KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT COMPUTERWETENSCHAPPEN
AFDELING INFORMATICA
Celestijnenlaan 200A – B-3001 Leuven – België

SAMPLING REFLECTANCE FUNCTIONS FOR IMAGE-BASED RELIGHTING

Promotor:
Prof. Dr. ir. Philip Dutré

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Pieter PEERS

August 2006

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT COMPUTERWETENSCHAPPEN
AFDELING INFORMATICA
Celestijnenlaan 200A – B-3001 Leuven – België

SAMPLING REFLECTANCE FUNCTIONS FOR IMAGE-BASED RELIGHTING

Committee:
Prof. Dr. ir. Willy Sansen, chairman
Prof. Dr. ir. Philip Dutré, advisor
Prof. Dr. ir. Yves D. Willems
Prof. Dr. ir. Marc Van Barel
Prof. Dr. ir. Luc Van Gool
Prof. Dr. Philippe Bekaert (University of Hasselt)
Prof. Dr. Paul E. Debevec (University of Southern California)
Prof. Dr. Céline Loscos (University College London)

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Pieter PEERS

U.D.C. 681.3*I3

August 2006

D/2006/7515/69

ISBN 90-5682-735-9

Sampling Reflectance Functions for Image-based Relighting

Pieter Peers

Department of Computer Science,
Katholieke Universiteit Leuven

ABSTRACT

A popular research topic in computer graphics is the acquisition of the appearance of real-world objects. An important aspect of the appearance of an object is how it reacts to incident illumination.

The goal of image-based relighting is to visualize real objects under novel incident illumination, and this without explicit knowledge of the object's geometry or its material properties. The appearance of an object under incident illumination is characterized by its *reflectance field*. To acquire such a reflectance field, a series of photographs, from a fixed viewpoint, of the object is recorded under different controlled illumination conditions. Applying different illumination conditions to an object is mathematically equivalent to sampling its reflectance field.

In the first part of this work, the physical and practical constraints imposed on image-based relighting are studied, and a mathematical framework is derived that encodes these constraints. This framework allows to describe, study, and compare existing relighting techniques, and allows to develop new, more efficient, methods.

In the second part, the implications of sampling the reflectance field are studied in detail, and several reconstruction techniques are presented to enhance the visual and numerical quality of the relit results. Additionally, it is shown that an equivalent downsampling operator can be defined on the incident illumination that yields identical results with less computations as the advanced reflectance field reconstruction techniques.

Two novel acquisition methods are presented in the third part of this dissertation. These methods differ from other acquisition methods in that they sample a wavelet represented reflectance field directly. The first of these methods samples the reflectance field selectively in the wavelet domain by progressively emitting selected wavelet basis illumination conditions. A feedback loop is used during acquisition to determine what part of the wavelet transformed reflectance field is worthwhile to sample in greater detail. The second technique also samples the reflectance field selectively in the wavelet domain, but decouples the acquisition process and the sampling of the wavelet represented reflectance field. To achieve this, the object is observed under a fixed number of wavelet noise illumination conditions. Afterwards, a progressive algorithm is used to determine the reflectance function of each pixel separately using the observed wavelet noise responses of this pixel.

All of the previous techniques are restricted to 2*D* incident light fields. In the final part of this work, a novel method is presented that is able to relight objects with 4*D* incident illumination. This allows to visualize objects with spatially varying illumination effects such as spotlights effects and partial shadowing.

The research that led to this dissertation has been supported by K.U.Leuven Grant #$OT/01-34$ and K.U.Leuven Grant $GOA/2004/05$.

Acknowledgments

Probably the most difficult part of writing a dissertation is this page. I apologize that this page only contains 15 times the word "*thanks*", which is far less than these people deserve.

First of all, I wish to thank my advisor, prof. Dutré, who's opinion, although not always apparent, I value highly. He has always supported my research for which I am very grateful. I am also eternally grateful for the support and trust prof. Willems invested in me, and for giving up his co-advisor position at the very last moment to enable me to finish this Ph.D. in time.

Many thanks must go to prof. W. Sansen, prof. M. Van Barel, prof. L. Van Gool, prof. Ph. Bekaert, prof. P. Debevec, and prof. C. Loscos for accepting to be part of my evaluation committee. I am extremely honored to have prof. Debevec as a member of my jury, and I am even more honored that I can join his team next month. I would like to thank prof. Loscos for sacrificing some beach time while at EGSR to go over this text with me.

Special thanks go to my current and former colleagues —Philippe Bekaert, Frank Suykens, Vincent Masselus, Karl vom Berge, Frederik Anrys, Bart Adams, Ares Lagae, Olivier Dumont, Muath Sabha, Peter Vangorp, and Toon Lenaerts— for great research discussions, lots of fun, and hours of coffee breaks. I am indebted to prof. Philippe Bekaert for following his instinct when it came to this strange guy who appeared six years ago at his office door looking for a job in the computer graphics group. My deepest respect go to Dr. Frank Suykens, who I consider to be one of my best friends, and who taught me many valuable things about graphics and life in general. I will miss our two-weekly lunches next year. This list would not be complete without thanking Dr. Vincent Masselus, with who I shared an office for 5 years, 2 papers, and enough fun to fill another complete dissertation.

I'd like to thank all the people with who I collaborated during the past few years. Many thanks to Thomas Koninckx and the rest of the vision group for the interesting discussions. I especially would like to thank Wojciech Matusik, who is a great researcher and a cool dude in general. Thanks to Ravi Ramamoorthi, Jason Lawrence, and Szymon Rusinkiewicz for a successful collaboration for this year's SIGGRAPH. Also Akira Kubota deserves a place in this list, for the inspiring discussions.

I cannot thank my family and friends enough who always supported me in pursuing my dreams. Thank you for understanding my (mental and physical) absence during deadline periods. Thank you all.

To thank the following person for her unwaivering support, her love, and for keeping me (in)sane and alive, I would need to write a complete chapter. However, due to time constraints this does seem possible, so I'll do the next best thing and dedicate this dissertation to her, my wife, my love, Saskia.

Heverlee – August 2006.

to Saskia

Contents

Notations and Symbols

Notations

s	scalar value
V	a vector of scalar values of length l (i.e., an $l \times 1$ matrix)
v_i or $(v)_i$	the i-th element of a vector V
$[v_i]_i$	a vector defined by the scalar values v_i with $i \in \{1,\ldots,l\}$
$\mathbf{M}$	a matrix of scalar values of size $l \times k$
$\mathbf{M}_i$	the i-th row of a matrix $\mathbf{M}$. The row is a $1 \times k$ matrix.
$\mathbf{M}_{i:s}$	the i-th row upto the s-th row of a matrix $\mathbf{M}$. The result is a $(s-i+1) \times k$ matrix.
$\mathbf{M}_{\cdot,j}$	the j-th column of a matrix $\mathbf{M}$. The column is an $l \times 1$ matrix.
$\mathbf{M}_{\cdot,j:t}$	the j-th column upto the t-th column of a matrix $\mathbf{M}$. The result is an $l \times (t-j+1)$ matrix.
$\mathbf{m}_{i,j}$ or $(\mathbf{m})_{i,j}$	the matrix element at the j-th column and i-th row.
$[\mathbf{M}_1;\ldots;\mathbf{M}_l]$	the matrix $\mathbf{M}$ defined by the rows $\mathbf{M}_i$. Note that a semicolon ";" is used to seperate different rows.
$[\mathbf{M}_{\cdot,1}\vert\ldots\vert\mathbf{M}_{\cdot,l}]$	the matrix $\mathbf{M}$ defined by the columns $\mathbf{M}_{\cdot,i}$. Note that a vertical line "$\vert$" is used to seperate different columns.
$\mathcal{R}(\cdot)$	an operator on a scalar, vector, matrix, or other operator.
$\mathtt{s}$	a set of length n.
$\mathtt{s}[i]$	the i-th element of a set $\mathtt{s}$.
$\vert\vert V\vert\vert_2$	the length of a vector.
$\tilde{\cdot}$	an approximation of a scalar, vector, matrix, or operator.
$\overline{\mathbf{M}}$	the dual of a matrix $\mathbf{M}$ (i.e., $\mathbf{M}\overline{\mathbf{M}}^T = I$).

Important Symbols

L	incident illumination vector.
C	outgoing illumination vector.
S	self-emitted illumination vector.
$\mathbf{T}$	light transport matrix.
$\mathcal{S}$	serialization operator.
$\mathcal{R}$	resample operator.
$\mathcal{U}$	upsample operator.
$\mathbf{B}$	general basis transformation.
Ψ	wavelet basis transformation.

1 Introduction

The “digital age” is perhaps how contemporary society will be referred to in a few centuries. Since the rise of electronics and computers, the world around us has changed tremendously. Shapes have become richer thanks to CAD; Colors and color combinations are more vibrant thanks to desktop publishing; ... Designing and producing goods, buildings, clothes, ... practically everything, is unthinkable without digital aid.

A very active, and relatively young research domain, is the *digitalization* of the real world. The impact of this digitalization is already changing everyday life. Digital cameras and the omni-present mp3-player are examples of consumer devices aimed at capturing instances, images and music respectively, of the world around us. The instances captured by these devices are not necessarily restricted to instances created with the aid of a digital device, but can also be digitalized snapshots of any real entity. Although these devices are only the beginning, they herald a new and exciting era to come.

This dissertation discusses and contributes to recent advancements in the *digitalization of the appearance of real objects*, or image-based relighting, a young but important research topic within the field of computer graphics.

1.1 Image-based Graphics

A famous computer graphics saying proclaims that *“Reality is 80 million polygons per second”* (Alvy Ray Smith quoted in Rheingold [84], p.168). But I must disagree: when saying that reality is *only* 80 million polygons per second, more than half of reality is omitted: the reflection behavior, textures, illumination, ... The reality of reality is that it is an enormously complex, if not impossible, thing to model.

The proverb that *“a picture says more than a thousand words”* is the essence of image-based graphics, offering an alternative approach to tackle the problem of the abundance of detail in reality. Taking a photograph of a real-world scene or object automatically contains all the details, so why not use this. Depending on the goal, image-based graphics can be subdivided in a number of sub-domains: image-based modeling, image-based rendering, and image-based lighting.

Image-based modeling. The goal of image-based modeling, is to extract all kinds of properties from a set of images of a real-world scene. These properties range from: scene appearance, scene geometry, lighting, reflectance properties, ... Image-based modeling is an extremely large research sub-domain. We will highlight a few methods to give an impression of the diversity and richness of image-based modeling. Providing a complete overview of all image-based modeling methods would fall beyond the scope of this dissertation.

In 1996, Debevec et al. [23] presented a method to model and render architecture from photographs. In this paper, three image-based modeling techniques are presented: photogrammetric modeling, view-dependent texture mapping, and model-based stereo. By providing minimal building blocks, photogrammetric modeling optimizes the user-provided correspondences to obtain a model of the scene. To improve the quality of the visualizations, view-dependent textures are used. Depending on the view-direction, different textures extracted from the source photographs are blended. Finally, model-based stereo can be used to add detail-offsets to the obtained model.

Another example is image-based visual hulls, presented by Matusik et al. [60]. A visual hull is a conservative shell bounding an object, defined by the silhouette information from a series of reference photographs of the object. Assuming a pinhole camera, each silhouette forms a cone with its apex at the camera origin, and spreading out along the silhouette edges. The intersection of these cones forms the visual hull. Instead of computing this intersecting hull explicitly (using expensive CSG operations), image-based visual hulls are created directly in image-space from the reference images. Each pixel in a novel view defines a ray starting at the viewpoint, and going through the pixel. By exploiting the epipolar geometry, these rays can be projected onto the silhouettes in the reference photographs. This allows to determine, on the fly, where the visual hull begins and ends along this ray.

A last example is by Lensch et al. [49], who measured spatially varying reflection properties of an object without making additional assumptions, from a small set of photographs ($10-25$) and a geometric model of the object. A key observation is that man-made objects contain very few different materials. This allows to cluster the different materials. Each cluster contains multiple surface points, and thus more information regarding the reflection properties is available than when a single surface point is used. From the reflection information in a cluster, a Lafortune reflectance model can be fitted.

Image-based rendering. The goal of image-based rendering, is to create realistic novel views of a scene at interactive rates, without performing a full global illumination simulation. The scene properties, such as lighting, geometry, ... are kept constant, except of course the viewpoint.

Image-based rendering techniques can be subdivided in three categories [94]: rendering with no geometry, rendering with implicit geometry, and rendering with explicit geometry. This should not be seen as three discrete categories, but as a continuum. A lot of work has been done in this area, and providing an exhaustive overview falls beyond the scope of this dissertation. We will review for each category one technique. For a thorough overview see [94] and [116].

Light field rendering [50] falls in the first category: rendering with no geometry. Light field rendering can generate novel views of real-world objects from a set of photographs of this object, without using any geometry, or depth information. Key to this technique is the interpretation of the photographs of an object as $2D$ slices of a $4D$ function: the light field. This abstraction reduces the generation of a novel image to a resampling and interpolation problem. Because of this, light field rendering is also a popular research topic in signal processing.

The lumigraph [33], is related to light field rendering [50], but uses an approximative geometry, and therefore belongs to the second category: rendering with implicit geometry. The approximative geometry is used to counter the non-uniform sampling of the $2D$ slices to improve rendering quality.

Layered depth images [92] fall in the last category: rendering with explicit geometry. Layered depth images are a view of an object from a single vantage point, containing multiple pixel and depth values along each line of sight. Layered depth images allow to render the object from novel viewpoints.

Image-based lighting. The goal of image-based lighting, is to create novel realistic images of a scene or object with light obtained from the real world. Image-based lighting is analogous to image-based modeling, except that now illumination is modeled in an image-based fashion.

A popular tool to capture real-world lighting is a light probe: a specular ball that is placed in a scene. By taking a photograph of this light probe, all illumination incident at this point is captured (see [83], chapter 9, for an in depth overview).

Debevec [19] used image-based lighting to render synthetic objects into real scenes. This is also called augmented reality. This method works by dividing the scene into three parts: the distant scene, the local scene, and the synthetic objects. The distant scene is considered not to be influenced by the synthetic objects, and is modeled using image-based lighting. The local scene is image-based modeled, and a global illumination simulation is used to visualize this local scene together with the synthetic objects. The obtained rendered image is merged into a photograph of the scene.

A sub-domain in image-based lighting, is image-based **re**lighting and has as goal to create novel realistic images of a real-world scene, under novel illumination conditions, and this without computing a full global illumination simulation. The scene properties, geometry, reflectance properties, ... are kept constant.

Popular work on image-based relighting was performed by Debevec et al. [20]. By combining different basis images of an object, each acquired under different controlled illumination conditions, a novel image of the object can be created under novel incident illumination. This *relit* image is created by weighting and summing the basis images. The weights are determined from a light probe image of the destination lighting.

Image-based relighting is the main focus of this dissertation, and an in depth introduction into this topic is given in chapter 2.

