels and so on. This is repeated until the desired number of materials is reached, as detailed in Section 6.4.

6.1 Lumitexel Selection

The fitting procedure described in Section 5 performs a relatively large number of operations per radiance sample. Thus, it is expensive to fit a BRDF using all lumitexels (and all radiance samples contained in the lumitexels) generated by the assembling procedure. Instead, it is sufficient to consider only a few thousand lumitexels at the beginning. Later on, we increase the number for an accurate fit.

A first, naive approach to choosing this subset for fitting selects every n-th lumitexel regardless of its reliability or possible contribution. However, as stated in [28] and [23], for a robust estimation of the specular part of a BRDF it is very important to include radiance samples within the specular lobe of the material. Unfortunately, these brightest pixels statistically also carry the largest error.

Following these ideas we select more lumitexels in areas where a highlight is likely to occur. These areas are determined by the surface normal, the light source position and a synthetic BRDF with a broad highlight.

6.2 Splitting

Fitting just a single BRDF to the initial cluster of course is not sufficient if the concerned object consists of more than one material. In order to decide which cluster to split, we compute the following error for all clusters C_j:

$$E(C_j) = \sum_{\mathcal{L}_i \in C_j} E_{f_r}(\mathcal{L}_i) \quad \forall C_j. \tag{3}$$

The cluster C_j with the largest error will be split into two new clusters each with a different BRDF. Further materials can be extracted by further splitting the clusters.

But how do we split a cluster? The BRDF fit to a cluster represents the average material of the lumitexels in that cluster. Fitting the BRDF using the Levenberg-Marquardt algorithm (see Section 5) will also provide us with the covariance matrix of the parameters. The eigenvector belonging to the largest eigenvalue of this matrix represents the direction in which the variance of the samples is highest, and is therefore a good choice for the direction in which the parameter space is to be split.

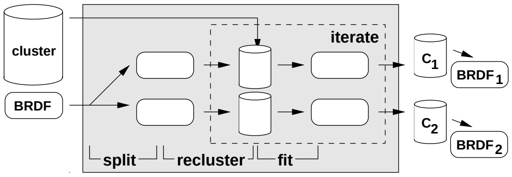


Fig. 2. *Split-recluster-fit process (SRF). The initial BRDF is split into two new BRDFs using the covariance matrix. The lumitexels from the initial cluster are distributed according to their distance to the BRDFs. Then we fit the BRDF again to each new cluster. We now iterate the reclustering and fitting until the resulting BRDFs and cluster have converged.*

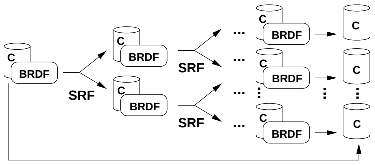


Fig. 3. *The complete splitting and reclustering algorithm including the global reclustering, which is similar to the recluster-fit iteration, only that all lumitexels are distributed among all clusters.*

Let $\vec{a}$ be the fit parameter vector of the BRDF $f(\vec{a};\hat{u},\hat{v})$ for cluster C. $\vec{e}$ denotes the eigenvector belonging to the largest eigenvalue λ of the corresponding covariance matrix. We then construct two new BRDFs:

$$f_1(\vec{a}+\tau\lambda\vec{e};\hat{u},\hat{v}) \quad \text{and} \quad f_2(\vec{a}-\tau\lambda\vec{e};\hat{u},\hat{v}), \tag{4}$$

where τ is a scaling factor to adapt λ to a moderate value. Two new clusters C_1 and C_2 are generated by distributing every lumitexel $\mathcal{L}_i$ of cluster C either to C_1 if $E_{f_1}(\mathcal{L}_i) < E_{f_2}(\mathcal{L}_i)$, or to C_2 otherwise. In the next step, f_1 and f_2 are fit to best approximate the lumitexels in the new clusters.

6.3 Reclustering

Because the parameters of the BRDF fit to a multi-material cluster are not necessarily the center of the parameters of the contained materials and due to improper scaling of λ and other reasons like noise, the performed split will not be optimal and the two new clusters may not be clearly separated, e.g. in the case of two distinct materials some lumitexels belonging to one material may still be assigned to the cluster of the other material.

A better separation can be achieved by iterating the procedure of distributing the lumitexels $\mathcal{L}_i$ based on $E_{f_1}(\mathcal{L}_i)$ and $E_{f_2}(\mathcal{L}_i)$, and then fitting the BRDFs again. The iteration stops when the number of lumitexels in the generated cluster does not change any more. In our experiments this reclustering operation leads to a clear separation of materials and is done after each split. The split-recluster-fit (SRF) process is visualized in Figure 2.

When more than two clusters have been generated by successive binary splits and a new material is clearly distinguished, it is helpful to clean the other clusters, which were not involved in the last split, from all lumitexels belonging to the newly discovered material. This can be done in a global reclustering step by redistributing all initial lumitexels $\mathcal{L}_i$ to the cluster C_j with

$$j = \underset{k}{\text{argmin}}\, E_{f_k}(\mathcal{L}_i). \tag{5}$$

And again, the BRDFs of all involved clusters have to be refit. This global reclustering is repeated several times to clearly separate the materials. We stop this iteration when the percentage of change is smaller than some ϵ, or a maximum number of iterations is reached. The complete splitting and reclustering algorithm is depicted in Figure 3.

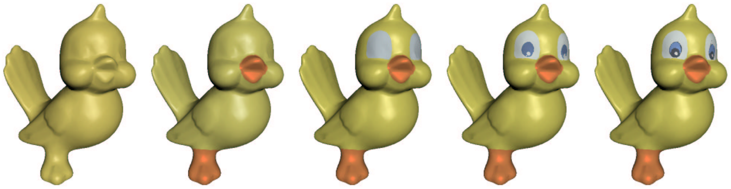


Fig. 4. *The clustering process at work. In every image a new cluster was created. The object was reshaded using only the single BRDFs fit to each cluster before the projection into a basis of multiple BRDFs.*

6.4 Termination of the Splitting Process

We still have to decide when to stop the splitting process. To do this we require the user to input the estimated number of materials $|M|$. We stop the splitting and clustering process after $2|M| - 1$ clusters have been created. We use this additional number of clusters to compensate for the often noisy and not absolutely accurate radiance samples (e.g. slightly wrong normals, noise in the images, misregistration, etc.).

This means that we do not have a one to one mapping between actual materials and clusters. This is not crucial since the projection, which we will present in the next section, uses a weighted sum of several BRDFs to accurately represent every lumitexel.

7 Projection

As can be seen in Figure 4 the representation of an object by a collection of only a few clusters and BRDFs make the virtual object look flat because real surface exhibit changes in the reflective properties even within a single material. These changes cannot be represented by a single BRDF per cluster since all lumitexels within the cluster would be assigned the same BRDF parameters.

To obtain truly spatially varying BRDFs we must find a specific BRDF for each lumitexel. But the sparse input data does not allow to fit a reliable or even meaningful BRDF to a single lumitexel because each lumitexel consists of only a few radiance samples. In addition, you would need to acquire a highlight in every lumitexel to reliably determine the specular part, as already explained in Section 6.1.

The solution is to project each lumitexel into a basis of BRDFs (see Section 7.1). The BRDF $f_{\pi i}$ of a lumitexel $\mathcal{L}_i$ is represented by the linear combination of m BRDFs $f_1, f_2, \ldots, f_m$:

$$f_{\pi i} = t_1 f_1 + t_2 f_2 + \ldots + t_m f_m, \tag{6}$$

with $t_1, t_2, \ldots, t_m$ being positive scalar weights. This forces the space of solutions (i.e. the possible BRDFs for a pixel) to be plausible since the basis BRDFs are already reliably fit to a large number of radiance samples.

Given the BRDFs, the weights have to be determined for each lumitexel. Let $r_{j=1\ldots|\mathcal{L}_i|}$ be the radiance values of the lumitexel $\mathcal{L}_i$. The weights are found by a least square optimization of the following system of equations using singular-value decomposition:

$$\begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{|\mathcal{L}_i|} \end{pmatrix} = \begin{pmatrix} \tilde{f}_1(\hat{u}_1, \hat{v}_1) & \tilde{f}_2(\hat{u}_1, \hat{v}_1) & \cdots & \tilde{f}_m(\hat{u}_1, \hat{v}_1) \\ \tilde{f}_1(\hat{u}_2, \hat{v}_2) & \tilde{f}_2(\hat{u}_2, \hat{v}_2) & \cdots & \tilde{f}_m(\hat{u}_2, \hat{v}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{f}_1(\hat{u}_{|\mathcal{L}_i|}, \hat{v}_{|\mathcal{L}_i|}) & \tilde{f}_2(\hat{u}_{|\mathcal{L}_i|}, \hat{v}_{|\mathcal{L}_i|}) & \cdots & \tilde{f}_m(\hat{u}_{|\mathcal{L}_i|}, \hat{v}_{|\mathcal{L}_i|}) \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_m \end{pmatrix}, \tag{7}$$

with $\tilde{f}(\hat{u},\hat{v}) := f(\hat{u},\hat{v})u_z$. Compared to the non-linear fitting of BRDF model parameters (see Section 5.2), we now have a linear problem to solve with a smaller degree of freedom and even more constraints. Above equation shows only the system for one color channel, whereas the weights t_i have to be the same for all channels. In contrast to this, BRDF parameters would require a distinct set of parameters per channel.

The least square solution may contain negative values for some t_k. But negative weights may result in an oscillating BRDF that represents only the given radiance sample accurately but will produce unpredictable values for other viewing and light directions, we therefore set t_k to zero and compute another least square solution for the remaining t's, until all t's are positive. This could also be seen as a constrained minimization problem.

7.1 Basis BRDFs

The next question is how to determine the set of basis BRDFs. Since the changes of the surface properties within one material tend to be small, a distinct set of basis BRDFs is assigned to each cluster. Therefore, it is sufficient to store just the scalar weights per lumitexel instead of the full set of BRDF parameters.

Finding the optimal set of BRDFs $f_1, f_2, \ldots, f_m$, that minimizes the error

$$E_\pi(C) = \frac{1}{|C|} \sum_{\mathcal{L}_i \in C} E_{f_{\pi i}}(\mathcal{L}_i) \tag{8}$$

for a cluster C, where $f_{\pi i}$ denotes the least square projection of the lumitexel $\mathcal{L}_i$ as defined in Equation 6, is a problem of principal function analysis (PFA) (see [27]). Principal function analysis is closely related to principal component analysis (PCA) with the important difference that functions f_m are optimized instead of vectors. Unfortunately, the PFA does not reduce to a simple eigenvalue problem as PCA does. To minimize $E_\pi(C)$, we again perform a least square optimization using the Levenberg-Marquardt method, this time fitting m BRDFs simultaneously. Within each iteration we recompute the projection $f_{\pi i}$ of lumitexel $\mathcal{L}_i$ into the currently estimated basis.

As for every optimization problem the initial parameters (BRDFs) are quite important. For a given cluster C, we use the following BRDFs as a basis:

- f_C, the BRDF fit to the cluster C,
- the BRDFs of spatially neighboring clusters to match lumitexels at cluster boundaries,
- the BRDFs of similar clusters with respect to the material,
- and two BRDFs based on f_C, one with slightly increased and one with decreased diffuse component ρ_d and exponent N.

In our experiments it turned out that this initial basis together with the projection already produces very good results with small errors. In most cases the PFA computed almost negligible changes to the initial BRDFs. This is to be expected because the initially chosen basis constructed through splitting and clustering already approximates the material properties quite well.

8 Rendering

As explained in Section 4.1 we know the position of every lumitexel, as well as the triangle it belongs to and the 2D coordinates within that triangle.

model	T	V	L	R	C	B	1-RMS	C-RMS	P-RMS	F-RMS
angels	47000	27	1606223	7.6	9	6	.2953	.1163	.1113	.1111
bird	14000	25	1917043	6.3	5	4	.1513	.0627	.0387	.0387
bust	50000	16	3627404	4.2	3	4	.1025	.0839	.0583	.0581

Table 1. *This table lists the number of triangles (T) of each model, the number of views (V) we used to reconstruct the spatially varying BRDFs, the number of acquired lumitexels (L) and the average number of radiance samples (R) per lumitexel, the number of partitioned material clusters (C), the number of basis BRDFs (B) per cluster, the RMS error for a single average BRDF (1-RMS), the RMS error when using per-cluster BRDFs, the RMS error after projecting every lumitexel into the basis of BRDFs, and finally the RMS error after doing a PFA on the basis BRDFs and projecting every lumitexel into the new basis.*

This information can then be used to generate an index texture for the full object. For every texel, that texture contains an index to the cluster it belongs to. Then we generate a weight texture map for every cluster that stores the weights resulting from the projection into the basis BRDFs. The parameters for the basis BRDFs of every cluster are stored in a small table.

Raytracing such an object is very simple, since for every point on the object that is raytraced we can simply look up the cluster the texel belongs to. Then we evaluate the basis BRDFs for the local light and viewing direction and compute the weighted sum using the weight texture map. So rendering basically reduces to evaluating a few BRDFs per pixel. Another big advantage of this representation is that mip-mapping can easily be used. Since the weighted sum is just a linear operation, the weights of neighboring texels can simply be averaged to generate the next coarser mip-map level.

If the original images are of high resolution and hence the object is sampled very densely, point sample rendering using forward projection is a viable alternative. It completely avoids the generation of texture maps and the resulting data can be used with almost no further processing. This method is used to display our results.

9 Results

We applied our algorithm to three different objects consisting of different materials with varying reflection properties in both the diffuse and the specular part. The model of the angels was generated by extracting an isosurface of a computer tomography scan. The geometry of all other models was captured using a structured light 3D scanner. Some statistics about the meshes and the number of acquired views are listed in Table 1. Acquisition of 20 views (each needing about 15 photographs) takes approx. 2.5h. The high dynamic range conversion and the registration with the 3D model takes about 5h but is a completely automated task. The clustering and the final projection takes about 1.5h.

In Figure 4 you can see how five successive split operations partition the lumitexels of the bird into its five materials. The splits were performed as described in Section 6. Only the per-cluster BRDFs determined by the clustering process are used for shading, making the object look rather flat. After performing the projection step every lumitexel is represented in a basis of four BRDFs, now resulting in a much more detailed and realistic appearance, see Figure 6.

The bust in Figure 5 shows another reconstructed object with very different reflection properties. The bronze look is very well captured.

In Figure 7 you can see a comparison between an object rendered with an acquired BRDF (using the projection method) and a photograph of the object. You can see that they are very similar, but differences can be seen in highlights and in places where not enough radiance samples were captured. Capturing more samples will increase the quality. The difference in the hair region is due to missing detail in the triangle mesh.

Another difference is due to the fact that the diffuse color of one lumitexel may not be represented in any of the constructed clusters because the number of lumitexels belonging to the same material can be so small that they nearly vanish in the mass of lumitexels of the cluster they are currently assigned to. This effect can for example be observed at the mouth of the larger angel which in reality exhibits a much more saturated red, see Figure 7.

In Table 1 we list RMS errors computed between all the radiance samples of a model and the reconstructed BRDFs. You can see that the error considerably decreases when going from one average BRDF to per-cluster BRDFs and then to per-pixel BRDFs (using projection). As already mentioned the PFA only slightly changes the RMS error.

Generally it can be said that for all the models only a few clusters were needed to accurately represent all the materials since the projection takes care of material changes. In our experiments even Lafortune BRDFs consisting of a single lobe were sufficient to form good bases for the clustering and projection.

The projection method also compensates for imprecise normals, and hence no refitting of the normals is needed. Using exactly reconstructed normals for example by applying a shape-from-shading approach such as the one by Rushmeier et al. [21] may yield even better results.

Due to the lack of a test object that had a single base color but varying specularity, we experimented with artificially generated data. The tests proved that our clustering algorithm is also able to clearly distinguish materials that have the same color but different specularity, even when noise was introduced in the data.

10 Conclusions and Future Work

We have presented an algorithm and demonstrated a system for reconstructing a high-quality spatially varying BRDF from complex solid objects using only a small number of images. This allows for accurately shaded, photorealistic rendering of these objects from new viewpoints and under arbitrary lighting conditions.

The output of our algorithm also allows to modify the object's geometry while preserving material properties, since the fitted BRDFs are represented on a per-texel basis and do not change with the geometry.

Both the number of input views required by our algorithm and the size of the output data (∼25MB) are very small compared to previous approaches for representing real-world objects, like surface light fields or reflection fields which needed up to 600 images [27].

We have demonstrated the quality and accuracy of our approach, by applying it to different objects. The resulting spatially varying BRDFs accurately represent the original materials.

Until now interreflections within the object are not considered, but it should be easy to remove the effects of interreflections by simulating secondary reflection using the results obtained by the presented algorithm, or e.g. using techniques from [19].

We also want to investigate the possibility to do hardware accelerated rendering with the spatially varying BRDFs. Since our data can be represented as texture maps and the Lafortune model is computationally fairly simple, this should be easily possible, e.g. using techniques from [8] or from [27].

References

1. K. Dana, B. van Ginneken, S. Nayar, and J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, January 1999.
2. P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the Reflectance Field of a Human Face. In *Proc. SIGGRAPH*, pages 145–156, July 2000. ISBN 1-58113-208-5.
3. P. Debevec and J. Malik. Recovering High Dynamic Range Radiance Maps from Photographs. In *Proc. SIGGRAPH*, pages 369–378, August 1997.
4. P. Debevec, C. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. SIGGRAPH*, pages 11–20, August 1996.
5. M. Garland and P. Heckbert. Surface Simplification Using Quadric Error Metrics. In *Proc. SIGGRAPH*, pages 209–216, August 1997.
6. A. Gersho and R. Gray. *Vector Quantization and Signal Compression.* Kluwer Acad. Publishers, 1992.
7. S. Gortler, R. Grzeszczuk, R. Szelinski, and M. Cohen. The Lumigraph. In *Proc. SIGGRAPH*, pages 43–54, August 1996.
8. J. Kautz and H.-P. Seidel. Towards Interactive Bump Mapping with Anisotropic Shift-Variant BRDFs. In *Eurographics/SIGGRAPH Hardware Workshop*, pages 51–58, August 2000.
9. L. Kobbelt. Discrete fairing. In *Proc. of the 7th IMA Conf. on the Mathematics of Surfaces*, pages 101–131, 1996.
10. J. Koenderink, A. van Doorn, and M. Stavridi. Bidirectional Reflection Distribution Function expressed in terms of surface scattering modes. In *Proc. 4th Europ. Conf. on Computer Vision*, pages 28–39, 1996.
11. E. Lafortune, S. Foo, K. Torrance, and D. Greenberg. Non-Linear Approximation of Reflectance Functions. In *Proc. SIGGRAPH*, pages 117–126, August 1997.
12. H. Lensch, W. Heidrich, and H.-P. Seidel. Automated Texture Registration and Stitching for Real World Models. In *Pacific Graphics '00*, pages 317–326, October 2000.
13. M. Levoy and P. Hanrahan. Light Field Rendering. In *Proc. SIGGRAPH*, pages 31–42, August 1996.
14. S. Lloyd. Least squares quantization in PCM. *IEEE Trans. on Information Theory*, IT-28:129–137, 1982.
15. R. Lu, J. Koenderink, and A. Kappers. Optical Properties (bidirectional reflectance distribution functions) of velvet. *Applied Optics*, 37(25):5974–5984, September 1998.
16. J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability*, volume 1, 1967.
17. S. Marschner, S. Westin, E. Lafortune, K. Torrance, and D. Greenberg. Image-based BRDF Measurement Including Human Skin. In *10th Eurographics Workshop on Rendering*, pages 131–144, June 1999.
18. G. Miller, S. Rubin, and D. Ponceleon. Lazy decompression of surface light fields for precomputed global illumination. In *9th Eurographics Workshop on Rendering*, pages 281–292, June 1998.
19. S. Nayar, K. Ikeuchi, and T. Kanade. Recovering Shape in the Presence of Interreflections. In *IEEE Int. Conf. on Robotics and Automation*, pages 1814–1819, 1991.
20. W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing (2nd ed.).* Cambridge University Press, 1992. ISBN 0-521-43108-5.
21. H. Rushmeier, G. Taubin, and A. Guéziec. Applying Shape from Lighting Variation to Bump Map Capture. In *8th Eurographics Workshop on Rendering Workshop*, pages 35–44, June 1997.
22. Y. Sato, M. Wheeler, and K. Ikeuchi. Object Shape and Reflectance Modeling from Observation. In *Proc. SIGGRAPH*, pages 379–388, August 1997.
23. H. Schirmacher, W. Heidrich, M. Rubick, D. Schiron, and H.-P. Seidel. Image-Based BRDF Reconstruction. In *Proc. of the 4th VMV Conference*, pages 285–292, November 1999.
24. K. Torrance and E. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of Optical Society of America*, 57(9), 1967.
25. G. Ward Larson. Measuring and Modeling Anisotropic Reflection. In *Proc. SIGGRAPH*, pages 265–272, July 1992.
26. S. Westin, J. Arvo, and K. Torrance. Predicting Reflectance Functions From Complex Surfaces. In *Proc. SIGGRAPH*, pages 255–264, July 1992.
27. D. Wood, D. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle. Surface Light Fields for 3D Photography. In *Proc. SIGGRAPH*, pages 287–296, July 2000.
28. Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse Global Illumination: Recovering Reflectance Models of Real Scenes From Photographs. In *Proc. SIGGRAPH*, pages 215–224, August 1999.
29. Z. Zhang. Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. In *Int. Conf. on Computer Vision*, pages 666–673, September 1999.

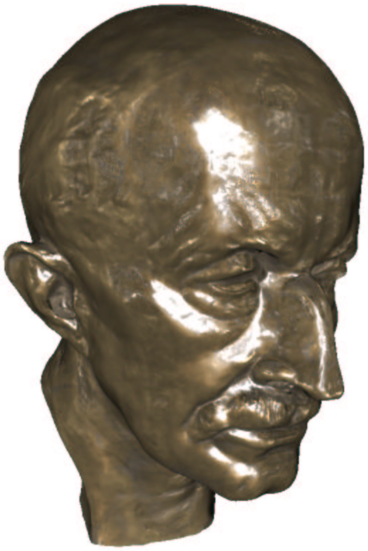


Fig. 5. *A bronze bust rendered with a spatially varying BRDF, which was acquired with our reconstruction method.*

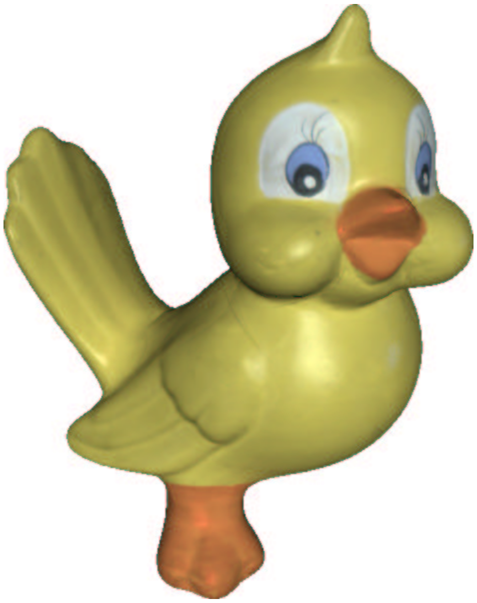


Fig. 6. *This image shows the bird with the spatially varying BRDF determined by projecting each lumitexel into a basis of BRDFs. Note the subtle changes of the materials making the object look realistic.*

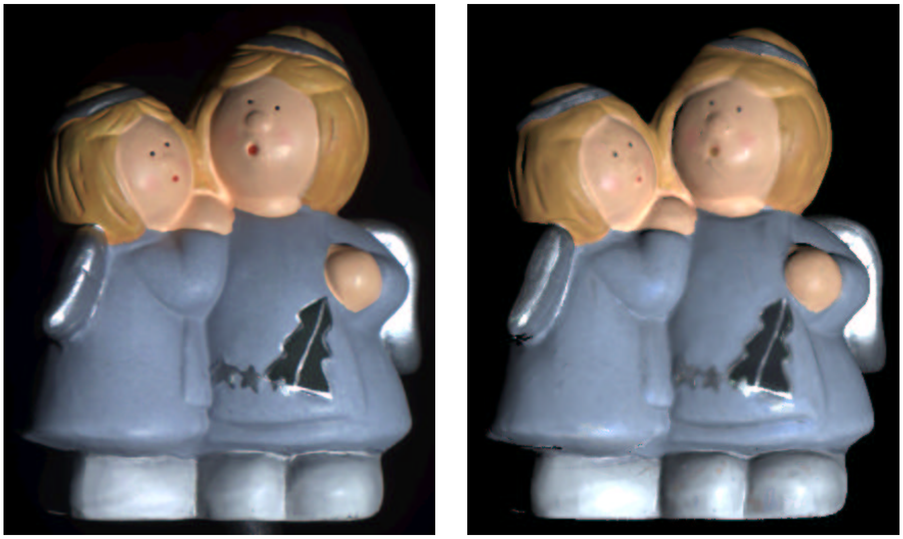


Fig. 7. *Left side: Photograph of model. Right side: Model with acquired BRDF rendered from the same view with similar lighting direction. The difference in the hair region is due to missing detail in the triangle mesh.*

Surface Detail Reconstruction from Integrated Shadow and Shading Constraints

(SIGGRAPH 2002 course notes)

Yizhou Yu **Johnny T. Chang**
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
{yyz,jtchang}@uiuc.edu

1 Shadow Constraints

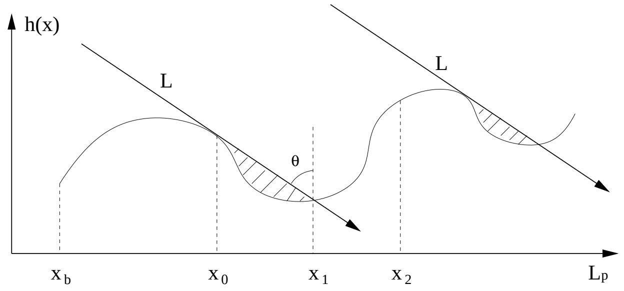


Figure 1: 2D schematic of shadowed and nonshadowed regions on a terrain-like surface. L is the parallel lighting direction. x_0 is an occluder, x_1 is on the shadow boundary caused by x_0, and x_2 is a non-shadowed point.

We consider recovering terrain-like height fields in this paper. For the convenience of discrete representation based on pixels, a height field is assumed to be a piecewise constant function with every pixel corresponding to a piece with constant height. Every piece of the height field is represented by the point corresponding to the center of the pixel. We also assume that the distance between the camera and the surface is large enough so that the orthographic projection model is accurate.

Let us first check what kind of constraints are available from images with shadows. Let $h(x)$ be a height field defined on a planar domain D with a finite area in the image plane and $\mathbf{L}$ be the lighting direction pointing downwards with a tilt angle $\theta(< 90^o)$. The normal orientation of this height field is denoted as $\mathbf{n}(x)$. The boundary curve of domain D is Ω. The projected vector of $\mathbf{L}$ in the domain D is $\mathbf{L}_p$. Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be two arbitrary 2D points in D. The line segment between them is denoted as a vector interval $[\mathbf{x_i}, \mathbf{x_j}]$ for convenience. Based on whether a point on the height field is in shadow or not under lighting direction $\mathbf{L}$, there are two different sets of constraints (Fig. 1).

- If any point on the line segment $[\mathbf{x_0}, \mathbf{x_1}]$ is in shadow, the points at $\mathbf{x}_0$ and $\mathbf{x}_1$ are the delimiting points of this shadow segment, and the point at $\mathbf{x}_0$ is the occluding point generating this shadow segment, we have the following *shadow* constraints.

$$h(\mathbf{x}) \le h(\mathbf{x_0}) - \frac{\|\mathbf{x} - \mathbf{x_0}\|_2}{\tan\theta}, \forall \mathbf{x} \in [\mathbf{x_0}, \mathbf{x_1}]; \tag{1}$$

$$h(\mathbf{x_1}) = h(\mathbf{x_0}) - \frac{\|\mathbf{x_1} - \mathbf{x_0}\|_2}{\tan\theta}; \tag{2}$$

$$\mathbf{L} \cdot n(x_0) = 0 \tag{3}$$

where the last equation means the lighting vector $\mathbf{L}$ falls inside the tangential plane at $\mathbf{x}_0$ if the original continuous height field is locally differentiable at $\mathbf{x}_0$.

- If the point at $\mathbf{x}_2$ is not in shadow, we have the following *antishadow* constraints.

$$h(\mathbf{x}) \leq h(\mathbf{x_2}) + \frac{\|\mathbf{x} - \mathbf{x_2}\|_2}{\tan\theta}, \forall \mathbf{x} \in [\mathbf{x_b}, \mathbf{x_2}] \tag{4}$$

where $\mathbf{x_b} \in \Omega$ and the line segment $[\mathbf{x_b}, \mathbf{x_2}]$ is in the same direction as $\mathbf{L}_p$.

2 Integrated Shadowing and Shading Constraints

Given a sparse set of images with known lighting directions, we would like to recover shape using both shadow and shading constraints. As we have seen, shadows impose explicit constraints over surface height values, but they are usually not sufficient if applied alone. On the other hand, shading information imposes constraints over normal orientation.

Since shape-from-shading is not the focus of this paper, we adopt the direct height from shading algorithm in [2] as the base for solving shading constraints. Since this technique computes a height field directly rather than through surface normals, it is relatively easy to incorporate shadow constraints and enforce surface upper/lower bounds from the previous section. The shape-from-shading problem is formulated to minimize the following cost function in [2]:

$$E_1 = \sum_{i,j}[\alpha(\rho R(p_{ij}, q_{ij}) - I(i,j))^2 + \lambda(u_{ij}^2 + v_{ij}^2)] \tag{5}$$

where ρ is the surface albedo, I is the observed image intensity, p_{ij}, q_{ij}, u_{ij}, v_{ij} are the symmetric first and second finite differences of the surface height field $\{h_{ij}\}$, α and λ are two constant coefficients, and R is the Lambertian reflectance model. The first term in Eq. (5) corresponds to the photometric error term. And the second is a regularization term on the smoothness of the surface.

This formulation can be easily generalized to accommodate multiple input images and shadow masks as follows.

$$\begin{aligned} E_2 \quad = \quad & \alpha \textstyle\sum_k \sum_{i,j} l_{ij}^k(\rho_{ij} R^k(p_{ij}, q_{ij}) - I^k(i,j))^2 + \\ & \lambda \textstyle\sum_{i,j}(u_{ij}^2 + v_{ij}^2) \end{aligned} \tag{6}$$

where $I^k(i,j)$ represents the k-th input image with corresponding reflectance map R^k, l_{ij}^k is a binary shadow mask indicating whether pixel (i,j) in the k-th image is lit by the light source or not, and ρ_{ij} is the unknown pixelwise surface albedo. This treatment is similar to photometric stereo, but solves the height field directly instead. With multiple images, the regularization term becomes much less important, and can be set close to zero. However, it may still have some effects at pixels that are lit in less than three different images.

To further incorporate the constraints in Eq. (1) and (4) into the above formulation, we notice that the constraints have the same form which looks like

$$h_{ij} - h_{i'j'} \geq d_{iji'j'}. \tag{7}$$

To enforce this kind of inequalities in a gradient-based minimization method, a differentiable half-sided parabola is adopted as a penalty function.

$$S(i,j,i',j') = \left(\begin{array}{lr} (h_{ij} - h_{i'j'} - d_{iji'j'})^2, & \text{if } h_{ij} - h_{i'j'} \\ & \leq d_{iji'j'}; \\ 0, & \text{otherwise}. \end{array} \right. \tag{8}$$

The penalty functions for all the inequalities and equalities can be inserted as additional terms into Eq. (6). The new cost function for surface reconstruction is given as follows.

$$\begin{aligned} E_3 \quad = \quad & \alpha \textstyle\sum_k \sum_{i,j} l_{ij}^k(\rho_{ij} R^k(p_{ij}, q_{ij}) - I^k(i,j))^2 + \\ & \lambda \textstyle\sum_{i,j}(u_{ij}^2 + v_{ij}^2) + \\ & \beta \textstyle\sum_k (\sum_{m_k} S^k(i_{m_k}, j_{m_k}, i'_{m_k}, j'_{m_k}) + T_k) \end{aligned} \tag{9}$$

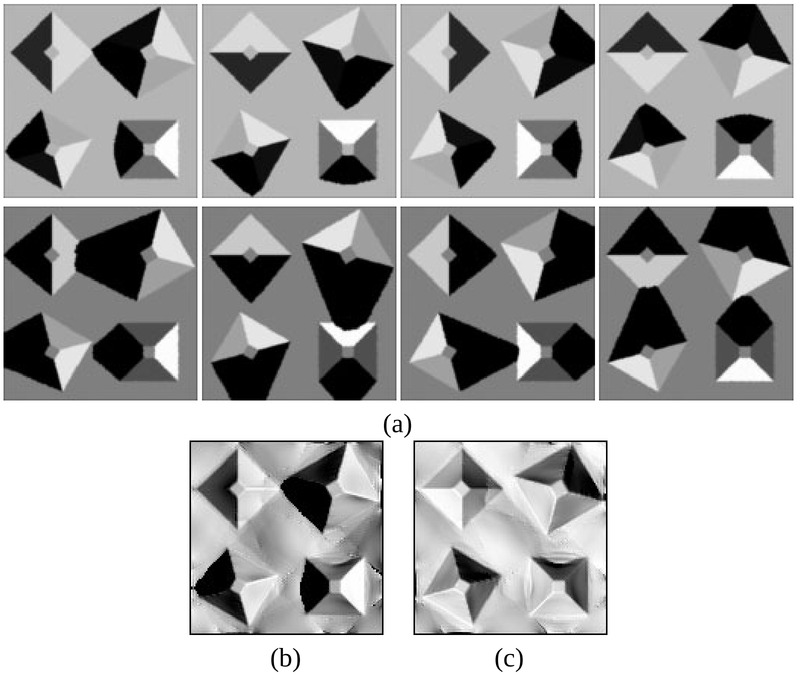
where m_k is the index of the inequality constraints from the k-th image, $S^k(i_{m_k}, j_{m_k}, i'_{m_k}, j'_{m_k})$ represents the actual penalty terms contributed by the k-th image, and T_k represents the collection of penalty terms for the equality constraints associated with shadows, such as those in Eq. (2) and Eq. (3) In our experiments, we use iterative minimization algorithms and set $\alpha = 1 - \lambda$, $\beta = 1$. λ is initialized to 0.1 and divided by a constant factor after each iteration.

All the above three cost functions can be minimized by the standard conjugate gradient algorithm [4].

3 Experiments

We have tested the algorithm using integrated shadow and shading constraints on both synthetic and real imagery.

3.1 Synthetic Data



(a)

(b) (c)

Figure 2: (a) Input images for the pyramid scene. The tilt angle of the lighting directions in the top row is 45 degrees, the bottom row 60 degrees. (b) A synthetic image of the recovered height field illuminated from the same lighting direction as in the first input images; (c) A synthetic image of the recovered height field illuminated from a novel lighting direction.

0% noise	E_2	E_3
Pyramids	3.6579	2.2424
Plaster	1.9344	1.4548
Face	4.4164	3.3335

5% noise	E_2	E_3
Pyramids	3.7621	2.2675
Plaster	1.9400	1.4089
Face	4.4522	3.4298

Table 1: Comparison of the two approaches on the three datasets: i) minimizing E_2 in Eq.(6), ii) minimizing E_3 in Eq.(9), The top table shows the RMS errors of the recovered height fields using noise free input images, and the bottom one shows the RMS errors using images with 5% noise. All numbers are given in the unit of a pixel.

Eight synthetic images were generated as input for each of the three representative datasets we chose. Four of them were lit from a tilt angle of 45 degrees and the others were lit from 60 degrees to create images with significant amount of shadow. We also generated two images for each example from the recovered height field. The first image

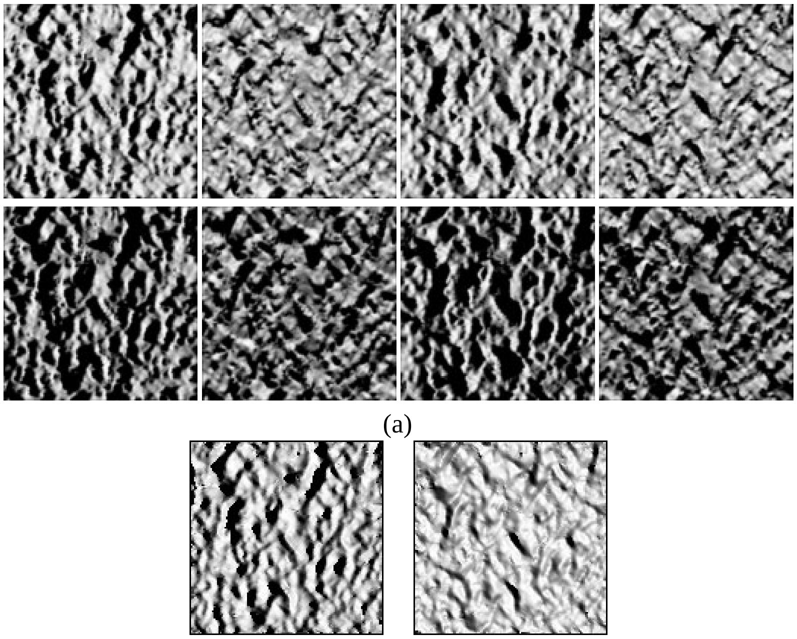

(a)
(b)
(c)

Figure 3: (a) Input images for the plaster material sample. It is lit from the same set of lighting directions as in Fig. 2. (b) A synthetic image of the recovered height field illuminated from the same lighting direction as in the first input images; (c) A synthetic image of the recovered height field illuminated from a novel lighting direction.

is lit from the same lighting direction as the first input image to verify both shadowed and non-shadowed regions. The second image is lit from a novel lighting direction which is different from the ones for the input images to show that the recovered height fields can be useful for creating images with correct appearance from novel lighting conditions. We also compared the recovered height fields with the ground truth to obtain error measurements which are shown in Table 1. In our examples, most points are lit from at least one lighting direction. The height field can be recovered from shape-from-shading or photometric stereo alone. However, the additional shadow constraints can definitely improve the accuracy of the results because shading-based techniques can introduce accumulated errors from pixel to pixel while shadow constraints are very good at enforcing long-range relative height constraints.