1.2 State-of-the-art and Goals

Scientific advancement does not happen overnight, it takes a lot of time and effort. When I started my research, the state-of-the-art was different than it is now. At that time, the Light Stage was introduced at SIGGRAPH 2000 in a seminal paper by Debevec et al. [20]. It was a huge leap forward in terms of relighting quality and practical efficiency. It is currently still the most widely used method for image-based relighting, and has been used in numerous Hollywood movies. The previous year, at SIGGRAPH ’99 in another seminal work by Zongker et al. [119], environment matting was presented. This work initiated a renewed interest in interests in matting techniques. Although not apparent at that time, it also has some relations with image-based relighting.

The following observations can be made about the state-of-the-art at that time:

- Not only did the acquisition setup for image-based relighting (using a Light Stage) and for environment matting (using a CRT monitor) differ significantly, they also had a different theoretical and historical background. Although some “combinations” of environment matting and image-based relighting had been attempted (among others in [20] and [63]), it was not apparent at that time that both actually solve a similar problem.

- Most relighting and environment matting techniques were limited to a subset of material types that they could handle correctly (i.e., diffuse to glossy for [20], and glossy to specular for [12, 119]). No technique existed that could handle all kinds of materials without imposing any additional restrictions.

- All image-based relighting techniques used a brute-force approach to capture and relight real-world objects.

- Incident illumination and reflectance functions were restricted to $2D$ instead of the complete $4D$ incident domain.

These observations resulted in the following research goals :

- A unifying mathematical framework is pursued that covers both environment matting as well as image-based relighting. This framework should be extensible to cover both 2*D* approximated incident illumination and the complete 4*D* incident illumination. Furthermore, this framework should be flexible enough such that new techniques can be easily incorporated.

- To develop an image-based relighting technique that can capture scenes and objects without placing any restrictions on the types of materials present.

- A compact representation of the reflection properties of the acquired objects is desired.

- Sub-linear acquisition complexity with respect to the acquisition resolution, in order to minimize acquisition duration.

- To develop a method to capture and relight scenes with 4*D* incident light fields.

All these goals are met in this dissertation, although not all in a single unifying solution.

1.3 Overview

This dissertation is organized as follows:

Chapter 2 begins with an intuitive description of image-based relighting.

An important element in image-based relighting is *incident illumination*, a 4*D* entity. For practical reasons, a 2*D* approximation of incident illumination is preferred. Therefore, a discussion is included on what kinds of approximations can be made in order to reduce the dimensionality of 4*D* incident illumination to a two dimensional one. A commonly used method for capturing a 2*D* incident light field approximation is by means of a light probe. The obtained approximation, also called an environment map, is described and discussed. Next, environment maps are validated against the proposed theoretical 2*D* approximations. To conclude the discussion on incident illumination, methods for the acquisition of the complete 4*D* incident light field are discussed.

Image-based relighting is intuitively introduced through the Light Stage [20]. An overview of an image-based relighting pipeline, using the Light Stage, is explained in detail. Important terms such as *reflectance fields* and *reflectance functions* are illustrated.

Chapter 3. In this chapter, the intuitive notion of image-based relighting is given a theoretical foundation.

First, the physical constraints and conditions to which image-based relighting must adhere are discussed. On these constraints, a mathematical framework is based. Some additional operators are defined that effectively remove the dependency on the dimensionality of the incident illumination. Finally, this framework is illustrated on Dual Photography [91], and on the Light Stage [20].

Chapter 4. The acquired reflectance functions, captured by a Light Stage apparatus, are basically a sampling of real continuous reflectance functions. In this chapter, we explore the implications of Nyquist's sampling theorem on reflectance functions. Next, different reconstruction methods for sampled reflectance functions are compared. Their performance and efficiency with respect to still and animated incident illumination is qualitatively and quantitatively validated.

Finally, using the mathematical framework presented in chapter 3, it is shown that for image-based relighting it is possible to define, for a large class of reconstruction methods (i.e., upsampling operators), an equivalent downsampling operator on the incident illumination. Downsampling the incident illumination offers a significant speed-up in computations compared to upsampling the sampled reflectance functions while maintaining relighting quality.

Chapter 5. High resolution, acquired or upsampled, reflectance functions have huge storage requirements. This chapter studies a particular class of lossy compression methods to reduce the required storage space.

By projecting the reflectance functions in a suitable (mathematical) basis, a sparse representation can be obtained. By further applying non-linear lossy approximations, this can be reduced even further. An advantage of basis projection methods is that they are basically linear operators with respect to light transport. As a consequence, relighting computations can be performed directly in the projected space, and the (linear) image-based relighting process does not incur a time-penalty to compute a relit image.

Chapter 6. Compression by means of basis projection helps to reduce the storage requirements, but it still require a complete sampling of the reflectance field during acquisition. Sampling reflectance fields at a high resolution poses two practical problems. First, it is practically difficult to extend a Light Stage such that it can sample these high resolutions (i.e., 10^6 samples). Second, taking a sample requires a fixed amount of time. The use of high-speed cameras can reduce the time required to take a single sample, but at extremely high resolutions, even these high-speed cameras cannot sample the full reflectance field in an acceptable time.

Based on environment matting, two new acquisition techniques are presented that use a CRT monitor instead of a Light Stage to acquire a reflectance field. Sub-linear acquisition (in terms of resolution) is attained by using algorithms based on the compression techniques of the previous chapter.

The first method explictly samples the reflectance field expressed in the wavelet domain. To reduce the number of samples which need to be taken, a progressive algorithm is used that, depending on the current approximation, decides what part of the reflectance field is important enough to sample in greater detail.

The second method, uses wavelet noise illumination patterns. Wavelet noise is a dense slice through the whole wavelet domain. This allows to implicitly sample the wavelet domain after acquisition from the observed responses under wavelet noise illumination. By acquiring a number of wavelet noise responses, an approximation of the reflectance functions can be computed for each pixel separately.

Chapter 7. Previous acquisition methods all capture $4D$ approximations of the $6D$ reflectance field (from a fixed vantage point). In this chapter a novel acquisition device, based on the Light Stage, capable of acquiring the $6D$ reflectance field is presented. Acquiring the complete $6D$ reflectance field in a naive manner is too time consuming. Therefore, an acquisition acceleration method is presented that can capture a $6D$ reflectance field in $\mathcal{O}(n^3)$ time complexity (instead of $\mathcal{O}(n^4)$).

Chapter 8. This chapter concludes this dissertation with a summary, a discussion of the contributions, and avenues for future research in image-based relighting.

2

Image-based Relighting

This chapter introduces the basic concepts of image-based relighting in an intuitive manner. The goal is to give the reader a basic understanding of the concepts which will be discussed more rigorously in chapter 3.

2.1 Introduction

Image-based relighting tries to solve the problem of how an arbitrary object or subject's appearance changes when the illumination is altered. A key difference between image-based relighting and classic modeling techniques is that image-based relighting does not require the shape of the object to be measured or approximated, but decouples the relighting process from the underlying geometry, and instead directly uses photographs of the object to create relit images of this object.

In a classic modeling approach, the appearance of an object is completely defined by describing each of the specific characteristics of the object by a model. The geometry of an object is usually described by a mesh of polygons or a cloud of points [77], and the material (reflection) properties are characterized by bidirectional reflectance distribution functions [68], which are denoted by the acronym BRDF. A BRDF describes how a ray of light is scattered when hitting the surface of the object at some point (figure 2.1). A BRDF is a 4*D* function, i.e., a 2*D* incoming light direction and a 2*D* outgoing light direction. This 4*D* function is usually compactly represented by an analytical model. A number of analytical models for BRDFs exist [2, 8, 14, 37, 41, 46, 72, 78, 80, 88, 103, 107], each tuned to describe a small subset of material types, and each depending on a small number of user-controlled parameters.

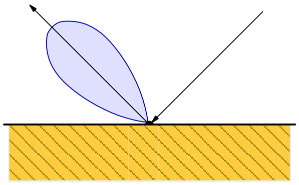


Figure 2.1: A BRDF describes how a ray of light is directionally scattered when hitting the surface at a specific point. The directional scattering is iconically depicted by the blue lobe.

To visualize the model under some illumination condition, a full scale global illumination simulation needs to be computed [29]. There are a number of problems with this approach:

- **Micro-geometric details.** Many objects have sub-geometrical details, such as dirt, wrinkles, rust, etc... Each of these sub-geometrical details can alter the reflection properties locally in an unpredictable way. Without these details, a model would look too clean, but modeling all these details exactly is in most cases impractical.

- **Ill-defined geometry.** Not all shapes can be easily represented by a mesh of polygons. Objects covered by hair or fur are notoriously difficult to represent using a mesh of polygons or point clouds.

- **BRDF parameters.** BRDFs are analytical models each describing a small subset of material types, e.g., plastic. Each BRDF model has a number of parameters, representing the properties of the material that influence the appearance. When modeling a real-world material, these parameters need to be tuned such that the approximation error between the model and reality is minimized. Tuning these parameters is usually a complex matter, requiring many measurements of the reflection properties of the material with varying viewpoints and illumination directions. Furthermore, the algorithms used to tune these parameters are very sensitive to the initial starting conditions, and do not always converge to a globally optimal solution.

- **BRDF selection.** The representational power of a model is limited. E.g., a BRDF model for describing metallic materials cannot accurately represent plastic materials. Selecting a suitable model is not a trivial problem. Materials for which no model exists, or materials which behave abnormally, cannot be represented accurately, and need to be approximated by the "best" resembling model.

- **Degree of realism depends on the method of visualization.** A full global illumination simulation is necessary to create a visualization of the object under novel incident illumination. Such a global illumination simulation is a lengthy process and the efficiency depends greatly on the algorithm used.

Image-based techniques overcome these problems by taking a different approach. Instead of representing the complete object by compact analytical models, image-based techniques start from the observation that a photograph contains all the details of an object seen from a single viewpoint at a specific moment in time discretized at a specific resolution. By directly using the information contained in multiple photographs of the object, the need for an analytical model is avoided, while all the details are implicitly present. In the specific case of image-based relighting, a series of photographs of the object are recorded, each under different illumination conditions. From the information present in the different photographs, the response of each pixel to incoming illumination is inferred, independently of the responses of other pixels. By directly using the information contained in the acquired photographs, the relighting process is decoupled from the underlying geometry of the measured object. An additional advantage of image-based relighting techniques is that no global illumination simulation is required, because they use the information directly from the photographs which already contain the full global illumination solution.

In this thesis only the relighting of objects seen from a fixed viewpoint is considered. A different branch of computer graphics, called image-based rendering, actually tries to solve the complementary problem: how does an object or subject's appearance change when the viewpoint is altered, keeping all other parameters, such as incident illumination fixed.

The remainder of this chapter is organized as follows; In section 2.2 the acquisition and representation of incident illumination is discussed. Based on this captured incident illumination, image-based relighting is introduced in an intuitive manner (section 2.3). Next, the concepts of reflectance fields and reflectance functions are introduced in section 2.4. We conclude this chapter in section 2.5.

2.2 Incident Illumination

As mentioned in the previous section, the goal of image-based relighting is to visualize an object under novel incident illumination. In this regard, it is important to first discuss ways to represent and capture incident illumination.

To define incident illumination, we first have to consider to what this illumination is incident to. In the case of image-based relighting, the illumination is defined incident to the object being relit. This implies, that for each point on the surface of the object, illumination coming from all directions needs to be known. However, this is not a flexible definition, because we do not want to tie the acquisition and representation of incident illumination to a single specific object. Instead, a more general, geometry-independent definition is required.

An important principle, widely applied in computer graphics, is that radiance traveling along a direction (in vacuum), does not change in intensity or color. This principle allows to decouple incident illumination from a specific object, assuming that the object is surrounded by free space. In practice, a bounding volume is considered in which the object, to which the incident illumination will be applied, fit. If a complete description of the incident illumination is known for each point on this bounding surface, the incoming illumination is known for each direction and point on the object's surface (see figure 2.2). Incident illumination is thus a $4D$ function, dependent on position ($2D$), and direction ($2D$). This $4D$ incident illumination function is also called a *"4D incident light field"* [58].

Capturing and representing a $4D$ function is a huge undertaking, and the practical problems are significant. Therefore an idealized reduced $2D$ approximation of the $4D$

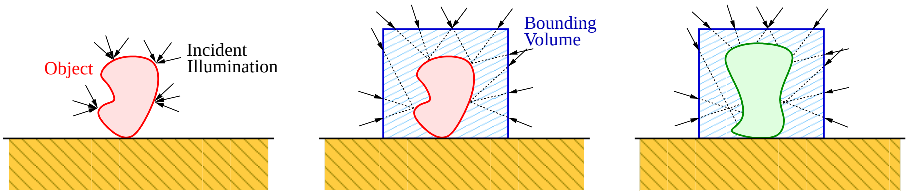


Figure 2.2: Decoupling incident illumination from a specific object. Left: the incident illumination on an object. Middle: incident illumination on a bounding volume decoupled from the object's geometry. Right: The incident illumination on the bounding volume can be applied to any object fitting in this bounding volume.

incident light field is mostly used. We can distinguish two idealized 2*D* approximations:

- **Directional incident illumination,** spatially constant. This implies that the incident illumination only varies directionally, and that at each point on the bounding volume the same incident illumination occurs (figure 2.3.a). Another way of looking at this is that the incident illumination comes from far away (infinity), and in effect reduces the object and bounding volume to a single point.

- **Spatial incident illumination,** directionally constant. This is basically the inverse of the previous. In this case we assume that the illumination varies spatially, but that im each point it is angularly constant or diffuse (figure 2.3.b).

The accuracy of both approximations depends completely on the characteristics of the incident illumination. If the bounding volume is small with respect to the surrounding environment, then the directional approach will be a better approximation of the incident illumination. On the other hand, if the bounding volume is a close match to the surrounding geometry, then the spatial incident illumination approximation would probably be a better match. Both cases are illustrated in figure 2.4.

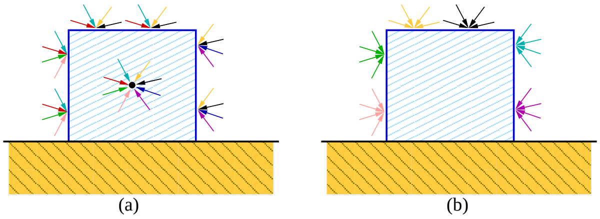


Figure 2.3: Different 2*D* approximations of the full 4*D* incident light field. Directional incident illumination (a): corresponding directions at different points have the same color and intensity. Spatial incident illumination (b): each direction for a specific point on the bounding surface has the same color and intensity.

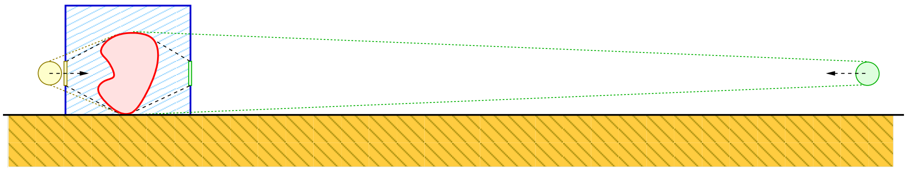


Figure 2.4: The accuracy of the 2*D* approximation depends on the type of approximation, and the nature of the incident illumination. A determining factor is the solid angle from which incident illumination hits the bounding volume. This depends on the distance between the origin of the incident illumination and the bounding volume. Two different cases of incident illumination are depicted; the yellow light source, located closely to the bounding volume, is best approximated by a spatial approximation, while the green light source, position further away from the bounding volume, is better approximated by the directional approximation.

Next, the acquisition of so called *environment maps* is discussed (subsection 2.2.1), followed by an illustration of the relation between environment maps and the 2*D* approximations of incident illumination (subsection 2.2.2). Finally, a method is detailed to capture 4*D* incident illumination (subsection 2.2.3).

2.2.1 Environment Maps

An environment map is a spherical map, representing illumination incident at (approximately) a single point in space. The most common ways to capture an environment map are:

- **Fish-eye lens.** A digital camera, equipped with a fish-eye lens, is positioned at the location for which an environment map is desired. Because a fish-eye lens only deflects a part of the surrounding sphere of directions (i.e., a hemisphere) to the camera sensor, multiple photographs have to be recorded, each with a different orientation, to capture the whole sphere of incident illumination (figure 2.5.a).

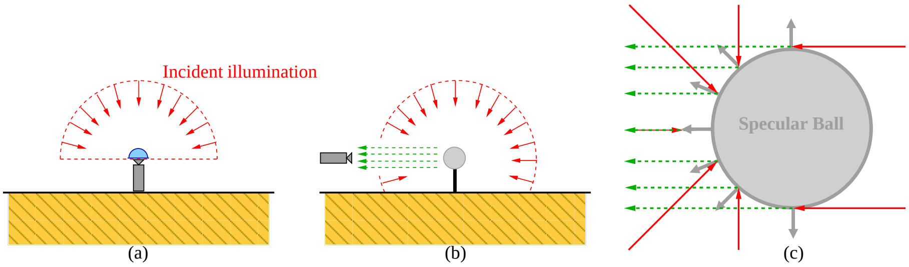


Figure 2.5: Capturing an environment map. Figure (a) depicts the capture of an environment map using a fish-eye lens. The environment lighting is only partially captured. In figure (b) the acquisition of an environment map using a specular ball is depicted. Figure (c) depicts a close-up of the specular ball and how incoming illumination is reflected towards the camera.

- **Light probe.** A light probe is in essence a specular ball. By positioning the light probe at the location for which an environment map needs to be captured, and taking a photograph of this light probe with a zoom lens, an environment map can be obtained (figure 2.5.b). Figure 2.5.c depicts how illumination coming from the whole environment to the light probe is reflected towards the camera.

Each of the two methods to capture an environment map has its strengths. The acquisition of an environment map using a fish-eye lens has the advantage that the camera is never visible in the environment map, but has as disadvantage that the lens distortions and chromatic aberration near the edge of the lens can be significant. Furthermore, fish-eye lenses are expensive and only capture an incoming hemisphere instead of a full sphere of incoming directions. Capturing an environment map using a light probe has the advantage that almost the whole environment map is captured in a single shot. A disadvantage is that the camera is visible in the light probe. The camera can be removed from the environment map by taking two photographs of the light probe from different positions and merging the parts of the environment maps in which the camera is not visible. Furthermore, the density of reflected rays is much greater near the edge of the specular ball, resulting in aliasing artefacts. The cost of a specular ball, however, is minimal, making a light probe an ideal tool for capturing environment maps.

Both techniques capture an environment map by taking a photograph. In order to create physically plausible results using the acquired environment maps, it is important to express the intensity of each pixel in a photograph in terms of a physical unit. Incident illumination on a small area (pixel), also denoted by “irradiance”, is expressed in $\frac{W}{m^2}$. A consumer digital camera converts the measured irradiance to non-linear pixel intensity values. Debevec and Malik [22] detail a method for undoing this non-linear transformation, and transform it back to (relative) irradiance values. The process of undoing this non-linear transformation is called *“High dynamic range photography”* and the resulting photograph is called a *high dynamic range photograph*, or, *HDR photograph* for short. In the remainder of this thesis we will always use HDR photography when acquiring data. A detailed overview of HDR photography can be found in [83], chapter 4.

To store the environment map in an acquisition device independent manner, a reparameterization of the acquired data is required. The reparameterization will maximize the amount of significant pixels (i.e., pixels corresponding to points on the light probe), and try to minimize distortions. Different spherical and hemispherical parameterization schemes can be found in [56]. In the remainder of this work, we will implicitly assume that environment maps are parameterized in a latitude-longitude parameterization, unless noted differently.

In figure 2.6, an example acquired by a light probe is shown. Both the captured photograph of the light probe (left) and a latitude-longitude reparameterization are shown.

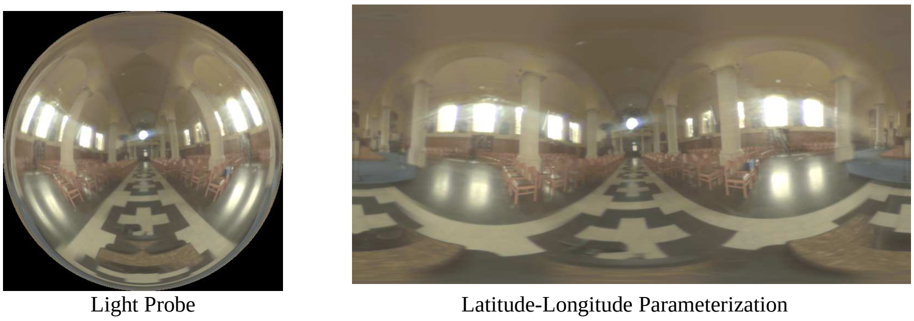


Figure 2.6: An example of a photograph of a light probe, acquired in a church, and a latitude-longitude reparameterization of the light probe image. (Photographs: courtesy of Vincent Masselus)

2.2.2 Relation to $2D$ Incident Illumination

The question remaining, is how an environment map relates to the $2D$ idealized approximations of incident illumination. It is clear that an environment map is a $2D$ representation of incident illumination, but it is unlikely that it exactly corresponds to one of the idealized $2D$ approximations: a directional or a spatial approximation.

Both techniques for acquiring environment maps deflect illumination incident at a single point towards the camera sensor. This implies that an environment map does not encode any spatial information. Furthermore, it also implies that an environment map does not encode any directional information of the emitted illumination, since for each point of the environment, at most a single direction of incoming illumination is deflected towards the camera. As expected, an environment map is not a directional $2D$ approximation, nor is it a positional $2D$ approximation. See figure 2.7 for an iconic illustration.

Actually, an environment map is a $2D$ slice of the $4D$ incident light field. Depend-

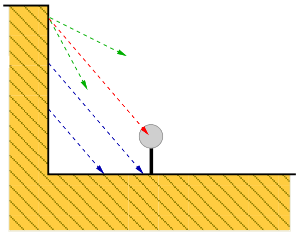


Figure 2.7: An environment map captures incident illumination at a single point (red), it does not capture directional outgoing illumination from the environment, but only captures a single direction for each point in the environment (green). Nor does it capture spatial, directionally constant information (blue).

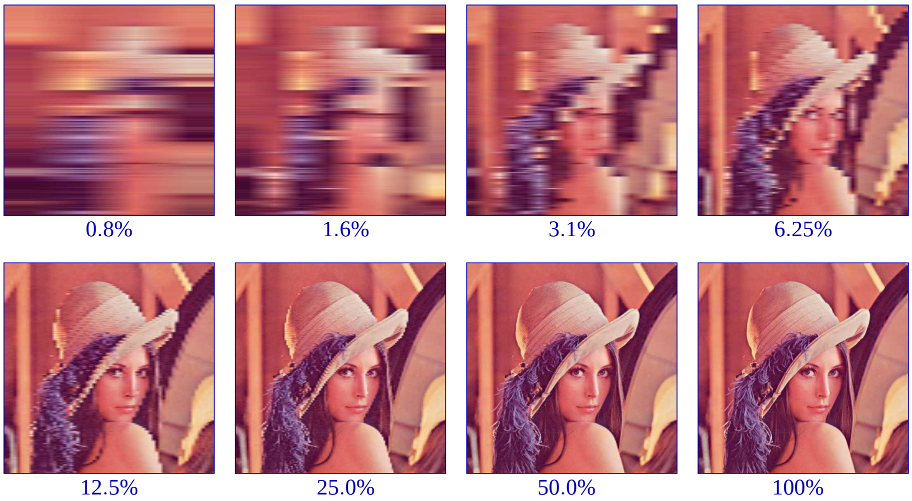


Figure 2.8: The effect of taking 1*D* slices (i.e., vertical lines) of a 2*D* function, illustrated on the well-known Lena image. Areas with low horizontal complexity need to be less densely sampled to be faithfully approximated (e.g., shoulder), than areas with abrupt changes (e.g., near edges). The reconstructions are computed by linearly interpolating horizontally between neighboring slices. The percentages indicate the sampling density of each image.

ing on the situation, it is a good approximation of either a 2*D* directional or spatial approximation. As shown in figure 2.4, an environment map is a good approximation of directional incident illumination if the surrounding environment is large in comparison to the bounding volume. Likewise, if the surrounding environment is of similar size as the bounding volume, then a spatial approximation and an environment map are a close match (assuming that the environment emits or reflects illumination diffusely).

We will show in the next section, that both 2*D* approximations are handled identically by image-based relighting. This implies that there is not really a difference between both approximations visually, but rather in the interpretation of the approximation. The accuracy of the 2*D* approximation is more related to the size of the bounding volume, and the geometry and reflection properties of the surrounding environment. In the remainder of this thesis, when using a 2*D* approximation of the 4*D* incident light field, we will omit the interpretation of the approximation.

2.2.3 Acquisition of 4*D* Incident Light Fields

Until now, only the acquisition of environment maps, a 2*D* slice of the 4*D* incident light field, has been discussed. The argument for using a 2*D* approximation is that the full 4*D* incident illumination is huge in size and difficult to capture. However, with the advancement of image-based relighting techniques, the need for acquiring 4*D* incident light fields is growing. In this section a brief overview is given of the technique

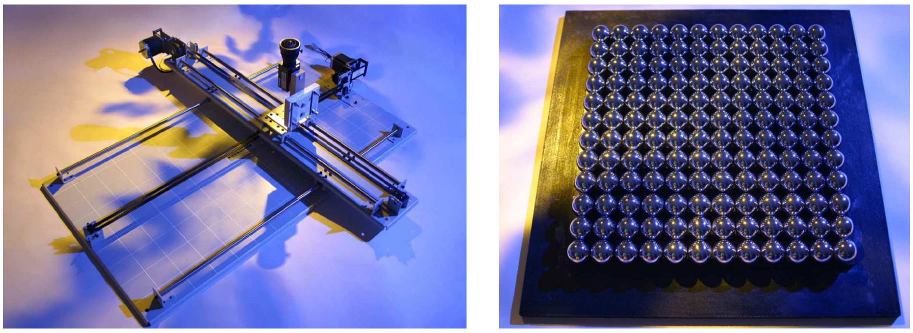


Figure 2.9: Two devices for the acquisition of 4*D* incident light fields. Left: a camera with a fish-eye lens mounted on an *xy*-table. Right: an array of specular balls. Both devices can capture the 4*D* incident illumination on a plane. (Photographs: Courtesy of Unger et al. [104])

of Unger et al. [104] for the acquisition of 4*D* incident illumination.

In Unger et al. [104], two methods for acquiring 4*D* incident light fields are outlined. At the core of both methods is the observation that an environment map is a 2*D* slice of the 4*D* incident light field. By capturing multiple 2*D* slices, a reconstruction of the full 4*D* function can be made. What happens mathematically, is that samples (i.e., 2*D* slices) are taken from a 4*D* function. The sampling frequency and the frequency response of the sampled dimensions of the 4*D* incident illumination function determine the accuracy of the approximation. Figure 2.8 illustrates this principle on a 2*D* image from which 1*D* slices are extracted. Increasing the sampling density increases the reconstruction quality. Note, that the quality of the approximation converges faster in areas where the original image varies little (horizontally).

Both methods for capturing 4*D* incident illumination take one of the devices for acquiring an environment map as a basis. Figure 2.9 shows the devices developed by Unger et al. to capture 4*D* incident light fields. The left device in figure 2.9 shows a camera with fish-eye lens mounted on an *xy*-table. The *xy*-table enables to accurately position the camera at different positions on a plane. For each position a high dynamic range photograph (2*D* slice) can be recorded. This approach can sample the 4*D* incident illumination very densely, at the cost of a long acquisition cycle. On the right in figure 2.9, an array of specular balls is shown. A single photograph of this array gives a coarse sampling of the 4*D* incident illumination. In effect the resolution of the 2*D* slices is traded-off with the sampling resolution. It is important to realize, that although the latter method captures a low resolution sampling of the 4*D* incident light field, it is captured in a single photograph, as opposed to the first method, which needs to record multiple photographs.

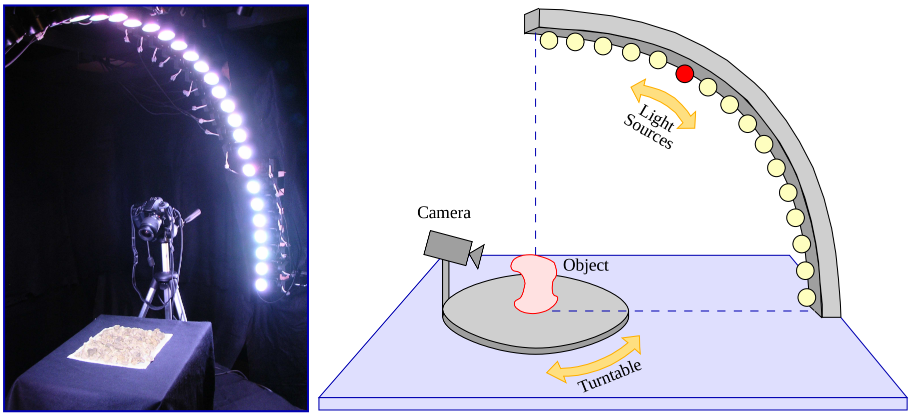


Figure 2.10: A photograph and schematic illustration of a Light Stage device (as presented in [59]). A number of computer controlled light sources are mounted on a quarter arc, while the object and camera are placed (relatively fixed) on a turntable. By switching individual lights on and off, a latitudinal movement of light sources is attained, while rotating the turntable provides longitudinal movement.

2.3 Intuitive Description of Image-based Relighting

In the previous section the acquisition, approximation, and representation of incident illumination is discussed. In this section image-based relighting is introduced in an intuitive manner. This description is based on the Light Stage [20], an apparatus for efficiently capturing the reflectance field of an object (i.e., the way the appearance of an object reacts to incident illumination).