The first dataset is an artificial scene with four pyramids shown in Fig. 2(a). The pyramids have different height and orientation. The two synthetic images from the recovered height field are shown in Fig. 2(b)&(c). The second dataset is a previously recovered height field of a real plaster sample using the approach presented in [3]. This height field serves as the ground truth to test the algorithm in this paper although we do not know the accuracy of this dataset. The input images are shown in Fig. 3(a) and the synthetic images from the height field recovered by the current algorithm are shown in Fig. 3(b)&(c). The third dataset is a face model shown in Fig. 4(a). And Fig. 4(b)&(c) give the images generated from the recovered face. In this example, the background plane is pushed down along the shadow boundaries in some of the input images to satisfy the shadow constraints. This is because shape-from-shading related techniques are better at estimating normal orientation than at estimating height values, and generated an inaccurate initial solution for our algorithm. A similar situation was also shown in the pyramid scene. Nevertheless, our algorithm still managed to enforce the shadow constraints and make the generated images look similar to the input ones.

3.2 Real Data

We also did test on a real dataset. Three 128x128 images of a concrete sample from the CUReT database [1] were used as the input to our final algorithm. They have various amount of shadow (Fig. 5(a)-(c)). We use 15 as the intensity threshold to detect shadowed pixels. The lighting directions of the input images are actually coplanar. Traditional photometric stereo would have problem to recover the height field. However, our algorithm successfully recovered it since it exploits shadow constraints and a regularization term. Minimizing E_3 in Eq. (9) took 5 minutes on a Pentium III 800MHz processor, and the iterative procedure for enforcing bounds took another half an hour. Synthetic images

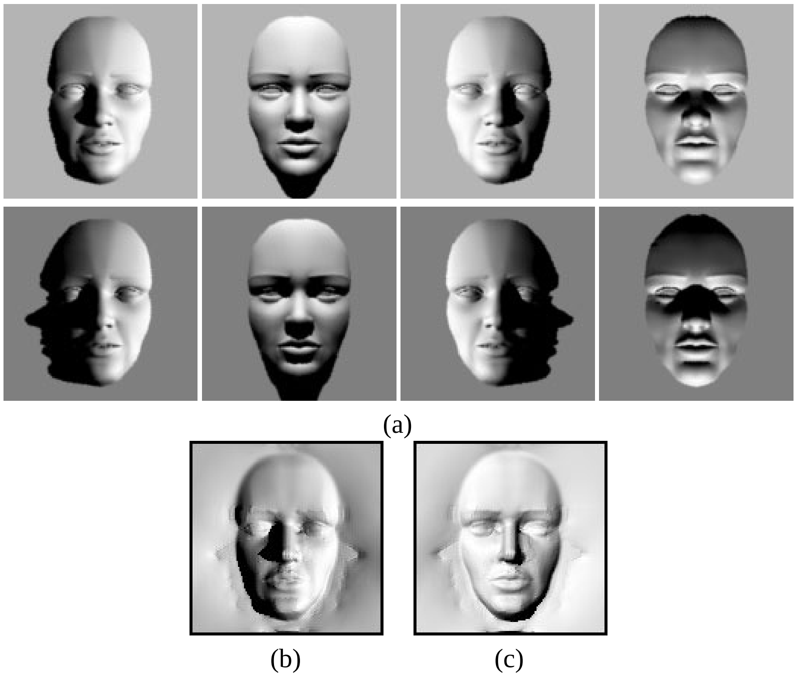


Figure 4: (a) Input images for the face model. It is lit from the same set of lighting directions as in Fig. 2. (b) A synthetic image of the recovered height field illuminated from the same lighting direction as in the first input images; (c) A synthetic image of the recovered height field illuminated from a novel lighting direction.

were generated from the recovered height field. The recovered dataset was illuminated from both original lighting directions (Fig. 5(d)-(f) of the input images and novel lighting directions (Fig. 5(g)-(h)).

4 Summary

We presented the concept of shadow graphs and proved that the shadow graph alone is enough to solve the shape-from-shadow problem from a dense set of images. We also developed a method of recovering shape from both shadow and shading constraints. A constrained optimization procedure has been developed to make the results from shape-from-shading consistent with the upper bounds derived from shadow constraints.

References

[1] K.J. Dana, B. van Ginneken, S.K. Nayar, and J.J. Koenderink. Reflectance and texture of real-world surfaces. In *Proceedings of CVPR*, pages 151–157, 1997.

[2] Y.G. Leclerc and A.F. Bobick. The direct computation of height from shading. In *Proc. of IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 552–558, 1991.

[3] X. Liu, Y. Yu, and H.-Y. Shum. Synthesizing bidirectional texture functions for real-world surfaces. In *Proceedings of SIGGRAPH*, pages 97–106, 2001.

[4] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C.* Cambridge Univ. Press, New York, 1988.

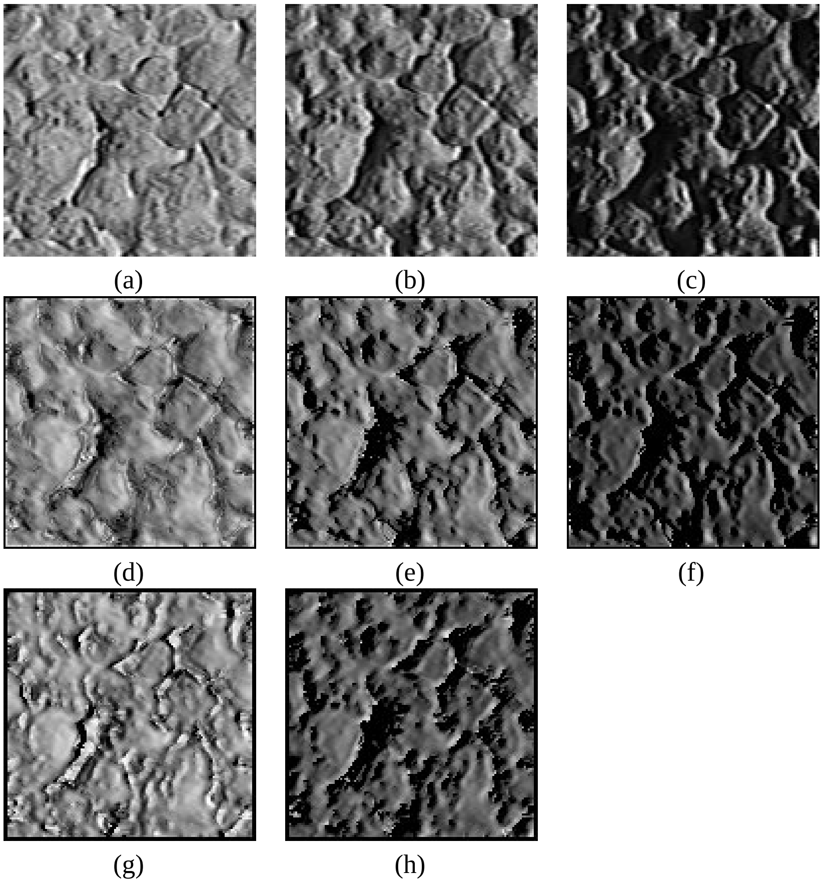


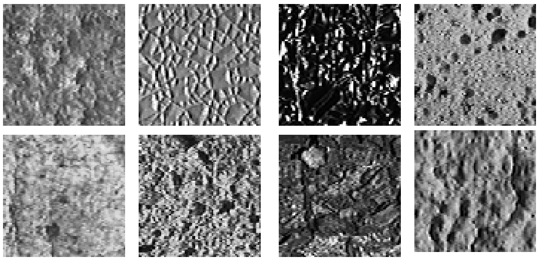
Figure 5: (a)-(c) Real images of a concrete sample; (d)-(f) synthetic images of the recovered height field illuminated from original lighting directions; (g)-(h) synthetic images of the recovered height field illuminated from two novel lighting directions.

SIGGRAPH 2002 Course on
Inverse Rendering

Modeling and Synthesizing 3D Textures

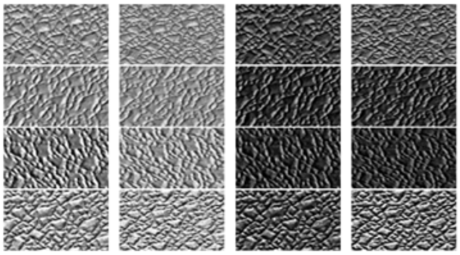
Yizhou Yu
Department of Computer Science
University of Illinois at Urbana-Champaign

Real-World Surfaces



CUReT Database [Dana et. al. 97]

Under Different Lighting/Viewing Directions



Levels of Geometric Details

- **Macrostructure**
  - Large scale geometry
- **Mesostructure**
  - Depth displacement and bumps
- **Microstructure**
  - Microfacet distributions

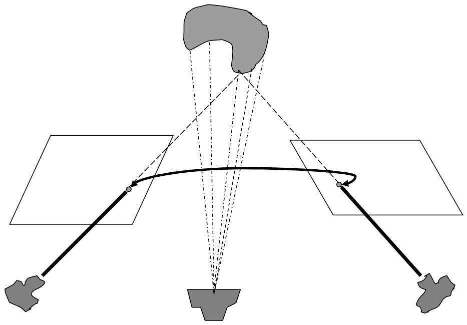
Mesostructure Recovery Methods

- **Stereo + Structured Lighting**
- **Shape-from-Shading and Photometric Stereo**
- **Shape-from-Shadow**

Stereo + Structured Lighting

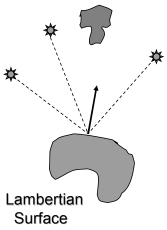
- **A projector casts a random color pattern**
- **At least two cameras for solving stereo feature registration**

Stereo + Structured Lighting



Photometric Stereo

- **A fixed camera position and at least 3 lighting directions**



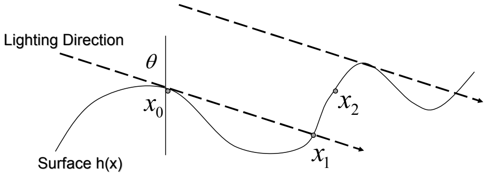
$$\vec{l}_i \bullet \vec{\mathrm{n}}_i = I_i$$

This is a linear system with the surface normal as the unknown. The normal vector is not normalized so its magnitude represents surface albedo.

Shape-from-Shading or Height

- **Shape from one single image by minimization**
  - Use an additional regularization term which is usually the summation of squared first or second finite differences of the unknowns.
  - Cost function, $\sum_{i,j}[\alpha(\rho R(p_{ij},q_{ij})-I_{ij})^2+\lambda S_{ij}]$
  - Surface normal, $(p_{ij},q_{ij},-1), p_{ij}=\frac{\partial z}{\partial x}, q_{ij}=\frac{\partial z}{\partial y}, z(x,y)$ defines the surface
  - Shape-from-Shading, (p_{ij},q_{ij}) are the unknows, S_{ij} involves first finite differences of (p_{ij},q_{ij})
  - Shape-from-Height z_{ij} are the unknows, S_{ij} involves second finite differences of z_{ij}

Shape-from-Shadow



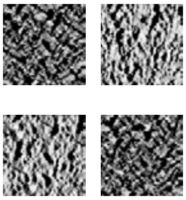
$$h(x)\le h(x_0)-\frac{\|x-x_0\|}{\tan\theta}, \forall x\in[x_0,x_1]$$

$$h(x)\le h(x_2)+\frac{\|x-x_2\|}{\tan\theta}, \forall x \text{ on left of } x_2$$

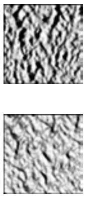
Integrated Shading and Shadow Constraints

- **Photographs of surface mesostructures usually contain both shading variations and shadows.**
- **Minimize the shape-from-shading cost function subject to shadow constraints**
  - Constrained optimization

Results from Constrained Optimization



Input Images of a Plaster Sample



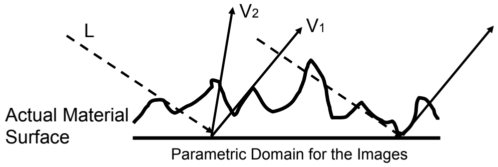
Synthetic Images from the Recovered Height Field

Appearance Models

- **BRDFs**
  - Surface microstructure
- **Reflectance Maps**
  - Heterogeneous surface microstructure
- **Bump/Displacement Maps**
  - Heterogeneous surface mesostructure
- **Bidirectional Texture Functions**
  - Both microstructure and mesostructure

Bidirectional Texture Functions (BTF)

- **A collection of images of the same surface under different lighting and viewing directions**
  - [Dana, Ginneken, Nayar & Koenderink 97]
  - $L \times V \Rightarrow I$



The same point in the parametric domain may correspond to different points on the material surface from different viewpoints.

Why BTF?

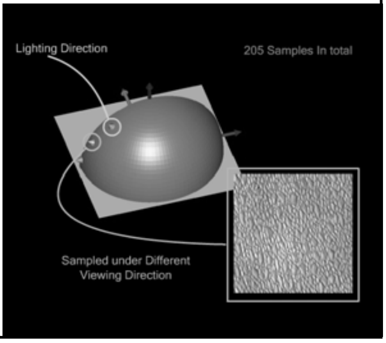
- **Visual Effects from Small-Scale Geometric Details (3D Textures)**
  - Shadowing
  - Occlusion and foreshortening
  - Spatially varying normal orientations
  - Inter-reflection
- **Spatially Varying Reflectance Properties**

The Problem

- **Acquiring a dense set of images in a 4D space is extremely expensive.**
- **Our Approach: BTF Synthesis**
  - To generate a continuous BTF
  - From a sparse set of images
  - Under any lighting/viewing setting

Related Work I

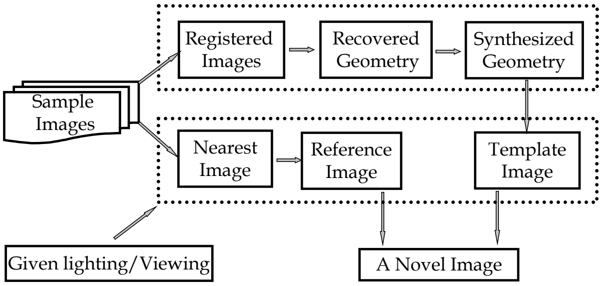
- **CUReT Database**
  - Sparse sampling: 205 images in 4 dimensions
  - Unevenly covering the lighting/viewing space
  - http://www.cs.columbia.edu/cave/curet/



Related Work II

- **BTFs and 3D Textures**
  - [Dana et. al. 97, 99], [Leung & Malik 99]
- **2D Texture Synthesis**
  - [Heeger & Bergen 95], [De Bonet 97], [Simoncelli & Portilla 98], [Zhu, Wu & Mumford 98], [Efros & Leung 99], [Wei & Levoy 00] ….
  - Patch-base texture synthesis [Xu, Guo & Shum 00]
- **Recovering Normal Maps**
  - [Rushmeier et. al. 97]

Overview of BTF Synthesis


Registered Images

Recovered Geometry

Synthesized Geometry

Sample Images

Nearest Image

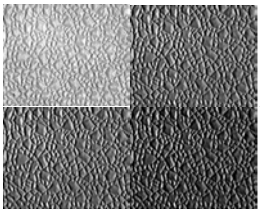
Reference Image

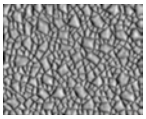
Template Image

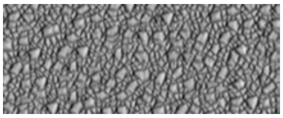
Given lighting/Viewing

A Novel Image

Geometry Recovery and Synthesis


Input: registered images with varying lighting directions


Recovered geometry


Synthesized geometry

Shape from Shading

- **Shape from Shading through Normals**
  - E.g., [Horn & Brooks 86]
- **Direct Height from Shading for Lambertian Surfaces [Leclerc & Bobick 91]**

$$E = \sum_{i,j} [\alpha \cdot (\rho \cdot R(p_{ij}, q_{ij}) - I(i,j))^2 + \lambda \cdot (u_{ij}^2 + v_{ij}^2)]$$

$$R(p_{ij}, q_{ij}) = \vec{n}_{ij} \cdot \vec{L} = (x_L \cdot p_{ij} + y_L \cdot q_{ij} - z_L) / \sqrt{p_{ij}^2 + q_{ij}^2 + 1}$$

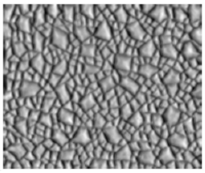
Knowns	Unknowns
$I(i,j)$ x_L y_L z_L	ρ $p_{i,j}$ $q_{i,j}$

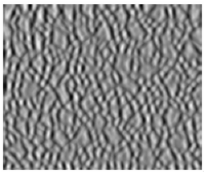
Modified Height from Shading

- **Modifications**
  - Pixel-wise albedo function ρ_{ij}
  - Shadow pixel classification $\eta(i,j)$
  - Adaptive geometry smoothness $\mu(i,j)$
  - Using multiple input images Σ_k

$$E = \sum_{i,j} \{ \alpha \cdot (\sum_k (\rho_{ij} \cdot R_k(p_{ij}, q_{ij}) - I_k(i,j))^2 \cdot \eta_k(i,j)) + \lambda \cdot (u_{ij}^2 + v_{ij}^2) \cdot \mu(i,j) + \gamma \cdot (s_{ij}^2 + t_{ij}^2) \}$$