To capture how an object reacts to incident illumination, a series of HDR photographs of the object, each under different lighting conditions, are recorded, while maintaining a fixed viewpoint. These lighting conditions are generated by a Light Stage device. There are a number of different incarnations of the Light Stage [20, 34, 57, 62, 21, 110, 59, 109]. Although these devices are mechanically very different, the basic idea is the same: to be able to illuminate the object, from different light source positions on a bounding sphere. The main difference between the devices is how they sample light source positions. In this thesis we use a Light Stage device as presented in [59], depicted in figure 2.10. This device consists of a quarter arc on which a number of light sources are mounted (i.e., 20). Each of these light sources can be turned on or off remotely by a computer. A turntable is mounted at the center point of the quarter arc. On this turntable the to-be-captured object and the camera are placed. Different lighting conditions can be obtained in two ways:

- **Latitudinal light source movement.** The repositioning of the light source along the latitudinal direction (i.e., from top to bottom) is achieved by switching

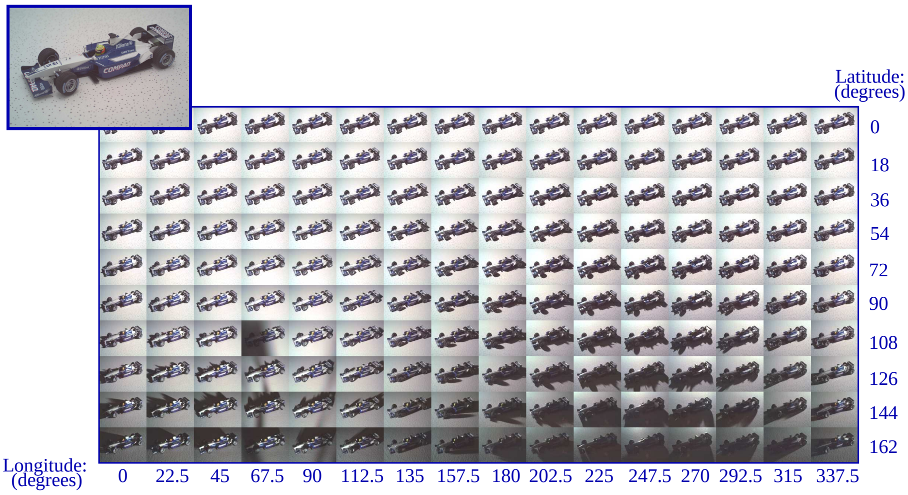


Figure 2.11: A subset of the photographs of a Formula One toy car (shown enlarged in the left-top) acquired with a Light Stage. For each photograph a different light source is turned on. Note that the dark anomaly at $(67.5, 108)$ longitude-latitude is caused by the camera partially occluding the light source for that position.

the correct light sources on or off. No mechanical movement is required, and hence it can be done very rapidly.

- **Longitudinal light source movement.** To move a light source along a longitudinal direction, the turntable is rotated in the opposite direction. Since the camera is mounted on the turntable, the net result is similar to rotating the arc. This requires some mechanical movement, and requires significantly more time than a latitudinal movement.

To minimize mechanical movement, and optimize efficiency, the acquisition scans at each longitude position, all latitude positions first. Similar to this setup is the Light Stage v2 [34], which rotates the arc instead of the object. An advantage of the setup used here, and in [59], is the ease of mechanical construction, and relatively fast acquisition.

Figure 2.11 shows a subset of acquired photographs of a Formula One toy car. Each photograph shows the effect of illuminating the Formula One car from a specific position on the bounding sphere.

Before introducing the general image-based relighting algorithm, a number of specific cases are examined. Suppose that the novel incident illumination, used to create relit images, coincides with one of the sampled light source's position and intensity. In that case, the relit image of the object is simply the acquired photograph of the object lit by that specific light source. In the case of a more complex illumination (e.g., the incident illumination consists of two light sources, coinciding with two different sampled light source positions), the relit image is the sum of the photographs associated with the respective light sources, due to the additivity of light. The additivity of light stems from the superposition principle, which will be discussed in detail in chapter 3. Finally, consider the case in which the incident illumination coincides with a light source at a sampled position, but with a different intensity, e.g., 50% of the acquisition intensity. Since the observed intensity corresponds linearly with the intensity of the light source, a relit image can be created by simply multiplying the acquired photograph corresponding to the respective light source, by the relative ratio of the desired intensity versus the acquired intensity. Note that color can be seen as having, in the case of RGB photographs, three independent intensities, one for each color channel.

Using the additivity and the scaling of the intensity, a general image-based relighting algorithm can be formulated. Assume that the incident illumination is given as a $2D$ approximation in the form of a latitude-longitude environment map. This parameterization is a convenient form, since the light sources are positioned on a latitude-longitude grid on the bounding volume due to the construction of the Light Stage. Note that the resolution of the latitude-longitude environment map, is probably much higher than the sampling rate of the acquisition setup. Therefore, we downsample the environment map to the sampling resolution of the setup, resulting in a one-to-one mapping of intensities in the downsampled environment map and sampled light source positions. However, the intensity values in the downsampled environment map do not take into account that the Light Stage acquisition device samples light source positions more densely near the top than down at the base. This can be corrected by taking into account the Jacobian of the reparameterization from a sphere to the latitude-longitude mapping. Thus, by scaling each photograph with the corresponding pixel intensity in the downsampled environment map times the Jacobian, and adding the scaled photographs, a relit image of the object can be computed.

The complete relighting process is schematically summarized in figure 2.12. In figure 2.13 some additional results of relighting the Formula One toy car under different incident illuminations are shown.

2.4 Reflectance Fields and Functions

The reader should now have an intuitive understanding of image-based relighting. Before introducing a well-defined mathematical framework for image-based relighting in the following chapter, two very important concepts need to be introduced: *reflectance fields* and *reflectance functions*.

Reflectance Field. A reflectance field describes how an object's appearance is influenced by incident illumination. In case $2D$ incident illumination is used, and the viewpoint is fixed, this is a $4D$ function (i.e., $2D$ pixel position and $2D$ incident

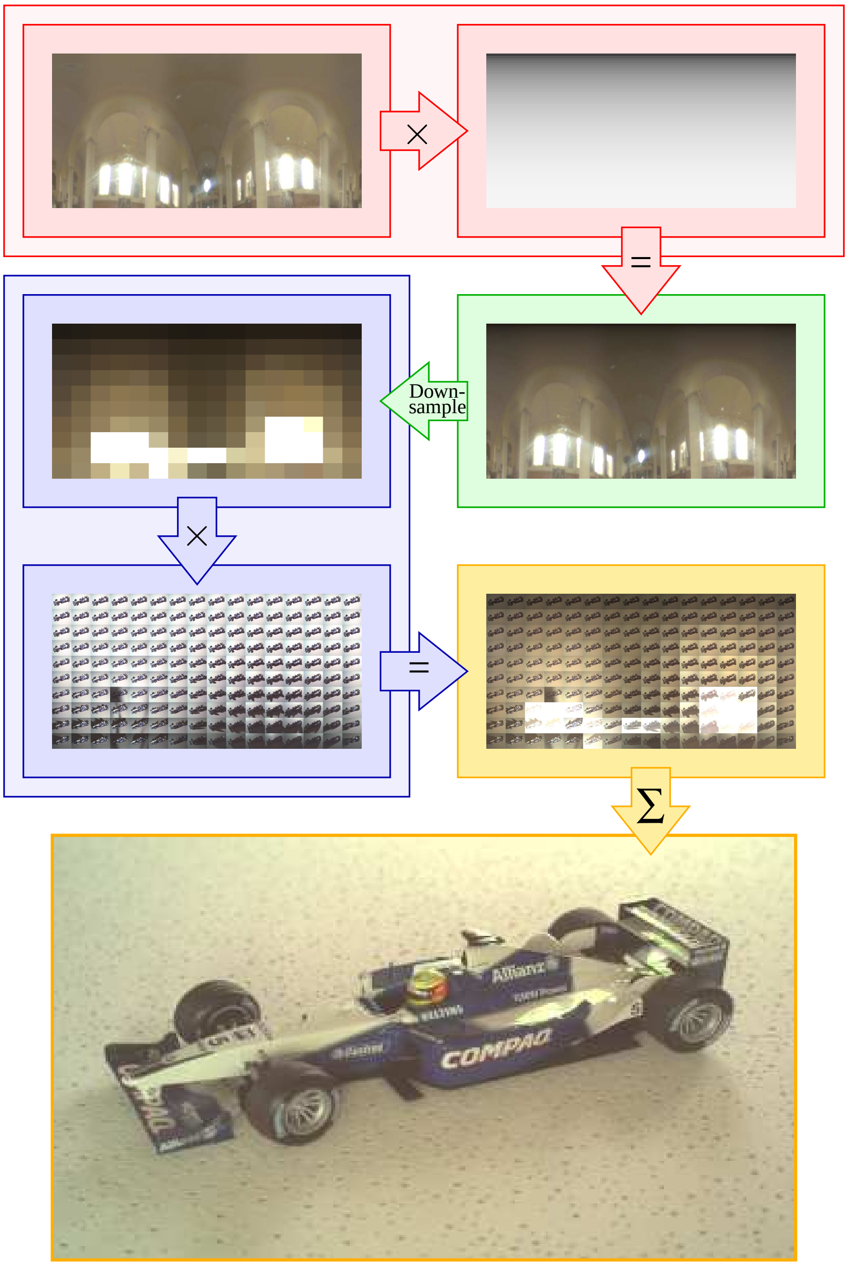


Figure 2.12: The image-based relighting process schematically illustrated. **Red:** the incident illumination is scaled by the Jacobian to compensate for the non-uniform sampling of the spherical bounding volume. **Green:** The scaled incident illumination is downsampled to match the Light Stage sampling resolution. **Blue:** The recorded images are scaled by the downsampled incident illumination. **Orange:** The intensity-scaled recorded photographs are summed into the resulting relit image.

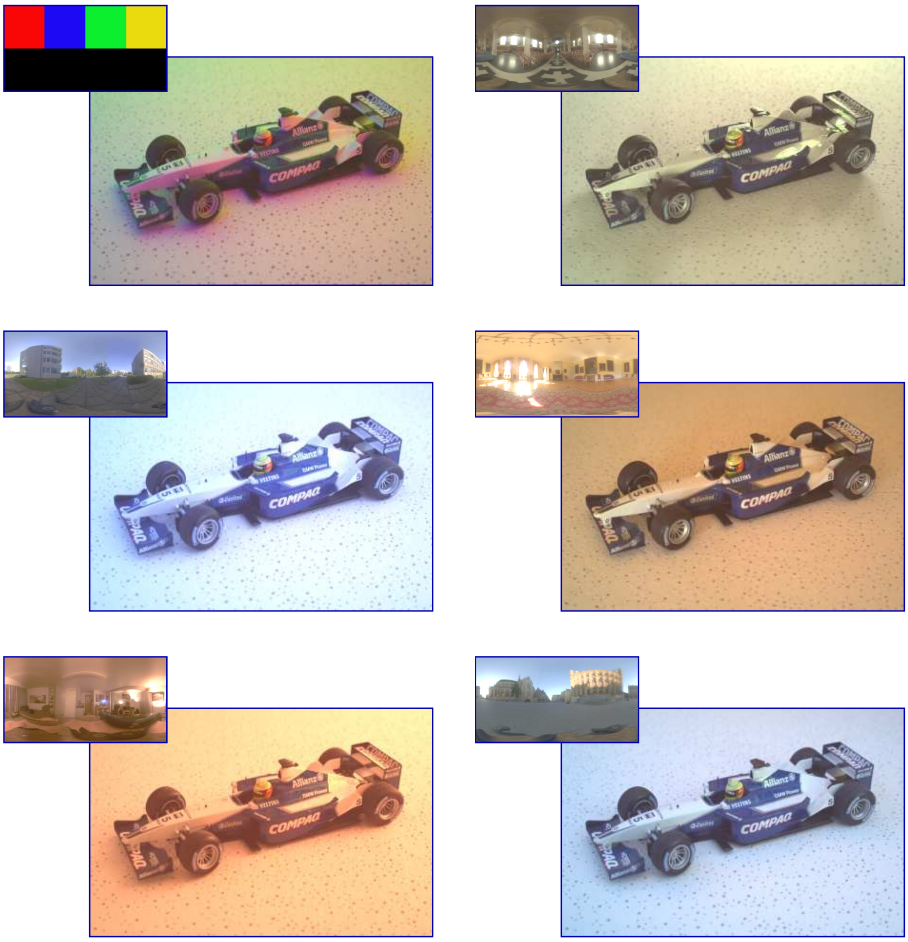


Figure 2.13: Some results of relighting the Formula One toy car example. (Environment maps: courtesy of Vincent Masselus)

illumination). In fact, figure 2.11 can be seen as a visualization of the reflectance field of the Formula One toy car. Once the reflectance field of an object is known, a relit image can be easily computed. Thus, image-based relighting can be redefined as *“the capturing of the reflectance field of a real object”*.

Reflectance Functions. A reflectance field is a difficult concept to visualize mentally, because it is a 4*D* (or higher dimensional) function. A reflectance function, on the other hand, is a 2*D* slice (in case of 2*D* incident illumination) of the reflectance field, and is much easier to handle. A reflectance function is defined for each pixel of the viewpoint, and is parameterized over incident illumination. Consider again the Formula One toy car example. In figure 2.14, the reflectance functions of a number of

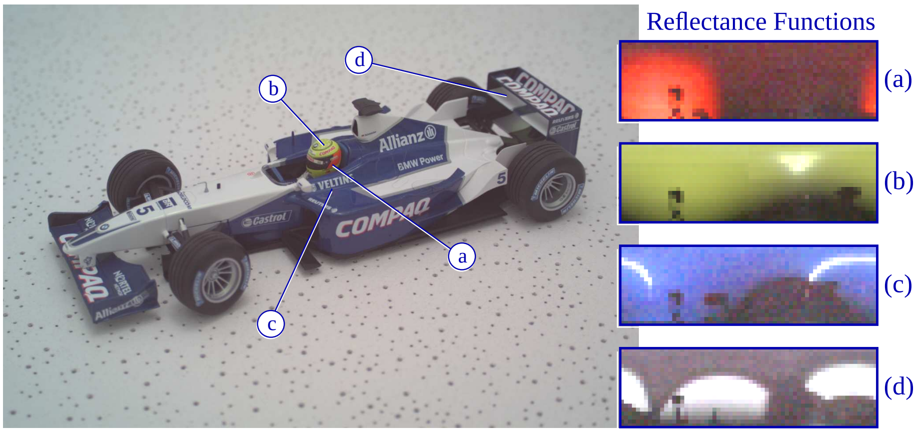


Figure 2.14: Four selected reflectance functions. Note that the dark spot visible in each of the reflectance functions, corresponds to the dark spot, caused by the camera blocking a light source, in figure 2.11. Each reflectance function is created by copying the pixel values from the same location in each of the recorded photographs.

pixels are shown. Basically, copying intensity values from the same pixel location of each recorded photograph, results in the reflectance function of that specific pixel.

Interpreting the relighting process in terms of reflectance functions yields an interesting view on image-based relighting. In the previous section image-based relighting is achieved by summing the recorded photographs scaled with the corresponding (corrected) incident illumination intensities. Since these operations are performed per pixel, it is easy to see that the same operations can be applied on the intensities in the reflectance functions (figure 2.15). The main difference between computing a relit image from the recorded photographs directly and computing a relit image using the reflectance functions, is the order of processing. In the first case the outer loop goes over the incident illumination, while the inner loop goes over the different pixels. In the second case, the outer loop goes over the reflectance functions, and thus pixels, while the inner loop goes over the reflectance function itself, and thus the incident illumination.

A reflectance function is sometimes called an apparent BRDF. However, it is important to realize that a reflectance function is **not** a real BRDF. The main difference between both is that a BRDF only contains local reflection effects due to direct illumination, while a reflectance function can also contain the effects of indirect reflections through the scene. A BRDF is associated with a single material and is independent of the object's geometry. A reflectance function, on the other hand, can be the cumulative effect of multiple interactions with different materials in the scene. To avoid confusion we will always use "reflectance function" and not "apparent BRDF".

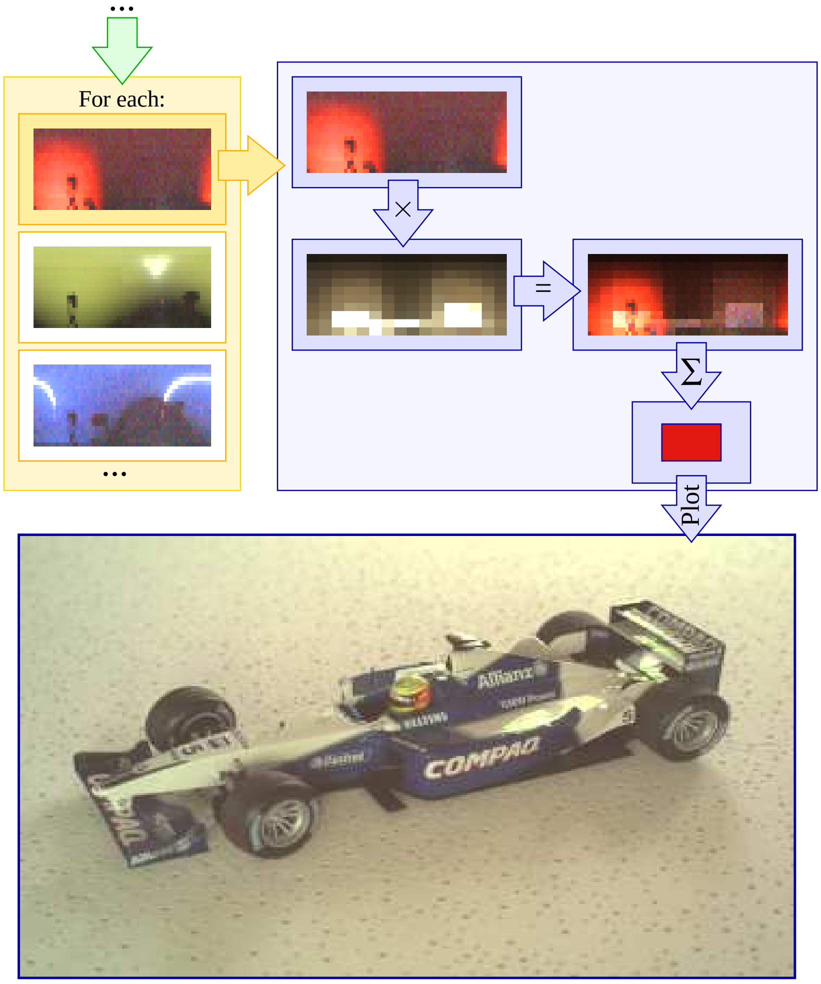


Figure 2.15: Image-based relighting using reflectance functions. **Green:** Identical to figure 2.12 from the start until the green arrow. **Orange:** For each reflectance function the *blue* part is repeated. **Blue:** The reflectance function is multiplied, and summed into the relit pixel value. This pixel value is copied into the final image.

2.5 Conclusion

In this chapter an intuitive description of image-based relighting was presented. The acquisition, approximation, representation, and interpretation of incident illumination formed the basis on which image-based relighting were detailed. The core idea behind all modern relighting techniques is the summation of reflectance values scaled by the incident illumination. Finally, the important concepts of reflectance fields and reflectance functions were presented.

3

Theoretical Framework

In this chapter a theoretical framework for image-based relighting is introduced. The goal is to create a solid theoretical basis on which the methods presented and discussed in this work will be founded. This framework enables to explore the limits of existing and novel methods, while keeping the notational overhead to a minimum.

3.1 Introduction

Although the number of publications on image-based relighting is already significant, a single unifying framework has not yet been proposed. Each paper uses its own, in many cases incompatible, notations, approximations, and derivations. Sometimes the theoretical context is so different, that at first, it is not clear that they are dealing with the same problem.

Early papers [26, 27, 31, 52, 69, 70] on image-based relighting use theoretical derivations based on the *rendering equation* [42]. The rendering equation, however, is not a good basis for developing a general image-based relighting theory, because it requires a detailed knowledge of the scene geometry. One of the few exceptions are [51, 113, 114], which found inspiration in image-based rendering techniques, more specifically the Lumigraph [33] and Light Fields [50].

Most image-based rendering methods start with the plenoptic function [1]. It also forms the basis of the theoretical framework of the Light Stage developed by Debevec et al. [20]. The plenoptic function is a 7*D* function (i.e., 3*D* position, 2*D* direction, 1*D* wavelength and 1*D* time) describing the flow of light in a scene. The framework developed in [20] forms the basis of a whole family of image-based relighting techniques such as [21, 30, 34, 36, 57, 58, 59, 65, 66, 87], and can be considered to be the de-facto standard theory of image-based relighting.

However, the framework developed by Debevec et al. [20] is specifically geared towards the Light Stage. In chapter 6, an acquisition setup different from a Light Stage is introduced. To accommodate both the Light Stage and this newly developed setup, a more general theoretical framework of image-based relighting is required.

The basic idea is to view the scene as a black box system (see figure 3.1 for an iconic illustration). This idea was already present in Debevec et al.'s framework, but to a lesser extent than in our framework. A theoretical framework must be supported

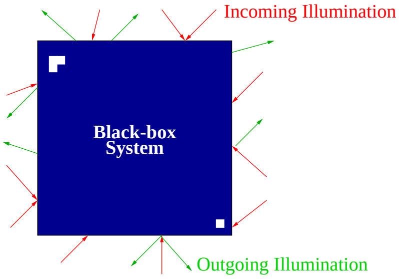


Figure 3.1: In this thesis the light transport is considered to be a black-box system. The goal of image-based relighting is to predict how the output of the system reacts to novel input. In our case, the input is incident illumination (red arrows), and the output is exitant illumination (green arrows).

by a well-defined mathematical notation. In this work we will introduce a notationally light framework. Furthermore, the apparent dimensionality of the system is reduced by means of serialization functions, which enable to develop theories and techniques on low dimensional functions, that can be extended to the full dimensional plenoptic setting.

Having a solid theoretical framework and a well-defined mathematical notation has a number of advantages:

- First of all, it helps to understand other image-based relighting techniques, even if these do not use the same theoretical framework. Not only does it help in understanding these other techniques, but also helps in validating and exploring the limits of these other methods.
- Second, new, non-trivial, methods can be easily developed once the rules and constraints of the problem are well defined. For example, the methods developed in this work are all designed using the framework developed in this chapter.
- Third, having a solid mathematical notation makes it easier to exchange fundamental ideas with other research domains such as discrete mathematics, physics, computer vision, ...

This chapter begins, in section 3.2, with a thorough study of the underlying physical principles and constraints that come into play in image-based relighting. Next, an elegant compact mathematical notation used throughout this work is derived in section 3.3. Finally, this framework is applied (section 3.4) to the Light Stage algorithm intuitively presented in chapter 2.

3.2 Physical Constraints

Before developing a mathematical framework, the physical constraints that apply to image-based relighting need to be investigated in detail. By understanding these con-

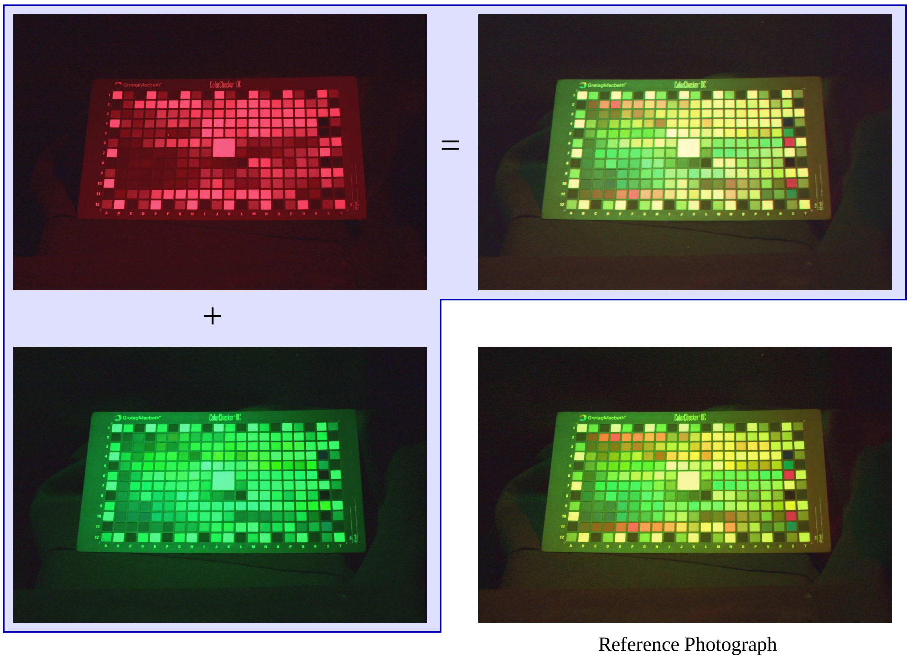


Figure 3.2: An empirical demonstration of linearity of light transport. A red and a green light source are aimed at a color checker chart. Two HDR photographs of the color checker are recorded, each lit by a different light source. Due to the linearity of light transport, the sum of these two images equals an HDR photograph of the color checker lit by both light sources at the same time. Both the sum and a reference HDR photograph (with both light sources lit) are shown.

straints, a better understanding of the accuracy and limits of an image-based relighting method is reached.

In this work the following physical and practical constraints are imposed: linearity of light transport, a time-statical scene, an orthogonal *RGB* colors pace, and a fixed vantage point.

3.2.1 Linearity of Light Transport

Linearity of light transport is the core physical principle behind image-based relighting and is based on the superposition principle. The superposition principle basically states that when two light waves (electromagnetic radiation) travel through a linear medium, the resultant apparent wave at any point can be described as the algebraic sum of the two light waves. This is illustrated in figure 3.2. A more fundamental principle that lies at the basis of the superposition principle, and thus of the linearity of light transport, is the conservation of energy, implying that a light wave cannot loose or gain energy, without interacting with matter. This implies furthermore that a light wave cannot influence the state of another wave[1].

More formally, linearity of light transport implies:

$$w_a f(\mathbf{a}) + w_b f(\mathbf{b}) = f(w_a \mathbf{a} + w_b \mathbf{b}),$$

where w_a and w_b are scalar weights, **a** and **b** are illumination conditions (i.e., a light source at a specific position, emitting light at a specific intensity and with a specific color), and $f(\cdot)$ is the light transport of an illumination condition through a scene. Furthermore, we assume that this scene does not emit any illumination. Or in words: the net effect of w_a times the effect of illumination condition **a** on a scene, plus w_b times the effect of illumination condition **b** on the same scene, equals the effect of w_a times illumination condition **a** plus w_b times illumination condition **b** on the scene.

In everyday life, this superposition principle holds. However, a recent branch of optics research, called *non-linear optics* [9] focuses on the exceptions in which this is not the case. By focusing a very intense laser beam on a refractive material, the index of refraction can change depending on the intensity, this is also called the “Kerr effect”. Other examples include four-wave mixing, self-phase modulation, Raman scattering, ... (see [9] for an overview). However, most of these effects are only significant in the presence of very intense laser light, therefore these effects can be safely ignored in the context of image-based relighting. Everyday light effects such as reflection, diffraction, dispersion, interference, ... are all linear optical effects.

3.2.2 Static Scene

In this work we assume that a scene does not change, and it is static over time (or at least during acquisition). Furthermore, we assume that light transport is instantaneous. The rationale behind this constraint is that, allowing a scene to change over time, introduces an extra dimension, i.e., time, of the plenoptic function which needs to be captured.

This constraint is broken, among others, in the following cases:

- **Moving objects.** All objects in a scene must have zero velocity during the duration of the acquisition. Care has to be taken when a scene includes objects which contain fur or hair. Even a faint breeze might move some of the fur or hair, resulting in blurred relit images.
- **Organic material.** Special attention has to be taken when capturing organic materials such as plants and leaves. Organic material can dehydrate, and thus alter significantly in reflectance properties. The heat from light sources could accelerate this process.

[1] This also holds for constructive and destructive interference. In both cases, it describes the **net** effect of the superposition principle applied to both waves. It does not imply that the two waves are *replaced* by a new wave.

- **Phosphorescence.** Phosphorescence is a chemical process in which light is *"absorbed"* by a material, and after a delay, re-emitted. This breaks the constraint of instantaneous light transport. Practically, this problem can be solved by waiting until the effect of the previous illumination condition has expired before applying a novel illumination condition.

Recently, a number of papers [66, 109] try to remove this constraint in order to capture the reflectance field of a video sequence. These techniques still use the same image-based relighting algorithm, but try to minimize the acquisition period such that the dynamic scene is quasi-static.

3.2.3 RGB Color Space

Color is a complicated topic, which is ignored or greatly simplified in much of the computer graphics literature, to great annoyance of color scientists. For an overview of color and color theory, see [99]. Color sensations are induced by light waves with different wavelengths, ranging from $380 \sim 400$nm (violet) to $700 \sim 780$nm (red). This wavelength interval is called the visible spectrum. The human eye consists of three different types of "sensors", each is (significantly more) sensitive to a limited sub-range of the visible spectrum, and thus light with a wavelength outside these sub-ranges cannot be seen. Because of this, the perceived colors can be regarded as a 3 dimensional space of colors. A consequence of this is that a great number of different color spectra are seen by the human eye as the same color, and are called metameric colors.

The space of all perceived colors can be represented by three basis color spectra, since colors are perceived as a $3D$ color space, This principle is called trichromacy. In consumer electronics, such as computer screens, digital cameras, ..., the most commonly used basis spectra correspond to red (R), green (G) and blue (B) observed (basis) colors. A new color sensation is generated by weighting and adding these basis spectra:

$$C = w_R R + w_G G + w_B B,$$

where C is the resulting spectrum and w_R, w_G, w_B are the scalar weights of respectively red, green and blue. It is important to realize that although the space of perceived colors can be (partially) generated (using R, G, and B), this does not include all visible spectra.