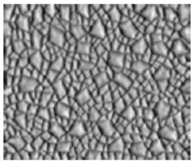
A Comparison of Recovered Geometry


Modified

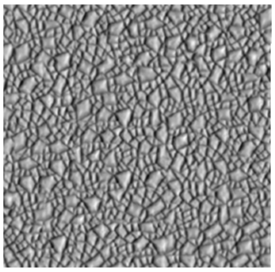

Leclerc & Bobick

These images are gray-scale coded

Geometry Synthesis

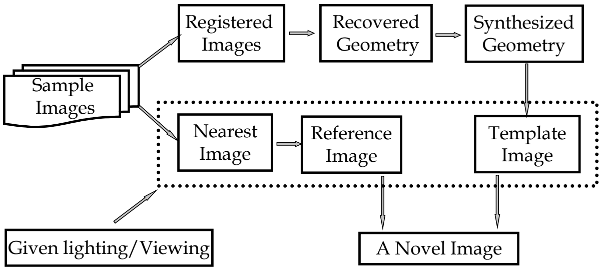

Recovered

– Patch-based sampling

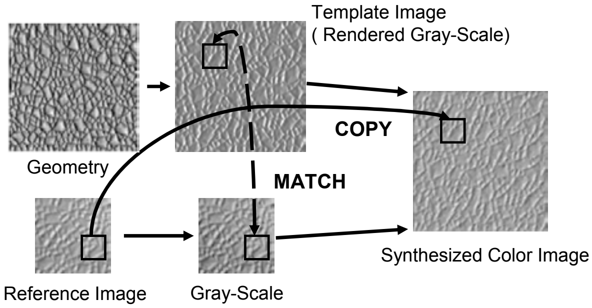

Synthesized

Overview of BTF Synthesis

Appearance preserving texture synthesis



Appearance Preserving Texture Synthesis



Template Image and Reference Image

Template Image

- Rendered from synthesized geometry given lighting/viewing
- Correct occlusion, fore-shortening, and shadows
- Approximate shading

Reference Image

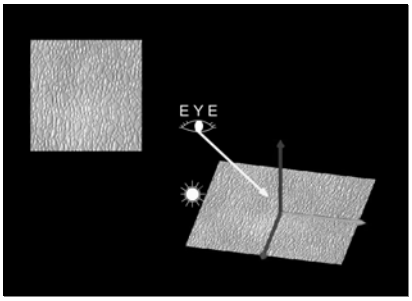
- Taken from the database with same lighting/viewing setting
- Correct color and shading variations
- Inconsistent geometry with the template image

Reference Image Generation

- **Pick one of the “nearest” sample images**
  - Closest viewing and lighting directions
  - Isotropic materials: “closest” under rotation
- **Warp it to the current viewing/lighting setting**
  - Fore-shortening
    - *Tilting angle of the viewing direction*
  - Lighting effects
    - *Azimuth angle of the lighting direction*

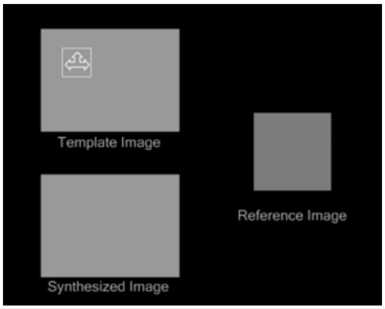
Three-step Warping Method

- **Back-project onto the material surface plane**
- **Align light azimuth by rotation in the material surface**
- **Re-project onto the desired viewing plane**



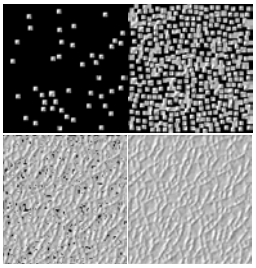
Synthesizing Novel Image

- **Block Copying**
  - Select a block from the template image
  - Find its best match in the reference
  - Copy the patch onto the synthesized image

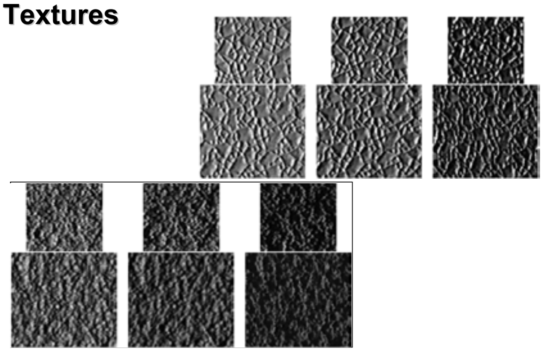


Block Copying

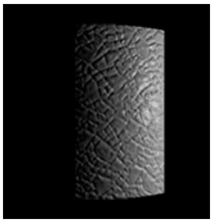
- **Details**
  - Feature ordering
    - *Best features fill first*
  - Feature matching
    - *Multiple block sizes*
  - Matching criteria
    - *Optimal block with normalized correlation*



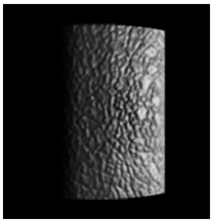
Reference Images vs. Synthesized Textures



Bump Mapping vs. BTF Mapping

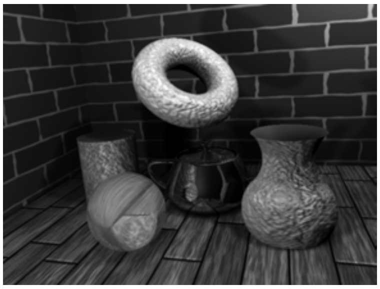


Bump Mapping



BTF Mapping

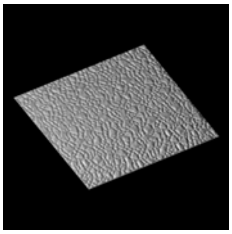
Two Synthetic Images with BTF Mapping

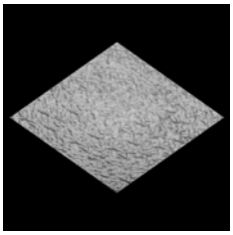


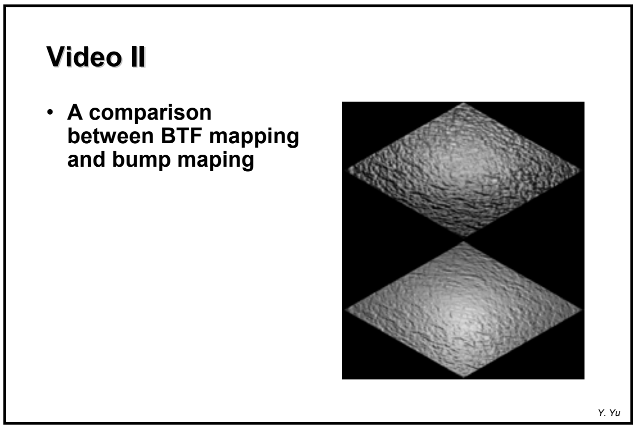
Two Synthetic Images with BTF Mapping

Video I

- **Surface appearance with a moving point light source**

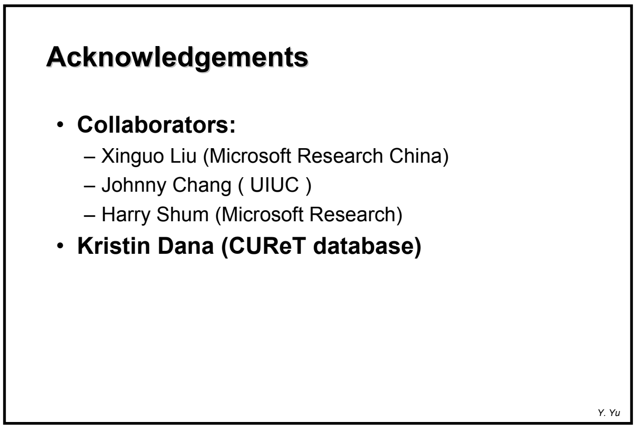






Video II

- A comparison between BTF mapping and bump maping



Acknowledgements

- **Collaborators:**
  - – Xinguo Liu (Microsoft Research China)
  - – Johnny Chang (UIUC)
  - – Harry Shum (Microsoft Research)
- **Kristin Dana (CUReT database)**

Synthesizing Bidirectional Texture Functions for Real-World Surfaces

Xinguo Liu [†‡] **Yizhou Yu** [††] **Heung-Yeung Shum** [†]

[†] Microsoft Research, China
[††] University of Illinois at Urbana-Champaign
[‡] State Key Lab. of CAD&CG, Zhejiang University

Abstract

In this paper, we present a novel approach to synthetically generating bidirectional texture functions (BTFs) of real-world surfaces. Unlike a conventional two-dimensional texture, a BTF is a six-dimensional function that describes the appearance of texture as a function of illumination and viewing directions. The BTF captures the appearance change caused by visible small-scale geometric details on surfaces. From a sparse set of images under different viewing/lighting settings, our approach generates BTFs in three steps. First, it recovers approximate 3D geometry of surface details using a shape-from-shading method. Then, it generates a novel version of the geometric details that has the same statistical properties as the sample surface with a non-parametric sampling method. Finally, it employs an appearance preserving procedure to synthesize novel images for the recovered or generated geometric details under various viewing/lighting settings, which then define a BTF. Our experimental results demonstrate the effectiveness of our approach.

CR Categories: I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—modeling and recovery of physical attributes I.3.7 [**Computer Graphics**]: Three-dimensional Graphics and Realism—color, shading, shadowing, and texture I.4.8 [**Image Processing**]: Scene Analysis—color, photometry, shading

Keywords: Bidirectional Texture Functions, Reflectance and Shading Models, Texture Synthesis, Shape-from-Shading, Photometric Stereo, Image-Based Rendering.

1 Introduction

Surface appearance modeling has drawn much attention from researchers since the dawn of computer graphics [6, 2, 3]. Appearance models are closely related to geometry. There are three levels of scales in geometry, namely, the macrostructure level, the mesostructure level [20] and the microstructure level. A geometric model usually refers to the macrostructure level, and is often specified as a set of polygonal and/or curved surfaces. The mesostructure level includes geometric details that are relatively small but still visible such as bumps and dents on a concrete surface. The microstructure level involves surface microfacets that are visually indistinguishable by human eyes. The last two levels of geometry contribute to surface appearance properties. For instance, bump maps are used to model the mesostructure level while BRDFs model the microstructure level.

We are interested in modeling appearance at the mesostructure level for real-world surfaces such as concrete surfaces, crumpled papers, pebbles and carpets. The presence of such small-scale details gives rise to a rich set of visual effects, including mutual shadowing, interreflection, occlusion and foreshortening, in addition to varying surface normal orientations. Without properly modeling such effects, surfaces would look too smooth to be real. Bump and normal mapping techniques can model the effects caused by changing normal orientations but not others. However, all the above visual effects for bumpy surfaces (as well as spatially varying reflectance) can be captured by bidirectional texture functions (BTFs).

A BTF is defined as a six dimensional function with a 2D texture associated with each possible combination of lighting and viewing directions which account for the other four dimensions. Thus, a BTF has two additional dimensions for textures compared to a 4D BRDF. The pioneering work by Dana et. al. [7, 9] on BTFs took an experimental approach that acquired images of material samples under various combinations of lighting and viewing directions. Their work led to the CUReT database that has a sparse set of images partially covering the lighting and viewing hemispheres for each material sample. Such a sparse sampling is not adequate to faithfully represent material appearances for graphical rendering purposes. On the other hand, acquiring a dense set of samples for BTFs is prohibitive in practice because a BTF has six dimensions.

In this paper, we study the following problem: given discrete samples of the BTF of a real-world surface with mesostructure details, can we synthesize the continuous BTF ? Specifically, from a sparse set of sample textures, can we synthesize a new texture at any given lighting/viewing setting ? Moreover, can novel BTFs be synthesized from a given BTF to emulate the stochastic properties of texture images of the given BTF under all lighting/viewing settings, similar to 2D texture synthesis ?

We present an algorithm to solve the above problems, by exploiting both geometric and photometric properties of material samples and effectively integrating them together. In our work, we use sample textures from the CUReT database. We first recover the height field on a material from a small number of images and synthesize novel 3D structures for the same material. The recovered or synthesized height fields are used for rendering synthetic images of the material under different combinations of lighting and viewing

[†] hshum@microsoft.com

[††] yizhouy@acm.org, www-faculty.cs.uiuc.edu/∼yyz

[‡] Xinguo Liu participated in this work when he was visiting Microsoft Research, China.

directions. The rendered images are then fed into an appearance preserving texture synthesis procedure, along with the set of acquired sample images, to synthesize high-quality sample images of the corresponding BTF.

In summary, this paper has the following three major contributions.

- First of all, we propose a novel hybrid approach for studying appearance models, which can be a useful idea for bridging other geometry-based and image-based techniques.
- Second, we introduce an algorithm that synthesizes complete BTFs, including the statistical structure and statistical texture, from a sparse set of sample images.
- Third, we develop a method to recover bump maps from photographs of real world materials by adapting an existing shape-from-shading algorithm.

The remainder of this paper is organized as follows. The next section provides the necessary background and related work. Section 3 gives an overview of our algorithm. The details of our algorithm will be discussed in Section 4 (geometry recovery), Section 5 (geometry synthesis) and Section 6 (BTF synthesis). And Section 7 presents our results.

2 Background and Related Work

2.1 BTFs

A BTF can be regarded as a mapping from the 4D space of lighting and viewing directions to the space of all 2D images:

$$\tau : L \times V \to I \qquad (1)$$

where L and V are lighting and viewing directions parameterized by a pair of tilt and azimuth angles (θ, ϕ), I is a mapping itself from R^2 to the RGB color space. This definition basically views a BTF as a collection of images, and favors texture analysis and synthesis.

We assume every image in a BTF observes a homogeneous Markov Random Field (MRF) model, which is a common assumption in the texture synthesis literature [12, 38, 41, 45]. Because of the visual effects caused by varying lighting and viewing directions, each image in a BTF has a distinct MRF model. MRF methods model an image as a realization of a local and stationary random process. That is, each pixel of a texture image is characterized by a small set of spatially neighboring pixels, and this characterization is the same for all pixels. A MRF model allows us to view every BTF image as a collection of local neighborhoods. Each realization of the MRF model can be viewed as a random rearrangement of these local neighborhoods in the image plane. We also assume that the height field inducing the BTF on a material sample observes a homogeneous MRF model which enables us to synthesize novel instances of the height field using existing texture synthesis algorithms [12, 38].

Alternatively, we can follow the line of previous work on light fields and plenoptic functions [1, 23, 14], and consider a BTF as a specific 6D reflectance field if ignoring wavelength and fixing time:

$$T = T(\theta_i, \phi_i, x, y, \theta_r, \phi_r) \qquad (2)$$

which provides the connection between reflected flux in a direction (θ_r, ϕ_r) and incident flux in another direction (θ_i, ϕ_i) at the same point (x, y) on a material sample. This is a simplified version of a more general 8D reflectance field in [10] for a general 3D object enclosed by a convex hull by assuming parallel light sources.

Since a BTF includes images of a material under all possible lighting directions, it essentially provides lighting-independent appearance properties of the material. Novel images of the material under an arbitrary lighting condition can be synthesized from its BTF by properly integrating the contribution from each individual BTF image.

2.2 Related Work

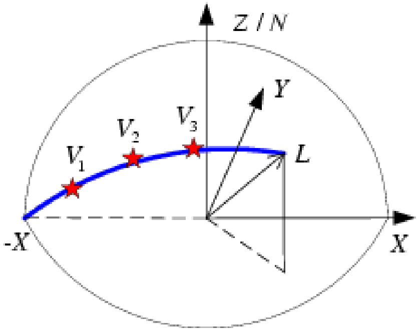


Figure 1: The sampled lighting and viewing directions of the BTF images in the CUReT database. Three typical viewpoints (V_1, V_2 V_3) are shown in between the lighting direction L and the negative X-axis.

Previous work on BTFs aims to capture appearance data for natural materials and represent them efficiently [7, 8, 9, 22]. However, each material in the CUReT database [7] has a sparse 4D sampling of only 205 images under 205 different viewing and lighting conditions. Specifically for each lighting direction L on the half hemisphere with $Z > 0$ and $Y < 0$, seven images were captured at viewpoints along the short spherical arc between the lighting direction and the negative X-axis. Fig. 1 illustrates a few such viewpoints.

Although [18] introduces a technique to precompute some of the visual effects for regular synthetic bump structures, stochastic details on real-world materials are not modeled.

Part of our work is inspired by the recent success of 2D texture synthesis [17, 4, 36, 12, 38, 41, 45]. In 2D texture synthesis, from a texture sample, a new texture is synthesized such that, when perceived by a human observer, it appears to be generated by the same underlying stochastic process. To the best of our knowledge, however, there has been no previous effort on synthesizing 6 dimensional BTFs where textures are under different stochastic processes with different viewing/lighting settings.

Our work also shares similarities with previous work on image-based rendering in that novel images of a real scene are generated from acquired image samples, without [25, 23, 14, 37, 34, 35] or with [11, 33, 31, 32, 44, 43, 42] the knowledge of the scene geometry. In particular, our 6D BTF synthesis problem has a similar spirit as plenoptic modeling where a continuous 5D plenoptic function is synthesized from discrete samples. Note that BTFs are different from surface light fields [39] and view-dependent textures [11] since the latter two only capture surface appearance under fixed lighting conditions. Appearance models from real images have become an active research topic in graphics [31, 32, 24, 44, 43, 10].

3 Overview

For the rest of the paper, a *viewing/lighting setting* means a combination of viewing and lighting directions. The set of input images to our method are called the *sample images* of a real material.

Two factors affect the appearance of bumpy surfaces: the 3D structure of the bumps and spatial reflectance variations. The geo-

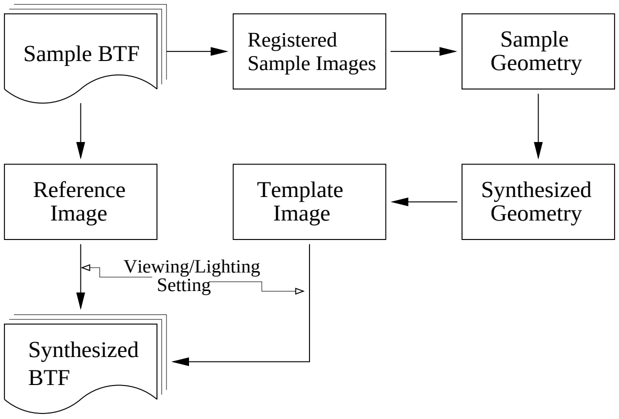


Figure 2: The flow chart of the overall BTF synthesis algorithm.

metrical structure produces shadows and occlusions. Both factors together generate texture and shading effects including highlights. The first step in our approach is to recover the surface geometry. However, the statistical properties of the 3D structure are more important than the exact geometry itself because of the random spatial distribution of the 3D features.

We represent the surface geometry using a height field on top of a supporting plane of the surface. From a collection of images taken under different viewing/lighting settings, we derive the geometry and its approximate albedo map using a modified version of the height from shading method in [21]. The modified method can recover depth variations by incorporating multiple input images in the formulation. More importantly, we explicitly handle shadows and possible highlights in the images. These input images are first globally registered using the video mosaicing technique developed in [37].

From the recovered height fields we can synthesize other statistically equivalent height fields that can then be used to generate new BTFs. By considering the height field as a sample (gray-scale) image, we can apply previously developed 2D texture synthesis algorithms to synthesize novel height fields. To synthesize an accurate BTF image from a novel viewing/lighting setting, one possibility could be texture mapping the recovered height field from the sample images followed by viewing/lighting dependent interpolation. Unfortunately, there are two issues here. First, for the recovered height field, the sample images may be irregularly distributed so that there is no neighboring image for certain viewing/lighting settings. Second, for the synthesized novel height field, we do not have any sample images at all. So simply interpolating from neighboring images becomes infeasible. Instead, we propose a local appearance preserving texture synthesis procedure.

Our BTF synthesis algorithm can synthesize a BTF image from two input images. One is a *template image* which is synthetically rendered with shadows from the input height field. The other is a *reference image* which is selected from the set of real sample images of the material. Both images should have the same viewing/lighting setting as that of the BTF image being synthesized. Either a constant albedo or the recovered albedo map can be used for rendering the template image. This template image exhibits correct shadows, occlusions for the synthesized height field and approximate shading effects associated with the material. But it does not have very accurate color, shading and mutual illumination at each point since we do not have point-wise 4D reflectance functions on the material surface at this stage. On the other hand, the reference image has the correct color and shading information. But its underlying geometry may be different from our input height field. Our method can combine the useful information from these two images. The flow chart of our BTF synthesis algorithm is illustrated in Fig. 2.

The technique of copying texture from the reference image is partially inspired by the block-copy synthesis method [41]. Compared with the pixel-wise synthesis scheme adopted in some previous 2D texture synthesis algorithms, the block-copy synthesis scheme is much faster while maintaining feature integrity. However, one must be careful about choosing the right places and order to paste blocks. Those regions with prominent features, such as corners and edges, should be synthesized first with higher priority.

Ideally, the reference image for our texture synthesis should have the same viewing and lighting directions as the template image. Such a reference image may not be available because of the limited size of the input image collection. Our solution is to find the “nearest” image, and then make it consistent with the desired viewing/lighting setting by warping. A more serious problem is that the sample image collection may not be distributed uniformly in the 4D space for lighting and viewing directions. A subspace may not have any corresponding images at all. When this happens, we exploit a certain level of isotropy exhibited by the material samples. A material sample has a large number of tiny bumps. Being “isotropic” means that for any bump B_i and any rotation angle between 0 and 360 degrees, there always exists another bump B_j whose shape and reflectance are rotated versions of B_i by the given angle. Most natural materials approximately satisfy this condition except for very elongated structures such as straw. Although a single “nearest” reference image is enough to synthesize a BTF image, the resulting image may be noisy. More reference images can be used to further improve the synthesis quality.

With the above introduction to our method, we can summarize the *requirements* we need to impose on the set of input sample images: a) The images should cover a reasonable number of random bumpy structures of a material for the MRF model to work well; b) for anisotropic materials, we need a sparse set of images covering the 4D space of viewing and lighting directions; c) for isotropic materials, we only need images covering a 3D subspace of the viewing/lighting settings since the azimuth angle of the lighting direction is no longer important.

4 Geometry Recovery

Geometry recovery is a classic problem in computer vision. Many kinds of geometry recovery algorithms have been proposed using different visual cues such as disparity, shading, focus and defocus [13]. We decided to use shading as the major cue to account for shading variations, which are present in the set of input sample images because of a changing illumination direction.

4.1 Shape from Shading

We adopt a shape-from-shading technique called height from shading [21]. Unlike most shape-from-shading and photometric stereo methods [19, 5, 40, 27], this technique computes a height field directly rather than through surface normals. This technique is efficient and robust. The geometry recovery is formulated to minimize the following energy functional [21]:

$$E = \sum_{i,j} [\alpha(\rho R(p_{ij}, q_{ij}) - I(i,j))^2 + \lambda(u_{ij}^2 + v_{ij}^2)] \quad (3)$$

where ρ is the surface albedo, I is the observed image intensity, p_{ij}, q_{ij}, u_{ij}, v_{ij} are the symmetric first and second finite differences of the surface height field $\{z_{ij}\}$, α and λ are two constant coefficients, and R is the Lambertian reflectance model:

$$R(p_{ij}, q_{ij}) = \vec{n_{ij}} \cdot \vec{L} = \frac{x_L p_{ij} + y_L q_{ij} - z_L}{\sqrt{p_{ij}^2 + q_{ij}^2 + 1}} \quad (4)$$

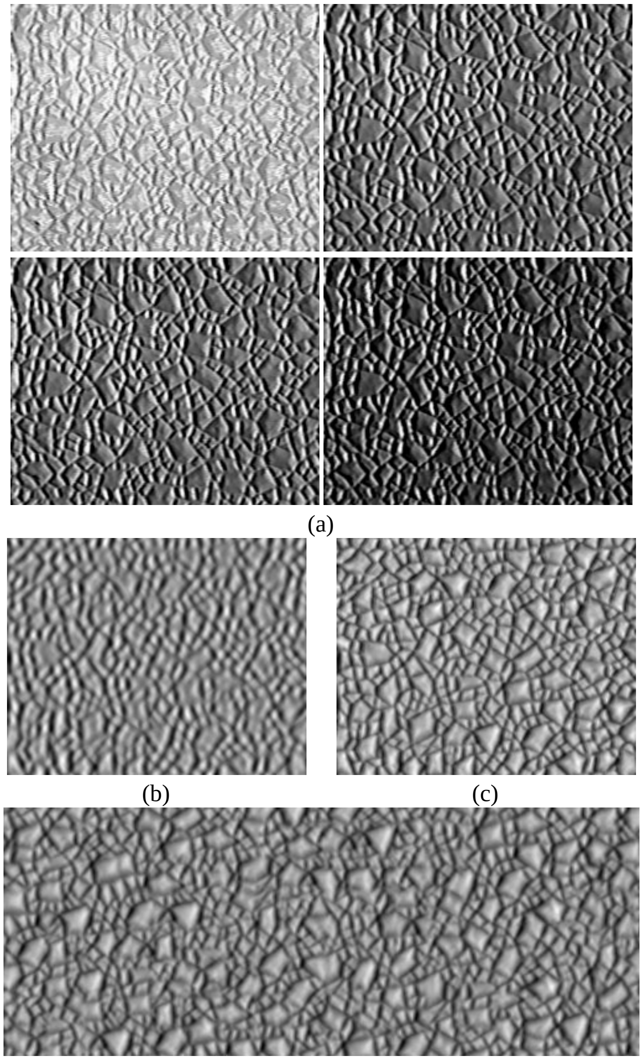


Figure 3: (a) Four calibrated gray-scale images used for recovering a height field; (b) a recovered height field from the height-from-shading algorithm in [21]; (c) a recovered height field from our revised algorithm; (d) a synthesized height field generated from (c). Height fields are visualized as gray-scale images in (b)-(d).

where $L = (x_L, y_L, z_L)$ is the unit vector of the light source direction.

The first term in Eq. 3 corresponds to the photometric error term. And the second is a regularization term on the smoothness of the surface. These two terms are balanced by two weights α and λ.

4.2 Modifications

It has been shown [21] that the above technique generates good quality height fields for smooth objects such as human faces. To deal with shadows, occlusion, or specular highlights which exist commonly on mesostructure surfaces, we make the following three significant modifications.

- **Albedo function.** An albedo function $\{\rho_{ij}\}$ is defined over the surface instead of a constant value. Accordingly, a regularization term for albedo variation is added to the original energy function:
$$\sum_{i,j}(s_{ij}^2 + t_{ij}^2),$$
where s_{ij} and t_{ij} are the symmetric second finite differences of the surface albedo function $\{\rho_{ij}\}$. This albedo regularization term is also weighted by another coefficient γ.

- **Classification of pixels.** Unlike [21] where each pixel from a single image has equal weight for calculating photometric error, we use a small number of registered images $\{I_k\}$ and set a different weight $\eta_k(i,j)$ for each pixel's contribution according to its type: normal pixel, shadow pixel and highlight pixel. Normal pixels are weighted more than shadow and highlight pixels which are treated as outliers. Following a Lambertian reflectance model, we detect outliers using robust statistics [15]. Although [40] proposed an approach to use multiple images, it does not have pixel-wise adaptive weights and regularization terms which are crucial to deal with shadows and specular highlights.

- **Geometry smoothness.** The discontinuity features on a surface (e.g., sharp creases, ridges and grooves) call for varying weights $\mu(i,j)$ for all surface points in the regularization term, such that they are not smoothed too much in the recovered geometry. In our implementation, we first determine the degree of smoothness at all points of the surface by detecting intensity edges of input images. Then points with larger edge responses are assigned smaller weights of regularization. Smaller weights does not necessarily lead to geometric discontinuities. Note that intensity edges may be caused by shadow boundaries or discontinuous reflectance on the surface as well, and might have been accounted for in the albedo function and shadow classification.

All of the above modifications lead to the following form of objective energy function:

$$\begin{aligned} E &= \sum_{i,j}\{\alpha[\sum_k(\rho_{ij}R_k(p_{ij},q_{ij}) - I_k(i,j))^2\eta_k(i,j)] \\ &\quad +\lambda(u_{ij}^2 + v_{ij}^2)\mu(i,j) + \gamma(s_{ij}^2 + t_{ij}^2)\} \end{aligned}$$

where

$$R_k(p_{ij},q_{ij}) = \vec{n_{ij}} \cdot \vec{L_k} = \frac{x_{L_k}p_{ij} + y_{L_k}q_{ij} - z_{L_k}}{\sqrt{p_{ij}^2 + q_{ij}^2 + 1}} \qquad (5)$$

Finally, the non-linear minimization problem is solved numerically using the conjugate gradient algorithm [30]. An example of a recovered height field is shown in Fig. 3(c). It is recovered from the four images shown in Fig. 3(a). For comparison, Fig. 3(b) shows the recovered height field from the original algorithm in [21].

5 Generating New Geometry

If the recovered height field is regarded as a gray scale image by converting height values into pixel intensities, we can apply 2D texture synthesis algorithms to generate new surface geometry. The height field image indeed exhibits the stochastic properties which make texture synthesis algorithms work well. Our synthesis algorithm is an accelerated version of the non-parametric sampling method [12], much similar to the multi-resolution algorithm in [38]. It is based on the MRF texture model, which assumes that pixel values in a texture are determined probabilistically by their surrounding patches [12]. An optimized K-D tree based searching algorithm [26] is applied to accelerate the patch matching process. An example of synthesized geometry is shown in Fig. 3(d).

6 BTF Synthesis

Given a natural material and a sparse set of sample images of its BTF, the goal of our synthesis procedure is to generate a complete BTF for a height field that has statistically equivalent mesosturctures as the considered material surface. Our approach works in the same way for both recovered and synthesized height fields. Basically we need to synthesize images for all viewing/lighting settings. This task cannot be done by simply running a general 2D texture synthesis algorithm on each image separately, since consistent underlying geometry for a changing viewing/lighting setting isn't guaranteed. As shown in Fig. 4, all the synthesized images are perceived to have the same statistical features as the original material under corresponding viewing/lighting settings. But they cannot be images of the same BTF because the perceived mesostructure details change from image to image. The reason is that all images are synthesized independently, and no constraints on the underlying mesostructure details are imposed.

BTF images arise not only from spatial variations of surface reflectance, but also from spatial variations of surface geometry, which lead to local shading, highlights, inter- reflection, shadowing and occlusion of local surface elements by neighboring elements. Note that geometry plays an important role in the generation of texture appearance. We take advantage of geometry to render a template image, and then use it as a constraint during texture synthesis.

6.1 Local appearance preserving texture synthesis

One of the most critical things in synthesizing BTF is to generate, under a varying viewing/lighting setting, consistent changes of features caused by the underlying geometry. Therefore, we generate a synthesized gray scale image of the geometry with features, such as shadows, occlusions and highlights, under each given viewing/lighting setting, and use it as a template texture during texture synthesis. We can tolerate minor errors in the recovered or synthesized geometry because the geometry is never used directly for producing the final images, and is only used for rendering the intermediate template images. We would like to make sure that every pixel in the final images is from somewhere in the input sample images to preserve the appearance of non-geometric features as well. Obviously, we should take pixels from the reference image, i.e., the sample image which was taken under the same viewing/lighting setting as the BTF image being synthesized.

With a real reference image and a synthetic template image, we can synthesize a final BTF texture efficiently. We do it block-by-block, rather than pixel-by-pixel. The main idea of our block-wise texture synthesis is: for each pixel of the template image, a block of appropriate size in the reference image is found, which best matches the corresponding neighboring patch in the template image, and then copied to the corresponding region centered at the pixel. This process is repeated until the synthesized image is filled. Since the reference image is taken from a camera and the template image is synthetically rendered, similar features of the material sample may have different intensity and color contrasts in the two images due to different image formation pipelines. Therefore, the reference image is converted into a gray scale image and the histograms of the gray scale template image and reference image are equalized first [17]. Of course, the copied blocks are taken from the original colored reference image.

Our synthesis algorithm consists of the following three main steps:

- Feature ordering
- Feature matching
- Block copying

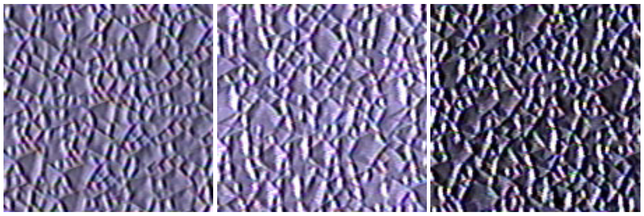
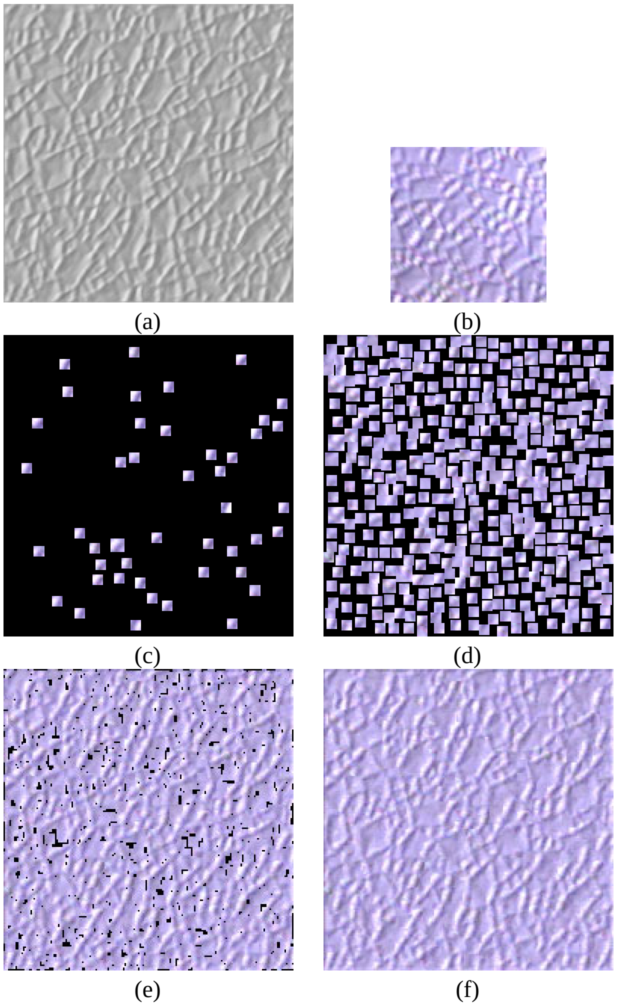


Figure 4: Three independently synthesized textures are generated from three sample images taken at three viewing/lighting settings. A 2D texture synthesis algorithm is used without knowledge of the underlying mesostructure details. However, putting together these synthesized textures does not give us a BTF because they can not be perceived to have the same geometry. Look, for example, at the upright corners of these three images. Clearly they are different geometrically.



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

Figure 5: Input images and different stages of feature preserving BTF synthesis. (a) The synthetic gray scale template image; (b) The real reference image; (c) An initial stage during block copying when only pixels with high priority are considered; (d) towards the end of block copying, most pixels are covered by nonoverlapping blocks while gaps among blocks remain as holes; (e) holes are being filled up with tiny blocks that are allowed to overlap with existing blocks; (f) the final result of a synthesized BTF image after hole filling. (f) is synthesized from (a) and (b).

First, we run the Harris feature detector [16] to prioritize all the pixels of the template image according to the gradient variation in the image, such that significant features such as corners will be considered first for synthesis. Then starting from the highest prioritized pixel, an appropriate surrounding block centered at this pixel is built and used to find a block with similar features in the reference image. At last, the found blocks are copied from the reference image to the corresponding positions in the template image.

Optimally the size of the blocks should be set adaptively, which is never an easy task. In practice, our method tests blocks with a few different predefined sizes and picks the best one. The predefined blocks consist of a series of $N \times N$ squares centered at the pixels. There are still two issues that we need to address. The first one is how to measure the similarity of two blocks, one of which is from the template image, and the other is from the reference image. A criterion can be established on any texture model. For simplicity and efficiency, we use the summation of squared differences (SSD), as in some 2D texture synthesis algorithms. The second one is how to compare the matching quality among blocks with different sizes. Since it is unfair to directly compare SSDs between blocks of different sizes, we normalize the SSDs by the number of pixels in each block.

For efficiency consideration, we do not allow blocks to be overlapped at first, so as to prevent pixels from being copied repeatedly. However this leads to some unfilled pixels after block copying. Therefore we need to do hole filling at the end. In fact hole filling is very similar to block copying except that only a smaller size for the blocks is used and the small hole-filling blocks are allowed to overlap with other copied blocks. Fig. 5 illustrates the block copying and hole filling processes. The block matching problem is equivalent to finding the nearest neighbor in a high dimensional space. This has been extensively studied, and many acceleration techniques have been put forward. In this paper, we take a K-D tree based searching algorithm [26] to accelerate our matching process.

The key to our algorithm, or why we can simply copy feature blocks at different locations from the reference image, is that we assume a Markov Random Field model for each BTF image, which enables us to view an image as a realization of the underlying stochastic process which randomly rearranges the collection of local neighborhoods in the image plane, as mentioned in Section 2.1. Note that the height field is not directly involved in the block matching process. But for recovered height fields, we could register all input images with the height field and find the best matching block by running block matching on the height field. This can probably generate more consistent shadowing and occlusion effects in the synthesized images with different viewing/lighting settings. However, registration is hard and interpolation of BTF is not obvious. In practice, we have found that block matching without a height field can generate very good results.