In computer graphics all computations are performed directly on RGB vectors. This implies that the scene characteristics are specified in RGB components: BRDF evaluations are directly performed in RGB, light source colors and intensities are specified in RGB, ... In other words, the simulated world is completely specified in RGB tuples. In image-based techniques the same assumption is made. However, this is not completely accurate.

Real world light sources, for instance, can have a completely different spectrum, but still look identical (i.e., metameric colors). Illuminating a surface with these identical looking light sources does not necessarily yield the same result, depending on the spectral response of the surface. This is illustrated in figure 3.3. This particular problem was also noted by Wenger et al. [110]. Wenger et al. proposed three different solutions: spectral illuminant matching, metameric illuminant matching, and metameric reflectance matching. *Spectral illuminant matching* tries to match the spectrum of the desired illumination as closely as possible to the available light sources. However, due to the limited freedom, the match is far from perfect. *Metameric illuminant matching* takes a similar approach as the previous one, except that it tries to match the perceived (by the camera) color as closely as possible. Finally, *metameric reflectance matching* takes the spectral response of the whole light transport system in account. Although the results of the latter are very impressive, the usability in terms of general image-based relighting is limited, since we are actually trying to recover the response of the system with respect to incident illumination.

A related problem is that the spectral response of a material cannot be directly observed, and that it is always relative to the spectrum of an illuminant. Specifically, in the case of image-based relighting, a series of photographs of the scene is recorded, each under different illumination conditions. These illumination conditions have a specific spectrum, and the response (spectrum) of the scene appearance is thus relative with respect to the color spectrum of the illumination conditions. Theoretically, if all spectra (of the light sources and surfaces) are known, the effect of the spectra of the illumination conditions can be divided out. However, when representing colors in *RGB*, spectral information is lost and approximation errors will be unavoidable. For

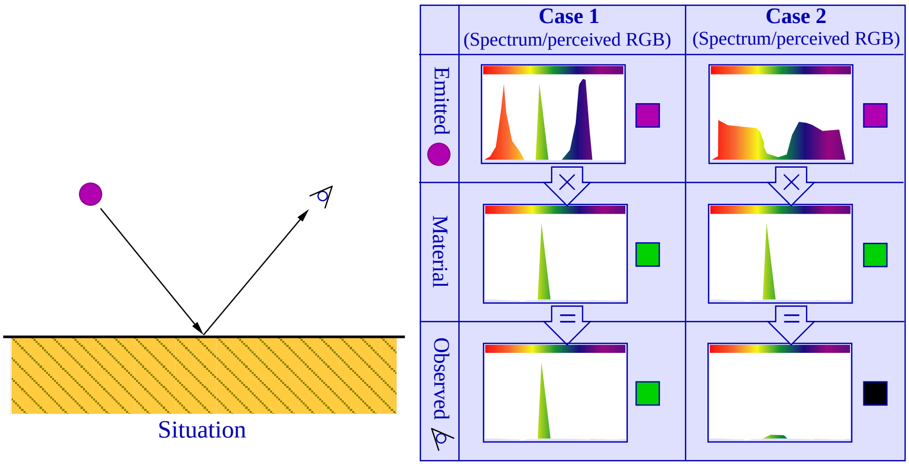


Figure 3.3: The effect on the perceived color, when using light sources with metameric colors (i.e., the same perceived color), purple in this case. Depending on the spectrum of the light sources and the underlying material, a significant portion of the reflected spectrum can differ, resulting in a different observed *RGB* color.

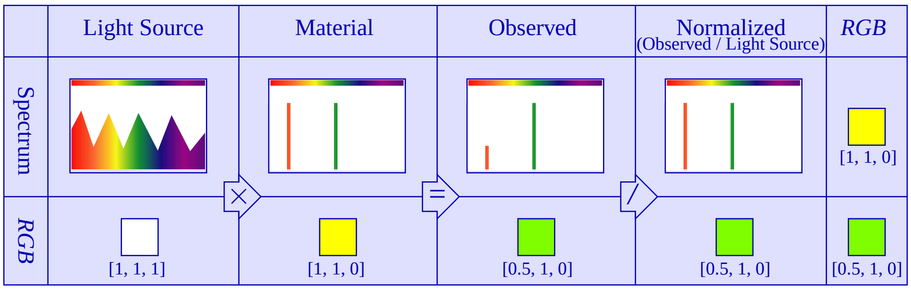


Figure 3.4: The observed color depends on the spectrum of the light source (see also figure 3.3). By dividing the observed spectrum with the spectrum of the light source, the true spectrum of the material can be computed. However, doing this normalization using *RGB* colors, does not yield the correct result.

example, consider the case in figure 3.4. The light source has a jagged spectrum, perceived as a white color. The material's spectrum, contains two peaks, red and green, and is perceived as yellow. The observed color (i.e., product of the light source color and the material color), is bright green. Dividing out the spectrum of the light source, yields the original color of the surface when working multi-spectrally. However, computing the normalization in *RGB*, does not yield the correct color of the material.

Berns et al. [6] state that the only way to ensure a color match for all observers and across changes in light sources is to work multi-spectrally. However, the current state of hardware does not yet facilitate easy multi-spectral acquisition. Therefore, in image-based relighting these problems are ignored, and the assumption is made that all components can be exactly represented and computed by orthogonal *RGB* tuples.

3.2.4 Fixed Viewpoint

This final constraint is more a practical one than a physical constraint. Capturing the appearance of an object for variable viewpoints increases complexity, because two extra dimensions need to be captured and represented.

In contrast to image-based relighting, image-based rendering techniques (for an overview see [116]) focus on the acquisition of the appearance of real objects under *fixed* illumination, but for a variable viewpoint. Because the incident illumination is fixed beforehand, these methods try to capture and represent a 4*D* slice of the plenoptic function [1]. Although related to image-based relighting, the problems faced in image-based rendering are significantly different. The main problem in image-based rendering is the interpolation or reconstruction of the sampled 4*D* slice of the plenoptic function without introducing visual artefacts, while maintaining an as low as possible sampling rate.

Image-based relighting captures a different slice of the plenoptic function than image-based rendering. By limiting image-based relighting to a single viewpoint, a number of difficulties are avoided. First of all, as mentioned before, less dimensions need to be sampled. Basically, the camera becomes part of the scene, and thus is located at a fixed position. Since the camera is static, no extrinsic and intrinsic camera calibration is needed, nor is any knowledge of the shape of the object required. Second, each pixel can be considered as a single independent entity, because the pixel's size and location are completely determined by the camera which is fixed. This allows to focus completely on the effects of incident illumination on the scene.

It is clear that at some point, the fixed-viewpoint constraint will have to be dropped in order to achieve full digitalization of real objects. Image-based rendering techniques will without any doubt play an important role in this. Initial, but cautious, steps have already been taken by relighting dynamic scenes (subsection 3.2.2), and in work related to *opacity hulls* [62, 63, 76]. The next, most logical, step in removing this constraint is probably the relighting of controllable (animated) dynamic scenes.

3.3 Notations and Operators

As mentioned before, previous work contains a good mathematical and physical framework, most notably [20], which is further extended in [56]. In this work we present a novel framework, which is in spirit identical, but uses a different notation. In our opinion, this new notation mirrors better the physical situation, incorporates the constraints implicitly, and offers a more flexible framework to include new methods.

In the following section the basis of our mathematical framework is introduced. At this point we ignore the size/dimensionality of the input and output data. In subsection 3.3.2 a number of crucial operators are introduced. These operators work on in- and output-data, and transform it into a more suitable form for the previously introduced notation. Finally, the properties of the whole mathematical framework are detailed in subsection 3.3.3.

3.3.1 Notations

As illustrated iconically in figure 3.1, light transport through a scene is considered as a black box system. Additionally, from the linearity constraint (subsection 3.2.1) follows that this is a **linear black box** system. This can be compactly denoted in a matrix notation:

$$C = \mathbf{T}L + S, \tag{3.1}$$

where:

- C is a vector of length k, representing the total exitant illumination of the black box system. In this work, this will be a camera image, serialized into a vector.

- L is a vector of length l, representing the external incident illumination onto the black box system. This can, for example, be an environment map, stacked into a vector, or a general $4D$ incident illumination field serialized into a vector. In case of the Light Stage [20], each element in this vector L corresponds to the emitted intensity of a single light source at a specific location on the surrounding sphere.

- S is a vector of length k, representing the self-emitted illumination of the scene. In other words, this is the exitant illumination of the scene when *no* external incident illumination is applied. The term "self-emitted illumination" should be interpreted in a broad sense. Any illumination from or onto the scene, which is not parameterized by the incident illumination vector L is included in this term. This term is independent, and thus constant, with respect to L. In many cases we will drop this term or implicitly assume that it is zero.

- $\mathbf{T}$ is a $k \times l$ matrix, representing the light transport of externally applied incident illumination onto the scene.

In subsection 3.2.3 the assumption is made that all computations can be exactly represented and computed with *RGB* tuples. This implies that equation (3.1) can be used for each color channel separately. Furthermore, it can be extended for any desired number of color channels (e.g., multi-spectral).

Equation (3.1) can be used to describe the acquisition of the light transport matrix $\mathbf{T}$. During acquisition, a series of incident illumination conditions $\{L_i\}_i$ are applied to the scene, and the resulting exitant illuminations $\{C_i\}_i$ are captured by taking photographs. In other words, by applying L_i to the scene, nature actually computes equation (3.1), and returns C_i which is captured by means of an HDR photograph.

Relighting can also be described using equation (3.1). When computing a relit image, the transfer matrix $\mathbf{T}$ is known, as are the self-emittance S of the scene and the incident illumination L. Filling these in equation (3.1), results in the relit image C.

The previous description allows to reformulate image-based relighting in a more formal manner: "*Image-based relighting attempts to infer the transport matrix* $\mathbf{T}$ *by observing the exitant illumination C under a series of user-controlled incident illumination L*". Formulating the relighting problem in this way, allows to regard it as an abstract mathematical problem. We will show in future chapters, that by applying non-trivial mathematical methods to equation (3.1), efficient novel image-based relighting methods can be obtained.

Before discussing the properties of equation (3.1) in detail, we first need to specify the serialization and resampling/discretization operators.

3.3.2 Operators

Equation (3.1) deals with **serialized and discretized** input and output illumination vectors L and C. Until now, this serialization and resampling (discretization) has not been formalized. In this section operators will be introduced to handle: *serialization of the incident illumination*, *serialization of the exitant illumination*, *exitant illumination resampling*, and *incident illumination resampling*.

Incident illumination serialization operator $\mathcal{S}_L$. A vector is a discrete list, while incident illumination can be either $2D$ or $4D$. Some form of projection has to be done to map this high dimensional data onto a vector. We denote this by a serialization operator $\mathcal{S}_L$. For example, suppose the incident illumination is given as $2D$ environment map in a latitude-longitude parameterization, discretized in a resolution of $n \times m$. In this case the serialization operator can copy an intensity value with a latitude-longitude coordinate of (θ, ϕ) to a serialized vector position $m\theta + \phi$. The exact form of serialization is often not important. It is clear that the number of discrete elements in the multi-dimensional incident illumination must be finite (either by construction, otherwise by resampling). Note that this operator is reversible, i.e., the vector of incident illumination L can be expanded back into the original discretized multi-dimensional incident light field. This implies that no information is lost during serialization. The inverse operator is denoted by ${\mathcal{S}_L}^{-1}$.

Exitant illumination serialization operator $\mathcal{S}_C$. Similar to the serialization of the incident illumination is the serialization operator of the outgoing illumination. Due to the assumption that the viewpoint is fixed, this operator transforms a $2D$ photograph (from a fixed viewpoint) into a vector. The inverse serialization is denoted by ${\mathcal{S}_C}^{-1}$.

Incident illumination resampling operator $\mathcal{R}_L$. The multi-dimensional incident light field is in most cases captured at a higher resolution than required for serialization into L, or is provided as a continuous function. Additionally, the parameterization of the incident light field often does not correspond to the (desired) parameterization of the reflectance field. A resampling operator is necessary to *downsample*, *discretize*, and *reparameterize* the incident light field to a suitable resolution and form. It is important to note that this is not a reversible operation. An approximate reconstruction operator $\widetilde{{\mathcal{R}_L}^{-1}}$ can be defined. However, this reconstruction operator is not uniquely defined. The reverse operation must follows the following rule:

$$L = \mathcal{R}_L\left(\widetilde{{\mathcal{R}_L}^{-1}}(L)\right).$$

Because the resampling operator $\mathcal{R}_L$ removes information from the incident light field L, for example by downsampling, the inverse operation $\widetilde{{\mathcal{R}_L}^{-1}}$ has to fill-in this missing information. Since this information is not known, some heuristic is used (e.g., by maintaining smoothness) to generate this information. Depending on this heuristic, different inverse operators are obtained.

Exitant illumination resampling operator $\mathcal{R}_C$. In this work, no exitant illumination resampling is used. The resolution of the acquisition camera is used when generating relit images. A resample operator for the exitant illumination can be useful when acquiring a scene from a variable viewpoint. The inverse resampling operator would, in this case, basically be an image-based rendering algorithm.

Given a multi-dimensional incident light field $i(\mathbf{x},\omega)$ defined over position $\mathbf{x}$ and direction ω, the resulting acquired or relit image $c(\mathbf{x}_p)$, over pixel positions $\mathbf{x}_p$ is defined by:

$$c(\mathbf{x}_p) = \mathcal{S}_C{}^{-1}\left(\mathbf{T}\mathcal{S}_L\left(\mathcal{R}_L\left(i(\mathbf{x},\omega)\right)\right)\right). \tag{3.2}$$

The exitant illumination resampling operator $\mathcal{R}_C(\cdot)$ in equation (3.2) is omitted since the assumption of a fixed viewpoint is made.

3.3.3 Properties

In this section the properties of equations (3.1) and (3.2) are discussed in detail, followed by a brief overview highlighting the advantages over existing notations. Finally, the developed framework is applied to “Dual light transport” to illustrate its flexibility.

As noted before, the vector L has length l. Given the fact that the serialization operator $\mathcal{S}_L$ is a one-to-one mapping of $\mathcal{R}_L(i(\mathbf{x},\omega))$ to L, this implies that the total number of discrete elements in the incident illumination equals l. For example, if the incident light field is represented by a $2D$ environment map, then the width times the height of the environment map equals l.

The vector C is a vector of length k: so the number of pixels in the acquired photographs, and thus also in relit images, equals k. The same holds for S.