6.2 Reference image generation

In the above synthesis algorithm, a reference image captured from the real world with the same viewing/lighting setting as that of the template image is assumed. However that can hardly be achieved in most situations since dense sampling of the 4D space of viewing/lighting settings is prohibitive, and we can only capture a limited collection of images. The CUReT database mentioned in Section 2.2 is such an example. However, the appearance of BTF images heavily depends on their viewing/lighting settings. In practice, for those viewing/lighting settings not sampled, we can find a sample image with the "nearest" viewing/lighting setting using a distance metric between two viewing/lighting settings.

Let $C_i =< V_i, L_i >=< (\theta_{V_i}, \phi_{V_i}), (\theta_{L_i}, \phi_{L_i}) >, i = 1, 2$, be two viewing/lighting settings. The distance metric is defined to be

$$dist(C_1, C_2) = \sqrt{\|V_1 - V_2\|^2 + \lambda \|L_1 - L_2\|^2} \tag{6}$$

where λ is the relative weight. A large λ value places more emphasis on the lighting condition. We measure the distance using the polar and azimuth angle of the viewing and lighting directions.

For materials with the isotropy we defined in Section 3, the azimuth angle is not important, but the difference between the viewing and the lighting azimuth angles is. Therefore, a more complicated distance metric is adopted for isotropic materials:

$$dist_{iso}(C_1, C_2) = \min_r \{dist(C_1, C_2(r)), dist(\widehat{C_1}, C_2(r))\} \tag{7}$$

where $C(r)$ is a rotation of C by the angle r around the normal of the surface, and $\widehat{C}$ is the reflected version of C about the lighting direction of C. This definition of distance between two viewing/lighting settings for isotropic materials enables us to make use of the images for isotropic materials in the CUReT database.

In the following discussion, we assume orthographic projection, and that the parallax introduced by the height field on the material surface is minimal. If the viewing/lighting setting of the "nearest" reference image is not the same as that of the synthesized texture, it needs to be morphed. Two important factors must be considered: the foreshortening effect caused by the tilt angle of the viewing direction, and the azimuth of the lighting direction. The first one affects the aspect ratio of mesostructure details, and the second one gives rise to lighting effects such as highlights and shadow patterns.

Our algorithm can be summarized in three steps (Fig. 6).

- First, we back-project the sample image (I_0) onto the surface plane to obtain an intermediate image (I_1), according to the camera parameters used to capture the image.

- Second, we rotate the projected image (I_1) around the surface normal such that the azimuth of its original lighting direction coincides with the azimuth of the desired lighting direction of the synthesized texture. We can perform this transformation because we assume that the mesostructure distribution of the geometry is isotropic. The resulting image is called I_2.

- Third, the final reference image (I_3) is obtained by reprojecting I_2, the rotated version of the back-projected image I_1, onto the target view to maintain correct foreshortening.

6.3 Using multiple reference images

In the above synthesis algorithm, only one closest reference image is used for each template image. Better results can be obtained by using more nearby reference images. And the respective results are weighted and averaged to obtain a final synthesized image. We use distance-based weights. Let C_T be the viewing/lighting setting of the synthesized image, and I_{R_i} be the reference images with viewing/lighting setting C_{R_i} , $i = 1, 2, ..., M$. The weight for each reference image I_{R_i} is set up as:

$$\frac{\exp(-\sigma \cdot dist(C_T, C_{R_i}))}{\sum_{k=1}^{M} \exp(-\sigma \cdot dist(C_T, C_{R_k}))} \tag{8}$$

where σ is a constant coefficient, such that the weight is close to 1 for the nearest reference image, and almost 0 for the furthest one in the set of chosen nearby images. The scheme enables a smooth transition when the furthest image is removed from the set and a new reference image is added. It is an interpolation scheme for irregularly scattered data, so it works well even when the input image collection does not uniformly sample the viewing and lighting directions. For sample images with regularly distributed lighting and viewing directions, quadrilinear interpolation in the 4D space of viewing/lighting settings would be a more appropriate choice.

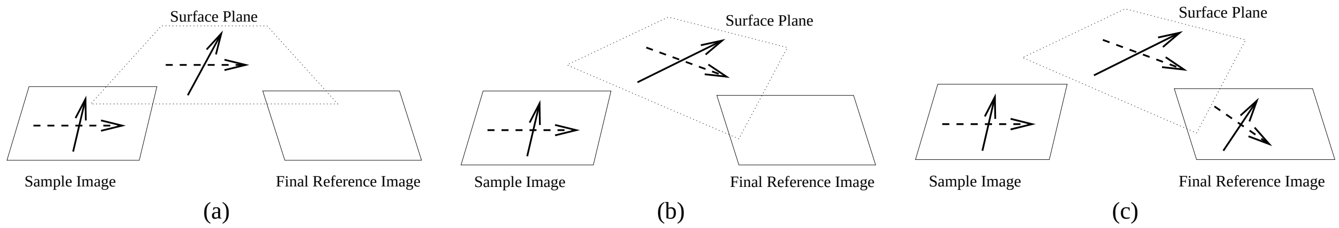


Figure 6: Three steps to generate a reference image from a sample image with variations in lighting and viewing directions: (a) back-project a sample image onto the surface plane; (b) rotate the surface plane to account for change in lighting direction; (c) re-project the rotated surface plane onto the target image plane with different viewing direction.

6.4 Compression

Compression is needed to reduce the amount of disk space for a discrete BTF. However, it is not the focus of this paper since there are many existing mature techniques. For example, synthesized BTFs can be compressed using the clustering technique in [22]. It is also quite straightforward to extend the compression schemes for surface light fields in [28, 39] to work for BTFs.

7 Results

We have successfully tested our algorithms on a few materials from the CUReT database, including rough plastic, plaster, pebbles and terrycloth. For each material, we recovered a 200x200 patch of its height field from four images using our revised height-from-shading algorithm. The four images are chosen to have different viewing/lighting settings and a relatively small number of shadowed pixels. The recovered height fields are then used to synthesize novel height fields at 512x512 resolution, which is much larger than the size of the recovered patches. The BTF synthesis algorithm can run on both recovered and synthesized height fields and generate complete BTFs from all possible viewing and lighting directions. Our program is able to synthesize a 256x256 BTF image in five seconds on a Pentium III 800MHz processor and generate all the images for a 5x12x5x12 grid in the 4D viewing/lighting space in 5 hours. As a result, the materials can be illuminated from all directions and viewed from all directions. Since BTFs belong to lighting-independent surface appearance properties, BTF mapped objects can be easily rendered together with other objects under novel illumination in a ray-tracing or global illumination system.

7.1 Comparison with Ground Truth

Fig. 7 compares three types of materials between the synthesized BTF images and corresponding real reference images. Three pairs of comparisons are shown for each material. Each of the pairs has a distinct viewing/lighting setting. The synthesized images were generated from synthesized height fields. Note that there is a different reference image for each synthesized image. The reference images for the same material are not registered with one another. So they may have different underlying height fields. Nonetheless, the synthesized images can be perceived to have consistent underlying height fields.

7.2 Example images of a synthesized BTF

Fig. 8 shows a plate of synthesized BTF images for rough plastic. The images cover a wide range of lighting directions. The azimuth angle of the lighting direction varies between -90 and 90 degrees. Its tilt angle varies between 45 and 75 degrees. Each column of images have the same viewing direction, but different lighting directions. The left column has a tilt angle of 25 degrees for the viewing direction, and the right column 45 degrees. Each row of images have the same lighting direction, but different viewing directions. The 1st, 3rd, 5th and 7th rows have a tilt angle of 45 degrees for the lighting direction, and the remaining have a tilt angle of 75 degrees. The azimuth angle for the lighting direction increases every other row from -90 to 90 degrees from top to bottom. The appearance of the material varies from image to image because of different locations of highlights and shadows as well as different intensity levels.

7.3 BTF mapping

BTFs can be easily mapped onto objects whose surfaces are parameterized on a rectangular region since texture coordinates for BTF mapping can be set up in the same way as regular 2D texture mapping. Locally shading BTF mapped surfaces from a point light source (instead of a parallel light source) can be carried out as follows. Given the pair of viewing and lighting directions at a certain point on the surface, we can find the corresponding BTF image. From the texture coordinates of that point, we can figure out which pixel value in the found BTF image should be used as the reflectance value for the point. To exploit existing texture mapping functions from graphics libraries such as OpenGL and RenderMan, we need to explicitly extract a 2D texture map for each surface using the above procedure. Note that the extracted texture map is only correct regarding the given viewpoint and light source position. Therefore, we need a distinct texture map for each light source. And the texture maps need to be updated from frame to frame on the fly for an animation with a changing viewpoint or moving objects.

BTF mapping can be implemented as a shader and integrated into any ray-tracing software. We have implemented a shader for BTF mapping in RenderMan BMRT to accumulate the contribution from multiple light source dependent texture maps. A comparison between bump mapping and BTF mapping is shown in Fig. 9. We can see that BTF mapping can deliver more prominent shadowing, occlusion and foreshortening effects as well as spatially varying reflectance. Therefore, the bumps in the BTF mapped image look more protruding and realistic. A scene with multiple objects rendered from RenderMan using ray-tracing is shown in Fig. 10. Some of the objects are BTF mapped. From these examples, we can see that BTF mapping can be considered as a basic rendering function to improve surface appearance.

8 Conclusions and Future Work

In the paper, we presented a novel approach to synthetically generate bidirectional texture functions. Our approach consists of three steps. First, it recovers the approximate 3D geometry of surface details using a shape-from-shading approach. Then, it generates a novel version of the geometric details that has the same statistical properties as the sample surface with a non-parametric sampling method. Finally, it exploits an appearance preserving procedure to synthesize novel images for the recovered or synthesized geometric details under various viewing/lighting settings, which then define a novel BTF. Our experimental results demonstrate that our approach generates BTFs effectively and efficiently.

There are some limitations of our approach in recovering mesostructure details. For example, we impose regularization terms for both geometry and reflectance. Although they have been made spatially-adaptive to account for discontinuities, it is still difficult to recover geometry for natural objects such as grass and straw. And the algorithm for recovering height fields needs a dominant local diffuse component. We would like to address these problems in our future work. Another future direction is to recover both the height field and point-wise non-diffuse reflectance functions simultaneously so that we could generate a BTF with all visual effects without texture synthesis. Note that the global BRDF of a material viewed from distance is most likely more complicated than the local point-wise BRDFs on the material surface to account for interactions among local geometric features. To enforce the global BRDF of a material, we can scale the average intensities of the synthesized BTF images according to the global BRDF. Obviously, mapping BTFs onto arbitrary free-form objects [29] would be a desirable operation.

Acknowledgments

The authors wish to thank Xin Tong for helpful discussion on BTF synthesis, Johnny Chang for implementing the RenderMan BMRT shader for BTF mapping, Yingqing Xu for discussion on patch-based texture synthesis, and the anonymous reviewers for their valuable comments. Timely proofreading by Steve Lin is greatly appreciated. This research was supported in part by the Faculty Startup Funds from University of Illinois at Urbana-Champaign.

References

[1] E.H. Adelson and J.R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20. MIT Press, Cambridge, Mass., 1991.

[2] J.F. Blinn. Models of light reflection for computer synthesized pictures. In *Computer Graphics, SIGGRAPH'77, Vol.11*, pages 192–198, 1977.

[3] J.F. Blinn. Simulation of wrinkled surfaces. In *SIGGRAPH'78*, pages 286–292, 1978.

[4] J. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proc. of Siggraph*, pages 361–368, 1997.

[5] M.J. Brooks and B.K.P. Horn. Shape and source from shading. In *Proc. Intl. Joint Conf. Art. Intell.*, pages 932–936, 1988.

[6] E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. PhD thesis, Univ. of Utah, 1974. Report UTEC-CSc-74-133.

[7] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, 1999.

[8] K.J. Dana and S.K. Nayar. Histogram model for 3d textures. In *Proc. of IEEE Conf. on Comp. Vision and Patt. Recog.*, 1998.

[9] K.J. Dana and S.K. Nayar. Correlation model for 3d textures. In *Intl. Conf. Computer Vision*, 1999.

[10] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Proceedings of SIGGRAPH*, pages 145–156, 2000.

[11] P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH '96*, pages 11–20, 1996.

[12] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *Intl. Conf. Computer Vision*, 1999.

[13] O. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, Cambridge, Massachusetts, 1993.

[14] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen. The lumigraph. In *Computer Graphics Proceedings, Annual Conference Series*, pages 43–54, 1996.

[15] F.R. Hampel, P.J. Rousseeuw, E.M. Ronchetti, and W.A. Stahel. *Robust Statistics*. John Wiley & Sons, New York, 1986.

[16] C. Harris and M.J. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–152, 1988.

[17] D.J. Heeger and J.R. Bergen. Pyramid-based texture analysis/synthesis. In *Proc. of SIGGRAPH*, pages 229–238, 1995.

[18] W. Heidrich, K. Daubert, J. Kautz, and H.-P. Seidel. Illuminating micro geometry based on precomputed visibility. In *SIGGRAPH'2000*, pages 455–464, 2000.

[19] B.K.P. Horn and M.J. Brooks. The variational approach to shape from shading. *Computer Vision, Graphics & Image Processing*, 33:174–208, 1986.

[20] J.J. Koenderink and A.J. van Doorn. Illuminance texture due to surface mesostructure. *J. Opt. Soc. Am.A*, 13(3):452–463, 1996.

[21] Y.G. Leclerc and A.F. Bobick. The direct computation of height from shading. In *Proc. of IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 552–558, 1991.

[22] T. Leung and J. Malik. Recognizing surfaces using three dimensional textons. In *Intl. Conf. Computer Vision*, 1999.

[23] M. Levoy and P. Hanrahan. Light field rendering. In *Computer Graphics Proceedings, Annual Conference Series*, pages 31–42, 1996.

[24] S. R. Marschner, S. H. Westin, E. P. F. Lafortune, K. E. Torrance, and D. P. Greenberg. Image-based brdf measurement including human skin. In *10th Eurographics Workshop on Rendering*, pages 139–152, 1999.

[25] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In *Computer Graphics Proceedings, Annual Conference Series*, pages 39–46, 1995.

[26] D.M. Mount. *ANN Programming Manual*. Dept. Comput. Sci., Univ. of Maryland, College Park, Maryland, 1998.

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

[28] K. Nishino, Y. Sato, and K. Ikeuchi. Eigen-texture method: appearance compression based on 3d model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pages 618–624, 1999.

[29] E. Praun, A. Finkelstein, and H. Hoppe. Lapped textures. In *Siggraph'00*, pages 465–470, 2000.

[30] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge Univ. Press, New York, 1988.

[31] H. Rushmeier, G. Taubin, and A. Gue'ziec. Applying shape from lighting variation to bump map capture. In *Proceedings of the Eighth Eurographics Workshop on Rendering*, pages 35–44, 1997.

[32] Y. Sato, M.D. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. In *Computer Graphics Proceedings, Annual Conference Series*, pages 379–388, 1997.

[33] S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. of IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 1067–1073, 1997.

[34] J. Shade, S. Gortler, L.-W. He, and R. Szeliski. Layered depth images. In *Proc. of SIGGRAPH*, pages 231–242, 1998.

[35] H.-Y. Shum and L.-W. He. Rendering with concentric mosaics. In *Proc. of SIGGRAPH*, pages 299–306, 1999.

[36] E. Simoncelli and J. Portilla. Texture characterization via joint statistics of wavelet coefficient magnitudes. In *Fifth Intl. Conf. on Image Processing, Vol.1*, pages 62–66, 1998.

[37] R. Szeliski and H. Shum. Creating full view panoramic mosaics and environment maps. In *Computer Graphics Proceedings, Annual Conference Series*, pages 251–258, 1997.

[38] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of Siggraph*, pages 479–488, 2000.

[39] D.N. Wood, D.I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle. Surface light fields for 3d photography. In *SIGGRAPH'00*, pages pp.287–296, 2000.

[40] R.J. Woodham. Photometric method for determining surface orientation from multiple images. In B.K.P. Horn and M.J. Brooks, editors, *Shape from Shading*, pages 513–532. MIT Press, 1989.

[41] Y. Xu, B. Guo, and H.-Y. Shum. Chaos mosaic: Fast and memory efficient texture synthesis. Technical Report MSR-TR-2000-32, Microsoft Research, 2000.

[42] Y. Yu. *Modeling and Editing Real Scenes with Image-Based Techniques*. PhD thesis, Computer Science Division, UC Berkeley, 2000.

[43] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *Proceedings of SIGGRAPH*, pages 215–224, 1999.

[44] Y. Yu and J. Malik. Recovering photometric properties of architectural scenes from photographs. In *Proceedings of SIGGRAPH*, pages 207–217, 1998.

[45] S. Zhu, Y. Wu, and D. Mumford. Filters, random fields and maximum entropy (frame)-towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998.

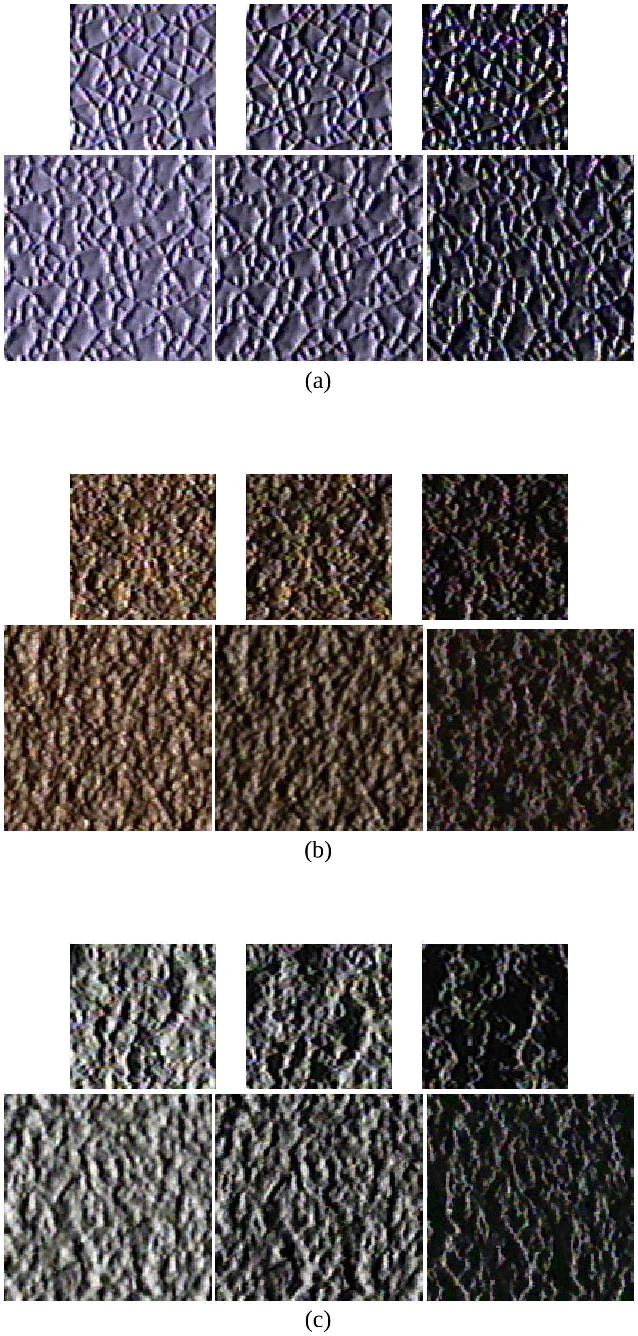


Figure 7: A comparison on three different materials between synthesized BTF images(large ones) and their real reference images(small ones). Three pairs of comparisons are shown for each material. The reference images are on top of their corresponding synthesized images. (a) Rough plastic, (b) pebbles, (c) plaster. The synthesized images were generated from synthesized height fields. Note that there is a reference image for each synthesized one. The reference images for the same material are not registered to one another. They may have different underlying height fields. Nonetheless, different synthesized images for the same material have consistent underlying height fields.

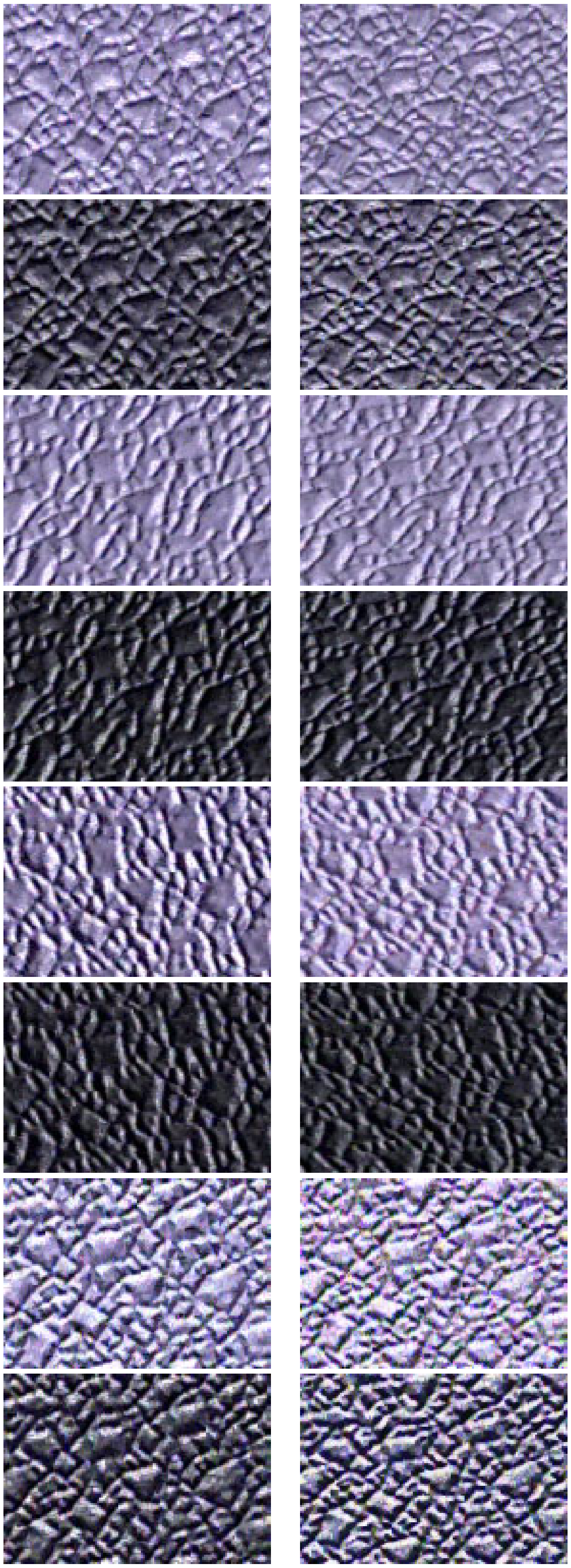


Figure 8: A plate of synthesized BTF images for rough plastic. The images cover a wide range of lighting directions. The azimuth angle varies in a range between -90 and 90 degrees, and the tilt angle varies between 45 and 75 degrees.

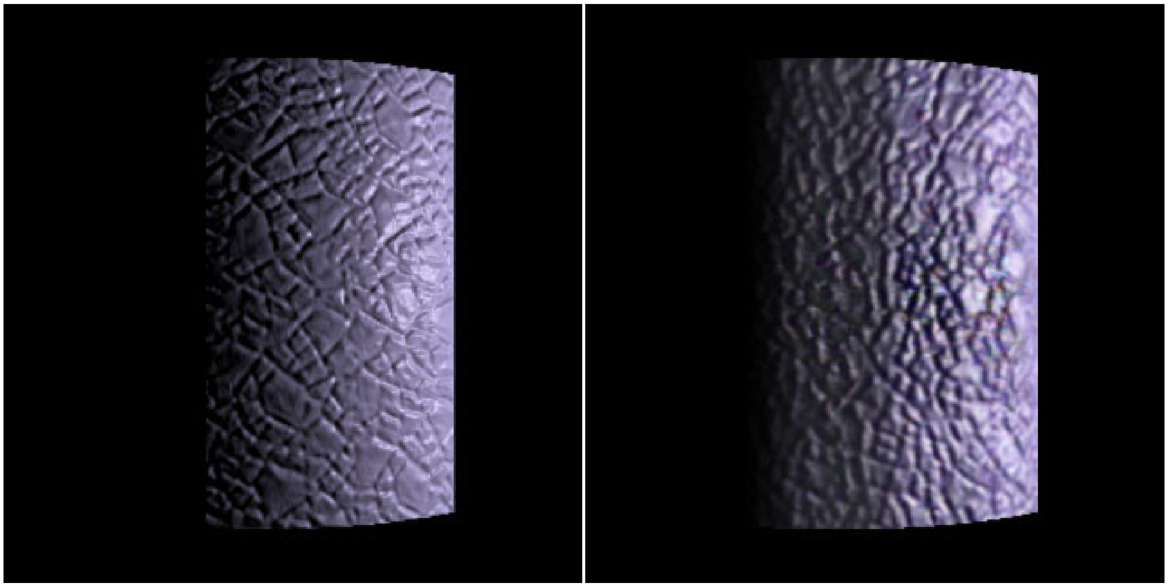


Figure 9: A comparison between bump mapping and BTF mapping. The left image shows a cylindrical surface with bump mapping under the illumination of a point light source from the right hand side. The right image shows the same surface with BTF mapping with the same viewing/lighting setting. We can see BTF mapping has more prominent shadowing, occlusion and foreshortening effects as well as spatially varying reflectance. Therefore, the bumps in the right image look more protruding and realistic.

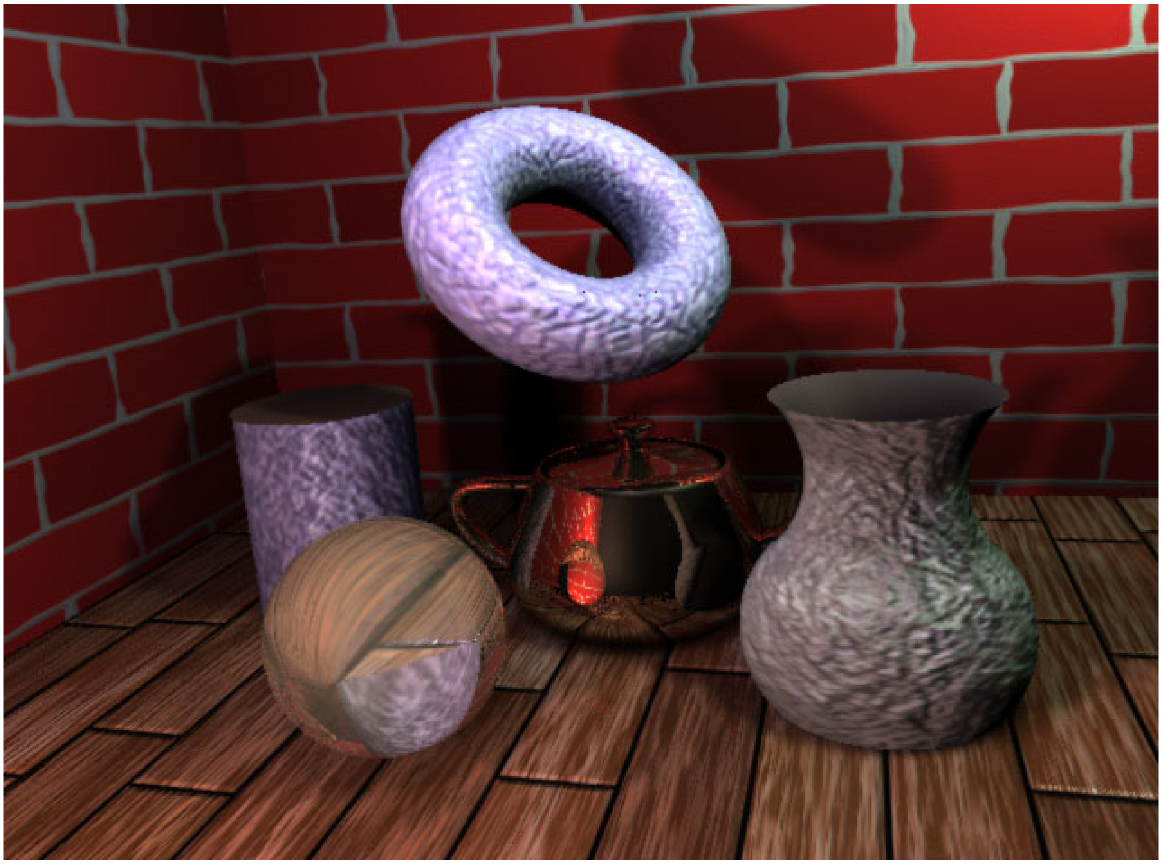


Figure 10: A scene with multiple objects rendered from RenderMan. The torus is mapped with a BTF for rough plastic. The vase is mapped with a BTF for plaster. And the cylinder is mapped with a BTF for terrycloth. We can see small scale shadows among bumps on BTF mapped objects as well as large scale shadows from ray-tracing. The bottom part of the torus has some reddish color coming from interreflection among the objects. The walls are texture mapped. The floor is bump mapped. The teapot has a metallic BRDF and the sphere is half transparent. The BTF mapped objects look more realistic than the texture or bump mapped surfaces.

SIGGRAPH 2002 Course: Acquiring Material Models Using Inverse Rendering

Image-based measurement of translucent materials

Steve Marschner

April 8, 2002

1 The diffuse BSSRDF

The Bidirectional Surface Scattering Reflectance Distribution Function, or BSSRDF, is a generalization of the more commonly used Bidirectional Reflectance Distribution Function, or BRDF, that drops the assumption that the point from which the light reflects is the same as the point where it hits the surface [5]. This allows for materials that are not opaque, through which light will travel some distance below the surface before it comes back out.

As shown in the slides, the full BSSRDF has two variables to describe the offset between the entry and exit points, in addition to the BRDF's four angular variables. If the material is isotropic, we really only need one variable, describing the distance between the two points. If we further assume that the surface behaves like a diffuse reflector overall, as we generally expect that surfaces with a lot of multiple scattering will, we can also do away with the angular variables, leaving just a function R_d that describes the diffuse reflectance per unit area as a function of distance from the incident point. We call this the "diffuse BSSRDF" and it describes the diffuse part of the behavior of many materials fairly well.

This diffuse BSSRDF is the function I will discuss modeling and measuring in this section of the course.

2 The diffuse BSSRDF of a translucent material

This section summarizes the model presented by Jensen et al. at SIGGRAPH 2001 [4]. To derive a BSSRDF for a class of translucent materials, we start off with some assumptions:

The material below the surface acts as a homogeneous scattering medium. This means that its behavior can be described by an absorption coefficient σ_a, a scattering coefficient σ_s, and a phase function $p(\alpha)$. The two coefficients describe how quickly light is attenuated by being absorbed in the medium and by being scattered, or redistributed into other directions. Of the light that gets scattered, the phase function tells us how it will be distributed to different directions. The phase function is often summarized by its average cosine g, which is positive, zero, or negative for forward scattering, isotropic, or backward scattering media respectively.

The medium is highly forward scattering. That is, $\sigma_s \gg \sigma_a$ and $g > 0$. The important implication of this is that by the time the light has propagated far enough to get back out of the material it has scattered enough times that its direction is random. Once this has happened, the phase function is not important in further scattering events, since an isotropic light distribution will still be isotropic after being scattered by any phase function. It turns out that under this condition of isotropic radiation the effect of any phase function is the same as isotropic scattering with the *reduced scattering coefficient* $\sigma'_s = \sigma_s(1 - g)$.

The mean free path is small on the scale of the objects we are rendering. This assumption ensures that the isotropy mentioned above takes over fast enough that we don't need to worry about the small region around the incident point where it does not hold. Another way of thinking of this is that the material should be translucent, not transparent; light gets through the material but we can't see anything through it.

Under these assumptions the transport equation reduces to a diffusion equation. For radiation coming from a point source in the medium there is an analytical solution for the radially symmetric light distribution that is produced by that source. As Eason et al. [2] showed, the effects of the interface at the surface of the medium (which reflects some of the light back down into the material) can be accounted for by a second virtual source, with a negative intensity, positioned outside the medium. Farrell et al. [3] showed that a point source is a sufficiently accurate approximation for a narrow parallel beam entering the material, so our model uses the expression for a point source. For further details the reader is referred to the paper [4].

3 Measuring translucent materials

Given a sample of a translucent material, we can measure its scattering behavior by just illuminating a very small spot, then taking a photograph to capture the distribution of light that reflects from the surface. Comparing this data to the model serves two purposes: we can find out whether the model actually describes the material we're looking at, and if it does we can find out the parameters that we should use to render it.

In order to capture the wide dynamic range necessary to make these measurements, we use a modified version of Debevec and Malik's method [1]. The modification is to the curve that assigns a weight based on the pixel value: to avoid the bias introduced by a value-dependent weight, we simply weight each image by its exposure time. We use the weight curve only to reject saturated pixels. The slides show an example of a HDR image and a few images from the exposure series used to compute it.

These images show the two-dimensional distribution across the surface, but since the material is homogeneous it has to be radially symmetric. So we just extract one slice through the image, passing through the brightest point in the center of the distribution. The pixel values in this slice are proportional to the function R_d (we use a picture of a ruler lying on the surface to calibrate the physical size of the pixels on the suface). The slides show examples of the resulting data for several different materials.

For reasons explained below, we also make a second measurement with the sample replaced by a standard white diffuse sample that has known reflectance.

4 Fitting the model to the measurements

The fitting process is a simple nonlinear least squares computation (minimizing relative error because the HDR imaging process produces approximately uniform relative accuracy), but there are some subtleties that are worth mentioning.

In the simplest case, we use just the measurement of the material in question, fitting the model as a function of its two parameters, σ_s' and σ_a, and an unknown scale factor k (which is needed because the camera and light source are not calibrated in any absolute units). (The model does also depend weakly on the medium's refractive index, which for most water-based materials we assume to be that of water. When a more specific value is available we use that instead.)

Using this three-parameter approach leads to very good fits between the model and the data, but the process is ill-conditioned and the resulting scattering coefficients are unstable under small changes to the data or the specifics of the fit. Worse yet, if we compute the total reflectance (albedo) of the material and compare it to the albedo we observed (using the white sample as a reference), the match is often quite bad.

To improve the conditioning, we can use the reference measurement to establish the scale factor. This improves the conditioning considerably, making the coefficients more stalbe. (It also reduces the closeness of the fit, since the fitting procedure no longer has the freedom to adjust the units.) It also brings the computed albedos closer to the measured values.

To get the most reliable results (but at the expense of the independent check provided by the albedo measurement), the albedo can be included in the fit as an additional measurement. This is the approach we used to compute the coefficients that are reported in the paper [4].

References

[1] P. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH 1997*, August 1997.

[2] G. Eason, A. Veitch, R. Nisbet, and F. Turnbull. The theory of the backscattering of light by blood. *J. Physics*, 11:1463–1479, 1978.

[3] T. J. Farell, M. S. Patterson, and B. Wilson. A diffusion theory model of spatially resolved, steady-state diffuse reflectance for the noninvasive determination of tissue optical properties in vivo. *Med. Phys.*, 19:879–888, 1992.

[4] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. In *Proceedings of SIGGRAPH 2001*, August 2001.

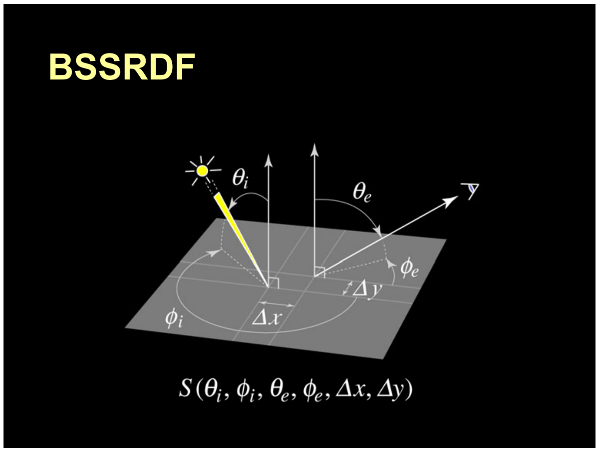
[5] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Geometric considerations and nomenclature for reflectance. Monograph 161, National Bureau of Standards (US), October 1977.