Finally, the transport matrix $\mathbf{T}$ is a $k \times l$ matrix. In other words, the height of the transport matrix $\mathbf{T}$ equals the number of pixels in the relit images. The width of $\mathbf{T}$ equals the number of samples acquired, i.e., the number of illumination conditions applied to the scene during acquisition. The transport matrix $\mathbf{T}$ is in fact the reflectance field. The rows $\mathbf{T}_i$ of $\mathbf{T} = [\mathbf{T}_1;\ldots;\mathbf{T}_k]$ are the reflectance functions. Note, that there are as many rows as there are pixels in C. The interpretation of the columns $\mathbf{T}_{\cdot,j}$ of $\mathbf{T} = [\mathbf{T}_{\cdot,1}|\ldots|\mathbf{T}_{\cdot,l}]$ can be easily inferred from equation (3.1). Consider the incident illumination $L_j = [\delta_{i,j}]_i$. Filling this in equation (3.1) gives the following resulting exitant illumination $C = \mathbf{T}L_j = \mathbf{T}_{\cdot,j}$. In other words, the columns $\mathbf{T}_{\cdot,j}$ of $\mathbf{T}$ are the serialized recorded photographs of the scene illuminated by a single light corresponding to the j-th element in the incident illumination vector L.

The serialization operators, at first might seem to complicate the mathematical framework. However, the serialization operator is fully reversible. An advantage is that in many cases an equivalent reasoning can be applied to both the full-dimensional problem or the serialized notation. This makes it much easier to develop new relighting and acquisition methods.

A popular way of looking at image-based relighting, is from the perspective of a reflectance function (as depicted in figure 2.15). To compute the relit pixel value c_i of the i-th pixel in the camera image C, all elements in the incident illumination L and the pixel's reflectance function $\mathbf{T}_i$ are multiplied and summed. Mathematically, this boils down to a vector dot product:

$$c_i = \mathbf{T}_i \cdot L. \tag{3.3}$$

This notation is especially attractive when using the integral notation of Masselus [56].

Comparison with existing frameworks. The introduced notation and framework has a number of advantages. It makes a clear separation between the approximation of the $4D$ incident light field and the light transport. The approximation is implicitly encoded in the resampling and serialization operators on the incident illumination, $\mathcal{R}_L$ and $\mathcal{S}_L$ respectively, and can be specified to the constraints and features of a specific image-based relighting method, while the linear light transport is invariably encoded by equation (3.1). In the notation of [20], all is encoded in a single integral. Furthermore, the derivation of the presented framework, follows directly from the physical constraints, dominated by the linearity of light transport. The derivation of the notation of [20] requires more effort and assumptions to arrive at a similar integral equation. Finally, it should be stressed that both notations basically are the same in spirit, only the assumptions, derivation, and interpretation differ.

Dual Light Transport. To demonstrate the flexibility of our framework, we examine the case of dual light transport [35, 91]. Dual light transport is based on Helmholtz reciprocity applied to an image-based setup. As in image-based relighting, a series of photographs of an object are recorded under different illumination conditions. These illumination conditions are in fact a special case of $2D$ incident illumination. Each of the lighting conditions consists of a directional light source, having a common point of origin (i.e., apex in a cone of directions). The illumination conditions are chosen such that an imaginary plane is "scanned" by the directional light sources. We will call this imaginary plane, the projection plane. Such illumination conditions can be obtained by using a projector [91] or by using a laser setup and a galvanometer [35]. See figure 3.5 for an illustration of the setup. The main idea is that, since light transport is reciprocal, the role of the camera and the light sources can be switched. In other words, dual light transport enables to create an image of a scene, seen from the projector/laser's apex point, and illuminated by "virtual" light emitted from the camera, and this without actually placing the camera at the projector/laser's apex point.

To understand how this fits in the presented framework, first consider what a single element $\mathbf{t}_{i,j}$ of $\mathbf{T}$ represents. This element $\mathbf{t}_{i,j}$ is the ratio of light emitted under

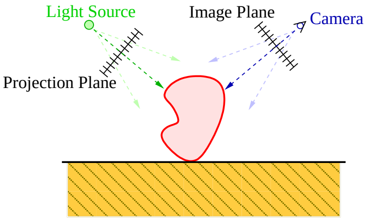


Figure 3.5: Dual Light Transport. A directional light source “scans” a projection plane. For each location on the projection plane, a photograph is recorded. Afterwards, the dual image can be computed, viewed from the light source, and lit by virtual illumination coming from the camera.

the j-th illumination condition, and arriving at pixel i of the camera image (excluding the self-emitted illumination by scene). The j-th illumination condition is in this case a beam of light going through the j-th position on the projection plane. Due to Helmholtz reciprocity, the direction of the light transport can be reversed. Thus $\mathbf{t}_{i,j}$ also represents the ratio of light emitted through the i-th camera pixel, and arriving through the j-th point on the projection plane at the projector’s apex. Extrapolating this to all points on the projection plane gives that a row $\mathbf{T}_i$ of $\mathbf{T}$ equals the resulting “image” at the projection plane, when emitting a beam of light through the i-th pixel from the camera. As noted before, concatenating the recorded “images”, serialized in rows, in a matrix, results in the dual transport matrix, which equals $\mathbf{T}^T$.

Denote the dual observed image (on the projection plane) as C'' and the dual illumination condition (emitted from the camera) as L'', then the dual light transport can be written as:

$$C'' = \mathbf{T}^T L''.$$

It is important to realize that the dual light transport of emitting a recorded photograph, obtained under any illumination condition L, does not result back in a dual photograph equal to the originally applied illumination condition L:

$$\begin{aligned} C &= \mathbf{T}L, \\ L &\neq \mathbf{T}^T C. \end{aligned}$$

Physically this follows from the fact that absorption during light transport, works in both ways, and lost energy cannot be gained in either direction. Mathematically this can be understood from the fact that $\mathbf{T}^T \neq \mathbf{T}^{-1}$. Note that no self-emittance term S is included in the derivation above. In general, the dual light transport cannot be computed in presence of self-emittance.

3.4 Application to the Light Stage

In this section an application of the theoretical framework to the intuitive relighting algorithm introduced in the previous chapter (section 2.3) is presented. First the operators are specified (subsection 3.4.1), followed by a study of the transport matrix **T** in subsection 3.4.2.

3.4.1 Operators

By defining the different serialization and resampling operators, the exact form of **T** is set. In the following chapter, we will see that the impact of the resampling operator can be pretty significant for the compactness of the matrix **T**.

In the case of the Light Stage, the acquisition and relighting process is already fixed. In order to match the acquisition and relighting process as closely as possible to the developed framework, the serialization and resampling operators need to be designed with this goal in mind. As a guideline, we would like to have a one-to-one correspondence between each element in L and a light source (position) on the Light Stage device.

Serialization Operator $\mathcal{S}_L$ and $\mathcal{S}_C$. Both serialization operators are similar, and transform a $2D$ discretized function into a vector. Any standard serialization method can be utilized, such as row-major and column-major serialization.

Resampling Operator $\mathcal{R}_L$. The resampling operator downsamples the environment map in latitude-longitude parameterization to the sampling resolution of the Light Stage. It also pre-multiplies the environment map with the Jacobian to take the solid angle of each pixel (i.e., light source area) into account.

3.4.2 Application

Acquisition. The reflectance field of a scene is acquired by recording HDR photographs of the scene. For each photograph a different light source is lit. As defined before, each element l_i in the $2D$ incident light field L corresponds to the state of a single light source at a specific location (θ, ϕ) defined on the bounding sphere by $\mathcal{S}_L{}^{-1}(l_i)$. As detailed in the previous subsection 3.3.3, this implies that each column in **T** corresponds to a recorded HDR photograph of the scene under this incident illumination. From this it follows that creating the transport matrix **T** is trivial once all HDR photographs are recorded: each photograph C_i $(= \mathbf{T}L_i$ with $L_i = [\delta_{i,j}]_j)$ is serialized using $\mathcal{S}_C$ and copied to the i-th column of **T**.

Relighting. Again, relighting is trivial once the transport matrix **T** is known. The novel incident illumination $i(\mathbf{x}, \omega)$ is transformed, first by the resampling operator $\mathcal{R}_L$, followed by a serialization operator $\mathcal{S}_L$. Next, the matrix-vector multiplication of equation (3.1) is computed. Finally, the resulting vector C is de-serialized using $\mathcal{S}_C{}^{-1}$.

3.5 Conclusion

In this chapter a theoretical framework for image-based relighting was introduced. The supporting mathematical framework is a direct expression of the physical constraints that come into play during image-based relighting. In this work, there are four major physical constraints: *linearity of light transport*, *time-invariance of the scene*, *idealized RGB color space*, and *a fixed viewpoint*. The mathematical framework consists of two parts: *a linear matrix system* expressing the light transport through the scene, and a collection of *serialization and resampling* operators that transform the input and output data into a convenient form.

4

Sampling Reflectance Functions

In this chapter the sampling and the reconstruction of reflectance functions acquired by means of a Light Stage [20] are discussed. The intuitive algorithm discussed in chapter 2, and formalized in chapter 3, can generate convincing looking results. However, as will be shown in this chapter, a significant improvement of the quality of the results can be obtained by reconstructing the reflectance functions. This allows to compute relit images using high resolution incident light fields, yielding more accurate relit images. The ideas presented in this chapter have also been partially published in [59] and [56].

4.1 Introduction

The Light Stage samples the reflectance field of a real object from a set of predetermined fixed illumination directions. Due to the construction, this sampling is usually uniform in a latitude-longitude parameterization. A notable exception is the Light Stage v3 [21]. This Light Stage, however, is intended for real-time performance relighting rather than acquiring reflectance fields.

Since the reflectance field is sampled, and thus also the reflectance functions, it also shares the same problems as "traditional" mathematical sampling methods. The most fundamental mathematical sampling theorem is without doubt the **Nyquist-Shannon sampling theorem**, which basically states: *"A discrete sampling is a complete representation of a continuous signal if and only if the highest frequency component is less than half the sampling frequency"*. This theorem was originally introduced by H. Nyquist [71] in 1928, and formally proven by C.E. Shannon [93] in 1949.

The sampling frequency referred to in the Nyquist-Shannon theorem is called the *Nyquist frequency*, and the *Nyquist rate* is the sampling rate of this frequency (i.e., rate = 1 / frequency). The frequency components above the Nyquist frequency are subject to *aliasing* when reconstructing the sampled signal. A *band-limited signal* is a signal in which the components above a finite frequency threshold have zero energy. It goes without saying that a band-limited signal can be completely sampled as long as the sampling rate is at least two times this frequency threshold's rate. An important aspect in the sampling of signals is the reconstruction of the sampled signal to (an

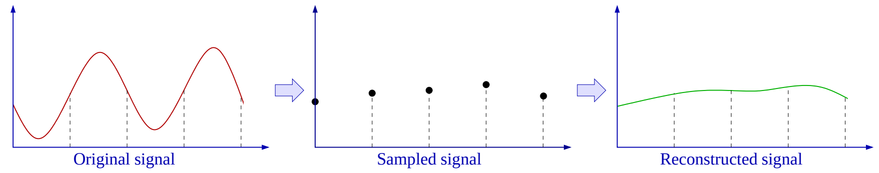


Figure 4.1: An example of aliasing. A signal (red) is sampled at 5 discrete locations. The reconstruction, shown in green, is far from similar to the original signal. The deviation from the original signal is caused by the fact that the sampling rate lies below the Nyquist rate, and consequently the reconstructed signal suffers from aliasing.

approximation of) the original signal. This reconstruction is achieved by some form of interpolation.

Preferably, a band-limited signal is sampled at a sampling rate greater than the Nyquist rate. However, it is possible that there is a limit on the sampling rate due to mechanical reasons, or that the signal is not band-limited. In such a case, aliasing can occur. In figure 4.1 an example of aliasing is shown. Basically, aliasing or undersampling is caused by energy “leaking” from higher frequency components into lower frequency components. There are two solutions to avoid aliasing: *using higher sampling rates*, and *pre-filtering a signal with a band-pass filter*. The first solution is evident, but not always feasible. The idea behind the second solution, is to smooth the to-be-sampled function, such that the effect of the frequency components above the Nyquist frequency is minimized. Note, however, that the reconstructed signal is only an approximation of the original signal, although it can be a perfect reconstruction of the filtered signal.

The adverse effect of aliasing greatly depends on the nature of the sampled signal and on the reconstruction method, i.e., interpolation, used to reconstruct the original signal. In this chapter the effect of sampling reflectance fields, and thus reflectance functions, is detailed. Furthermore, the impact of using different reconstruction methods is investigated.

This chapter is organized as follows. The sampling of reflectance functions with a Light Stage is reviewed, and potential problems are discussed in section 4.2. In section 4.3, and, section 4.4 the bulk of this chapter is discussed: the different reconstruction methods. Finally, section 4.5 concludes this chapter.

4.2 Sampling Reflectance Functions

As mentioned before, the Light Stage is a physical device for sampling the reflectance **field** from a fixed set of sampling directions. More specifically, it samples a collection of reflectance **functions** simultaneously, i.e., for every camera pixel a single reflectance function. This collection of reflectance **functions** together forms a reflectance **field**. In the remainder of this chapter, the focus will lie on the sampling and reconstruction of reflectance **functions**.

Sources of Aliasing. A reflectance function, obtained by means of a Light Stage, is a 2*D* function. If a reflectance function is band-limited, then this function can be completely represented by a limited number of samples. However, this is not the case, as is illustrated in figure 4.2. In this figure, two generic cases are shown. Both scenes consist of a single surface. The reflectance function, for a specific view direction is closely related to the upper-hemisphere of the BRDF of the surface. In figure 4.2.a the material of the surface is diffuse, and thus low frequent. In figure 4.2.b the surface is a perfect mirror, and thus the reflectance function is equal to a single (delta) spike aimed at the perfect reflected direction. In the first case, the reflectance function is obviously low frequent, while in the second case it is very high frequent. This simple case illustrates the fact that reflectance functions are not band-limited, and can, theoretically, span the complete frequency domain.

In general, to avoid aliasing problems, an additional physical constraint is imposed when acquiring reflectance fields with a Light Stage: *the scene/object cannot contain specular, or, near-specular materials*. This also includes refractive materials such as glass. But even if this constraint is met, the reflectance functions are still not band-limited. Another source of high-frequency discontinuities are occlusions (i.e., shadows). This is illustrated in figure 4.3. Although a shadow results in an abrupt change in the intensity values in a reflectance function, it does usually not result in spikes in the reflectance functions. This implies, that although the exact location of the shadow edge can be missed, the shadow effect itself is not missed. In contrast, specular reflection yields spikes in the reflectance functions, which can be missed completely during

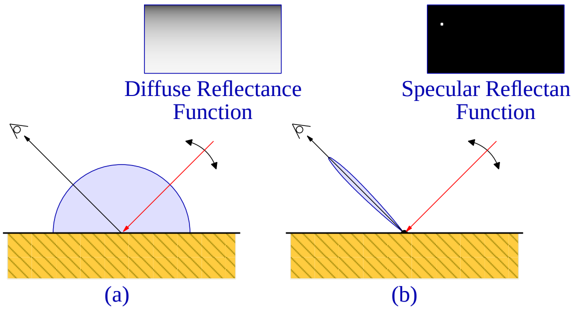


Figure 4.2: Depending on the underlying material, a reflectance function may be band-limited or not. For each example, the view direction is fixed (eye), and the reflectance function is parameterized over incident illumination (red arrow). (a) In case the underlying material exhibits a diffuse reflection behavior, the resulting reflectance function is low frequent as can be seen in the top-right (in a latitude-longitude parameterization). (b) A specular material, on the other hand, yields a reflectance function with a single spike located at the perfect reflected direction. This reflection function is not band-limited.