SAN ANTONIO SIGGRAPH 2002

Course 39: Acquiring Material Models Using Inverse Rendering

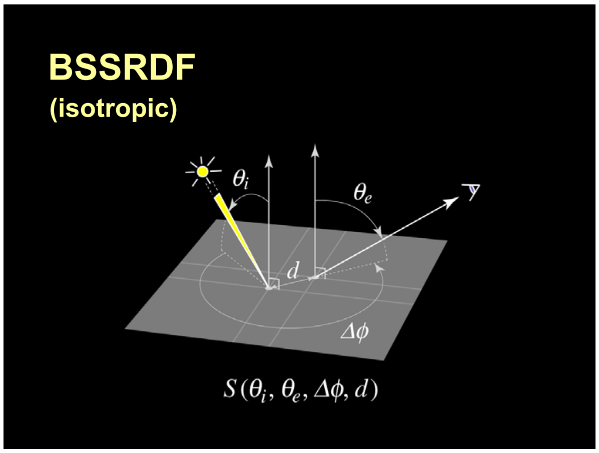
Image-based measurement of translucent materials

Steve Marschner



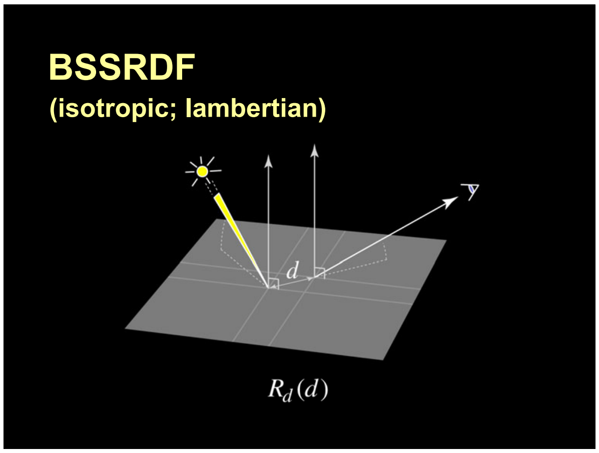
BSSRDF

$S(\theta_i, \phi_i, \theta_e, \phi_e, \Delta x, \Delta y)$



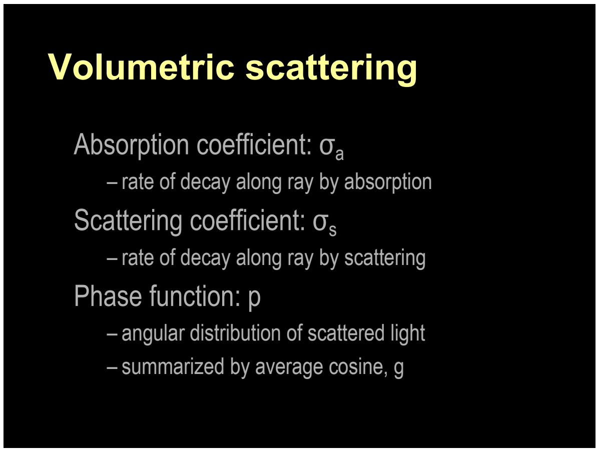
BSSRDF
(isotropic)

$S(\theta_i, \theta_e, \Delta\phi, d)$



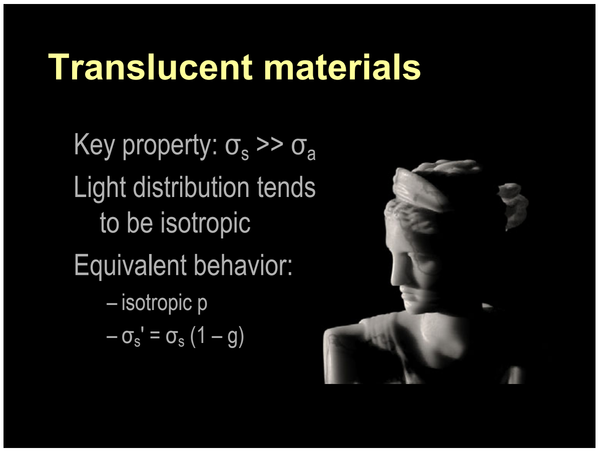
BSSRDF
(isotropic; lambertian)

$R_d(d)$



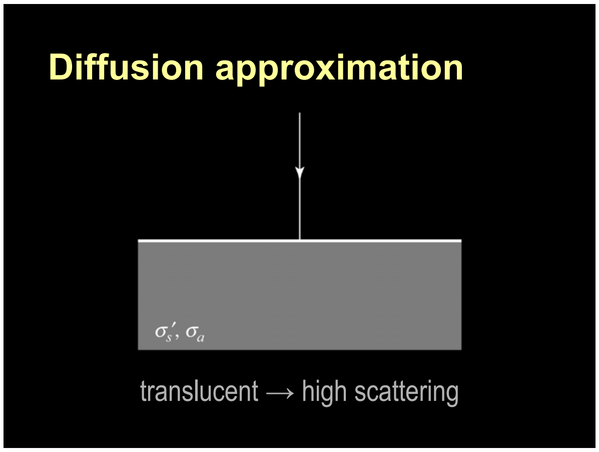
Volumetric scattering

- Absorption coefficient: σ_a
  - rate of decay along ray by absorption
- Scattering coefficient: σ_s
  - rate of decay along ray by scattering
- Phase function: p
  - angular distribution of scattered light
  - summarized by average cosine, g



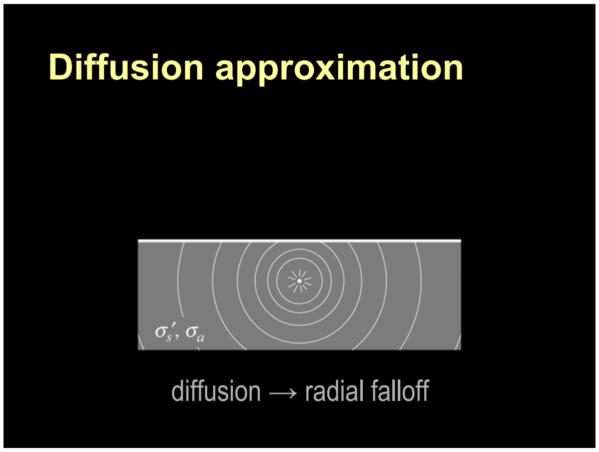
Translucent materials

- Key property: $\sigma_s >> \sigma_a$
- Light distribution tends to be isotropic
- Equivalent behavior:
  - isotropic p
  - $\sigma_s' = \sigma_s (1 – g)$



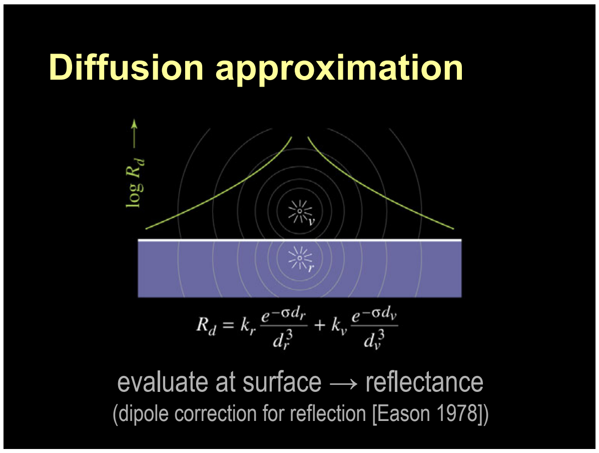
Diffusion approximation

translucent ⟶ high scattering



Diffusion approximation

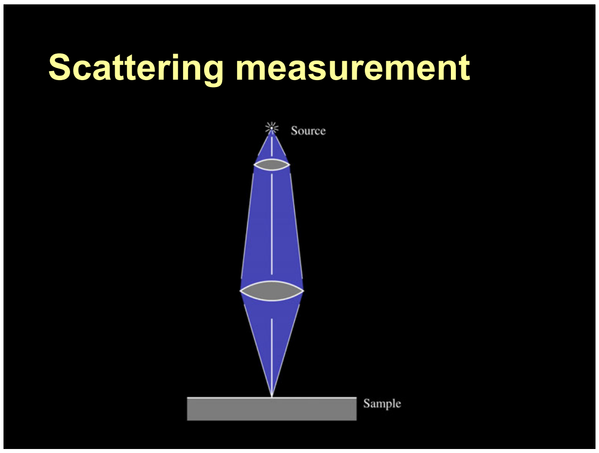
diffusion ⟶ radial falloff



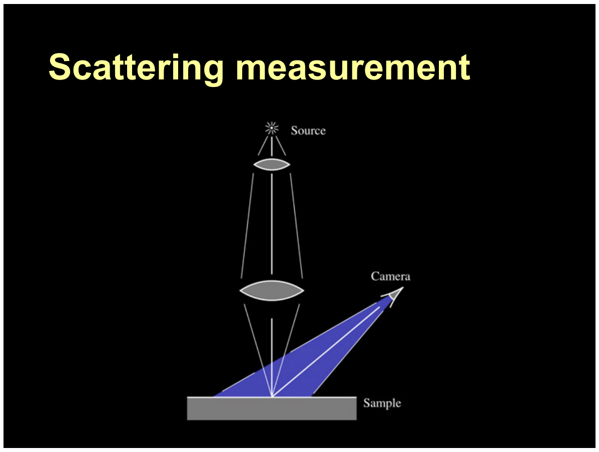
Diffusion approximation

$$R_d = k_r \frac{e^{-\sigma d_r}}{d_r^3} + k_v \frac{e^{-\sigma d_v}}{d_v^3}$$

evaluate at surface ⟶ reflectance
(dipole correction for reflection [Eason 1978])

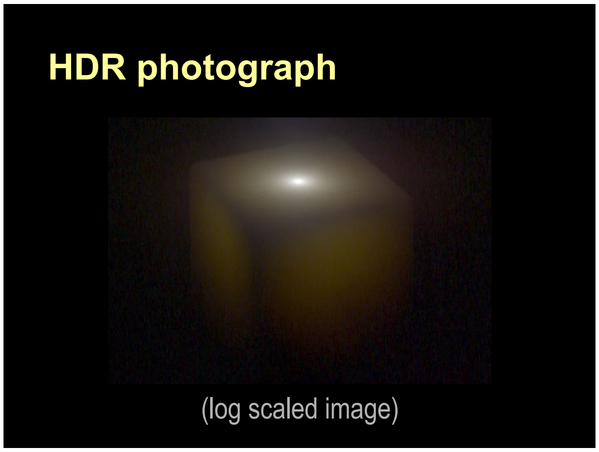


Scattering measurement



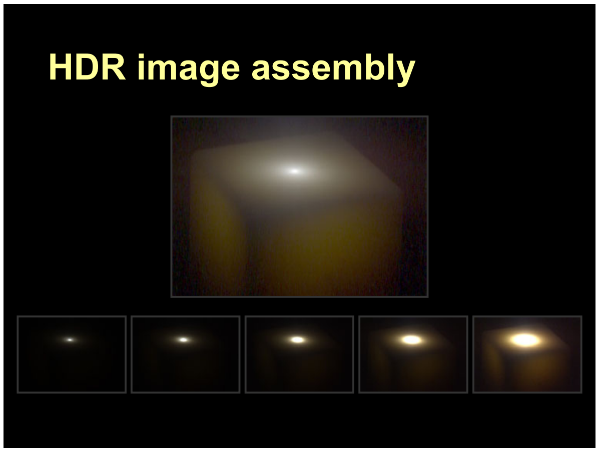
Scattering measurement

Marble sample

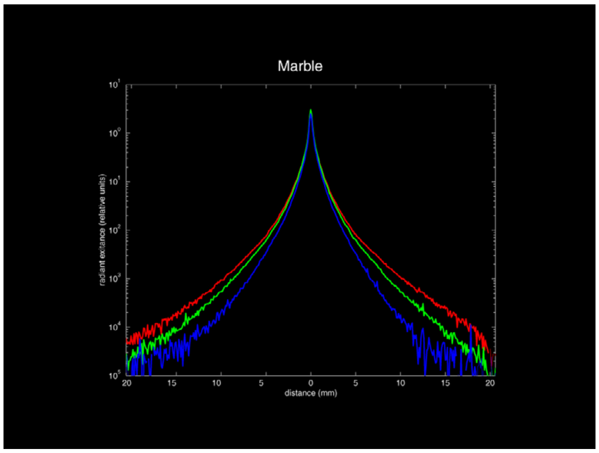


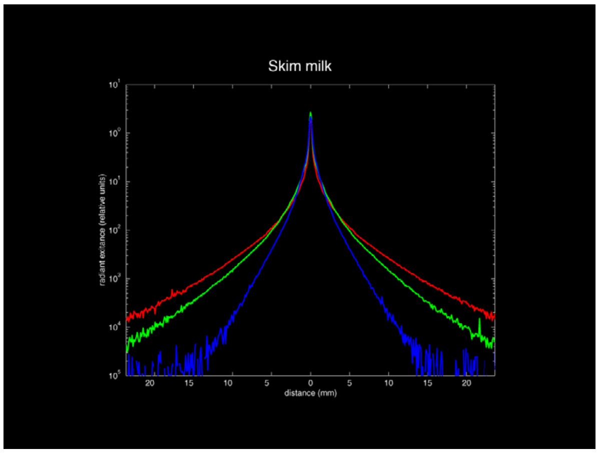
HDR photograph

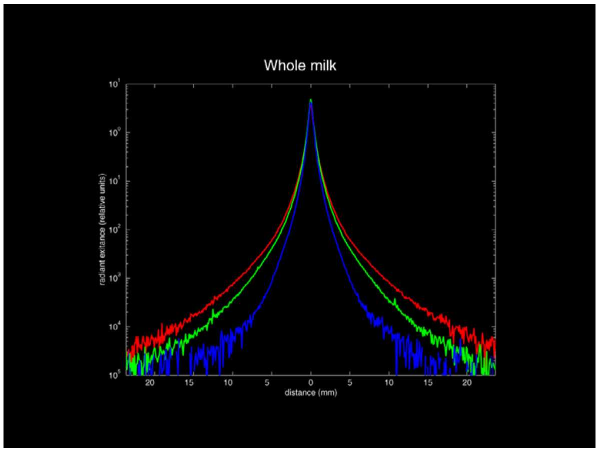
(log scaled image)

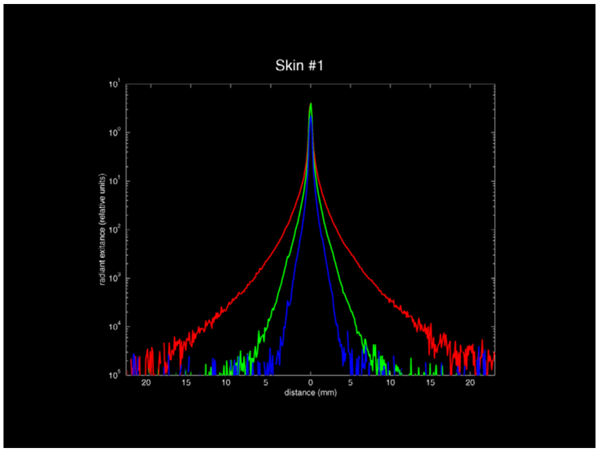


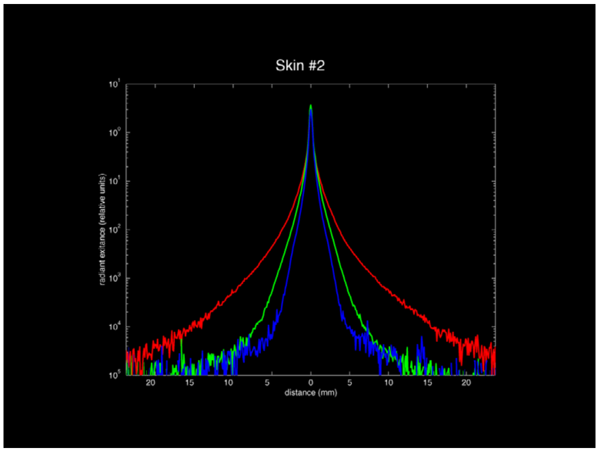
HDR image assembly










Skin #2

Data fitting

Two purposes

- validate model
- measure parameters

Fit minimizes relative error

Principal variables: σ_s', σ_a

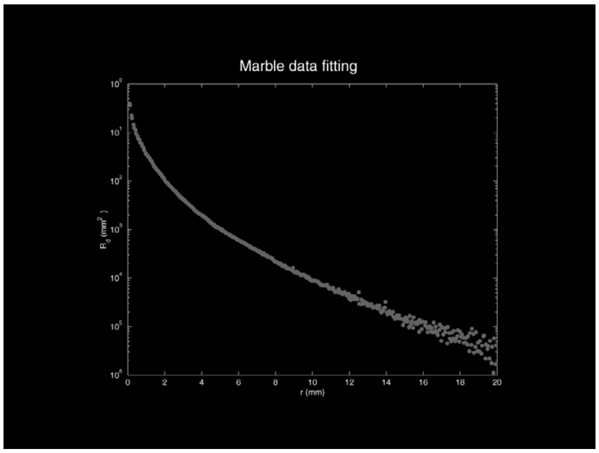
Relative measurements → also fit scale

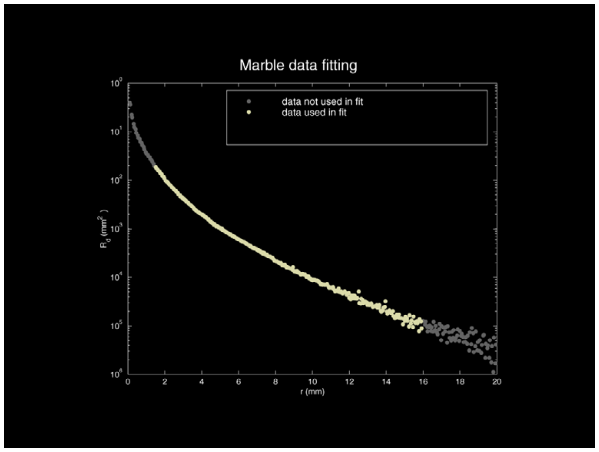
Improving conditioning

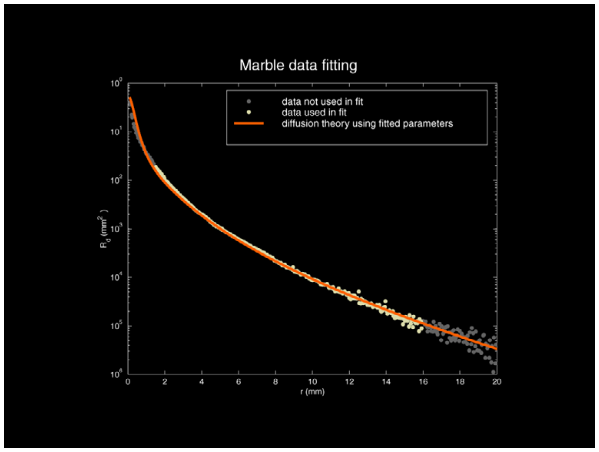
With scale factor conditioning is bad

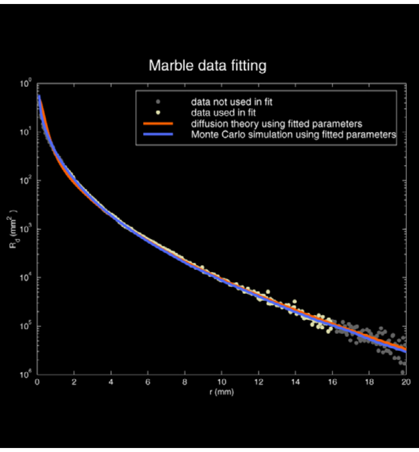
Measure diffuse white reference sample

- gives incident flux (eliminates scale factor)
- indirectly gives sample reflectance


Marble data fitting


Marble data fitting


Marble data fitting


Marble data fitting

Conclusion

- Simple way to measure reflectance spatially
- Extension: non-uniform materials
- Extension: angular measurements