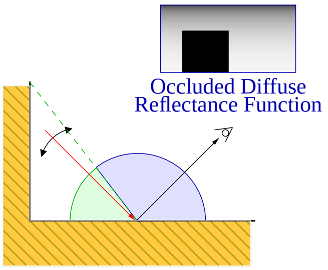


Figure 4.3: Shadows as a source of high frequency discontinuities in reflectance functions.

sampling.

As a side note, related research has been done in the context of precomputed radiance transfer [44, 81, 95]. In this research, the light transport of a synthetic scene is pre-computed in order to visualize it at interactive rates using graphics hardware under complex incident illumination (i.e., environment maps). Seminal work has been performed by Ramamoorthi et al. [81, 82]. In [81], a frequency analysis of diffuse, unoccluded light transport is done. An interesting conclusion for image-based relighting is that a diffuse BRDF can be represented by only the first three spherical harmonics bands, yielding an approximation error of less than 1%. This confirms that a reflectance function of a diffuse reflection is low frequent, and thus band-limited. Further research in this area shows that glossy and specular BRDFs contain much higher frequency components [67]. Ramamoorthi et al. [82] also investigated the frequency response of cast shadows. The main conclusion of this paper is that cast shadows can be efficiently represented as a convolution of a Heaviside function and the unoccluded BRDF. The frequency response of a Heaviside function is zero for even terms, and decays with $1/f$ for odd frequencies. From this follows that the aliasing impact of cast shadows decreases significantly for each increase in sampling resolution.

Classic Solutions to Minimize Aliasing. As stated before, the classic solutions to reduce or even avoid aliasing are: *increasing the sampling rate*, and, *pre-filtering the to-be-sampled function*.

Increasing the sampling rate implies, in the case of the Light Stage, that the number of light source positions needs to be increased. Although this seems a feasible solution at first, there are some practical issues. First of all, acquisition duration will increase considerably. Recording an HDR photograph easily takes a few seconds. Even if a single HDR photograph would only take 1 second to record, then only 3600 light source positions can be handled in an hour. Note that 3600 samples is still a rather low frequent sampling (i.e., similar to the level of detail in a 60×60 environment map). Moreover, there are practical limitations to how accurately a light source can be positioned on the bounding volume. Furthermore, as shown before, reflectance functions are not band-limited, and thus increasing the sampling resolution will not

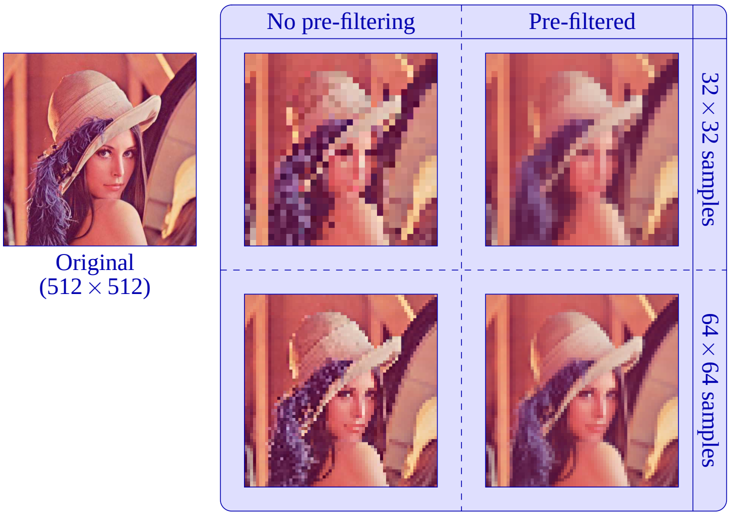


Figure 4.4: Pre-filtering illustrated on the Lena image. A 512 × 512 image is sampled with respectively 32 × 32, and, 64 × 64 samples. The left column is sampled without pre-filtering the image. The right column is pre-filtered before sampling. The pre-filtered sampled images exhibit less aliasing, particularly noticeable on the feather, than the non-filtered images.

eliminate aliasing problems completely.

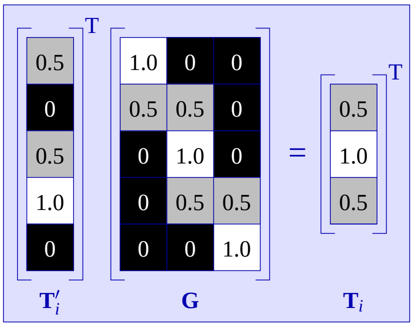
A second solution is to pre-filter the to-be-sampled function. The effect of pre-filtering a function is illustrated in figure 4.4. In case of the Light Stage, this implies that each reflectance function should be convoluted by a band-pass filter such as a Gaussian filter. More formally:

$$\mathbf{T} = \mathbf{T}'\mathbf{G},$$

where $\mathbf{T}$ represents the sampled reflectance field, $\mathbf{T}'$ a high(er) resolution, in the limit continuous, version of the reflectance field. $\mathbf{G}$ is a matrix representing the band-pass filter. This matrix is a banded matrix, in which the band-pass filter (i.e., Gaussian), is repeated for each row centered along the diagonal (see figure 4.5). Inserting this in equation (3.1) yields:

$$\begin{aligned} C &= \mathbf{T}L \\ &= (\mathbf{T}'\mathbf{G})L \\ &= \mathbf{T}'(\mathbf{G}L). \end{aligned} \tag{4.1}$$

Careful examination of the $(\mathbf{G}L)$ factor suggests that pre-filtering of a reflectance field, can be done during acquisition. To understand this, consider the case in which $L = [\delta_{i,j}]_i$, i.e., a single light source at position j. The resulting incident light field



$$\begin{bmatrix}0.5\\0\\0.5\\1.0\\0\end{bmatrix}^{T}\begin{bmatrix}1.0&0&0\\0.5&0.5&0\\0&1.0&0\\0&0.5&0.5\\0&0&1.0\end{bmatrix}=\begin{bmatrix}0.5\\1.0\\0.5\end{bmatrix}^{T}$$

Figure 4.5: An iconic illustration of band-pass filtering of a high resolution reflectance function $\mathbf{T}'_i$ by a linear band-pass filter $\mathbf{G}$. A single vector $\mathbf{T}'_i$ is band-pass filtered, and simultaneously downsampled to $\mathbf{T}_i$. Note that the (linear) band-pass filter $\mathbf{G}$ is not normalized in this example.

in such a case equals a Gaussian fall-off, centered around the j-th position. In other words, if a light source can be constructed, such that its radial fall-off is similar to a Gaussian, then the desired effect is obtained. It is important to note, however, that subsequent filtered light source footprints overlap in space[1]. Practically this means that a Light Stage design in which a single light source is mechanically moved around as in [20] can deliver the desired effect if the spatial fall-off of the light source resembles the desired band-pass filter. Note that the fall-off does not need to be a Gaussian fall-off, any band-pass filter fall-off can be used, although visual quality can degrade. Adapting a Light Stage design as in [34, 59] is less trivial. Mounting a diffuser surface at a small distance from the light sources generates a similar effect. This is illustrated in figure 4.6. Although theoretically possible, little control over the shape of the filter is possible, making this method less suitable in a practical setup.

Upsampling operator. The classic solutions, all required changes to the Light Stage setup. It is interesting to note that, even when a non-adapted setup is used, significant better relighting results can be obtained by using a suitable upsampling operator on the sampled reflectance functions.

First consider the relation between L, a low resolution incident light field, and L', a higher resolution incident light field:

$$\begin{aligned} L &= \mathcal{S}_L\left(\mathcal{R}_{L,L'}\left(\mathcal{S}_{L'}{}^{-1}(L')\right)\right), \\ \mathcal{S}_L{}^{-1}(L) &= \mathcal{R}_{L,L'}\left(\mathcal{S}_{L'}{}^{-1}(L')\right), \end{aligned}$$

where $\mathcal{R}_{L,L'}(\cdot)$ is a resample operator which downsamples the high resolution discretized incident light field to a lower resolution version. This relation is not reversible, because:

[1] If the footprints do not overlap, then certain frequencies are not filtered out, resulting in aliasing. Taking a 'point' sample can be seen as an extreme example of this.

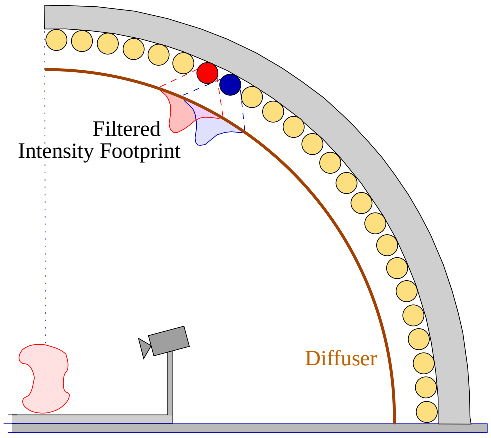


Figure 4.6: Pre-filtering the light sources in a Light Stage setup. Light emitted from a light source is filtered by the diffuser. Due to the angular spread of the light sources, the resulting footprint can overlap other light sources' footprints (as illustrated by the red and blue marked light sources).

$$\begin{aligned} L' &\approx \mathcal{S}_{L'}\left(\widetilde{\mathcal{R}_{L,L'}}^{-1}\left(\mathcal{R}_{L,L'}\left(\mathcal{S}_{L'}{}^{-1}(L')\right)\right)\right) && (4.2) \\ &= \mathcal{S}_{L'}\left(\widetilde{\mathcal{R}_{L,L'}}^{-1}\left(\mathcal{S}_{L}{}^{-1}(L)\right)\right). && (4.3) \end{aligned}$$

Note that the inverse operator $\mathcal{S}_{L,L'}\left(\widetilde{\mathcal{R}_{L,L'}}^{-1}\left(\mathcal{S}_{L}{}^{-1}(\cdot)\right)\right)$ upsamples an incident light field vector of length l to a high resolution incident light field vector (i.e., L'). We denote this upsampling operator as $\mathcal{U}(\cdot)$. The resampling operator, can in general not be undone exactly, hence the $\approx$ in equation (4.2)).

Now, reconsider equation (3.3):

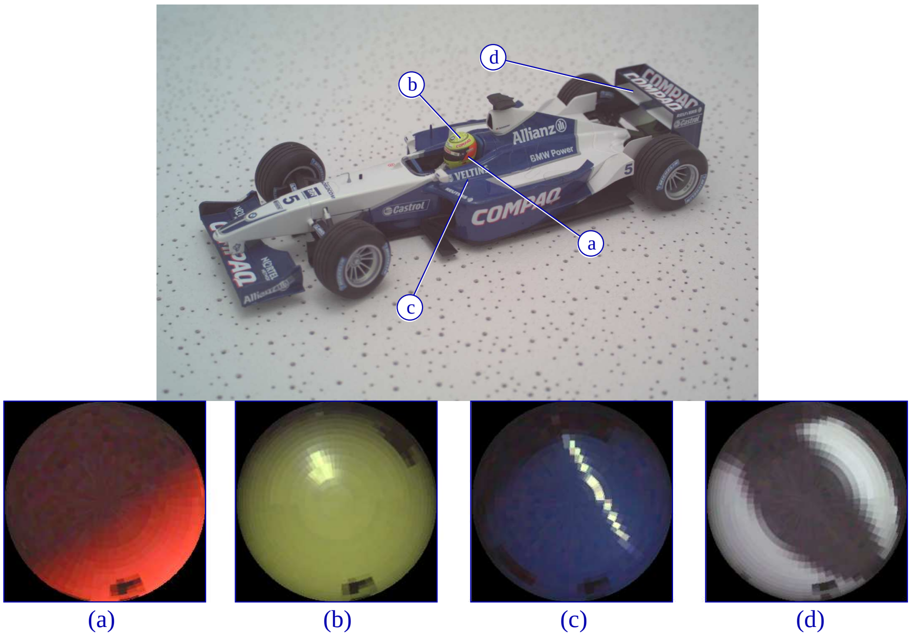
$$c_i = \mathbf{T}_i \cdot L.$$

The reflectance function $\mathbf{T}_i$ is also a vector of length l. Thus, theoretically, the upsampling operator can also be applied to it. Plugging this in equation (3.3):

$$c_i = \mathcal{U}(\mathbf{T}_i) \cdot \mathcal{U}(L), \tag{4.4}$$

Combining equations (4.3) and (4.4) results in:

$$c_i \approx \mathcal{U}(\mathbf{T}_i) \cdot L'. \tag{4.5}$$



(a) (b) (c) (d)

Figure 4.7: A Formula One toy car. Also depicted are the reflectance functions of four selected pixels acquired by sampling the reflectance field from 1280 lighting directions. Reflectance function (a) represents a mainly diffuse material, reflectance function (b) a fairly glossy material, reflectance function (c) a specular material, and reflectance function (d) contains complex occlusion features.

Examining equation (4.5) reveals some interesting things. First thing to notice is the approximation sign ($\approx$). It is important to realize that this is an approximation to the undersampled original reflectance function, which suffers from aliasing, and thus also contains artefacts caused by aliasing. The question now is, whether this “approximation” also contains these artefacts. Second, the high resolution incident illumination L' is not resampled. Resampling implies loss of information, thus equation (4.5) can potentially generate more accurate relit images since the incident illumination contains more information.

Nyquist’s theorem cannot be circumvented; if the sampling rate is below the Nyquist rate, information is lost, and aliasing will happen. No matter how clever the reconstruction operator $\mathcal{U}(\cdot)$, lost information is never regained. However, additional constraints can be placed on the reconstructed reflectance functions, that can ensure visually more pleasing results. Examples of such additional constraints are: smoothness of the reconstructed signal, non-negativity of the reconstructed signal, minimized measurement noise, compact representation, ... In the following section a number of reconstruction/interpolation techniques for reflectance functions are studied, and their results are thoroughly analyzed.

4.3 Reconstruction

The upsampling operator $\mathcal{U}(\cdot)$ is not uniquely defined. In this section a number of possible reconstruction operators are studied for reflectance functions acquired by means of a Light Stage device. A result of each of the reconstruction operators is illustrated on the reflectance functions of four selected pixels of the Formula One toy car example used in chapter 2. In this chapter we will use the Nusselt embedding (i.e., an orthogonal projection of a hemisphere to a circle) to illustrate the resulting upsampled reflectance functions. This parameterization is only used to visualize the results, not to compute them.

In figure 4.7 the Formula One toy car is shown, together with the four selected pixels and their corresponding reflectance functions. The reflectance functions shown in this figure are acquired by sampling 1280 illumination directions. To better show the effects of the reconstruction method, the upsampled reflectance functions in the following subsections will only use $\frac{1}{4}$ of the samples (i.e., 320), regularly selected from the 1280 samples.

4.3.1 Zero-order Hold

Zero-order hold interpolation is the most commonly applied interpolation method, and is basically a piece-wise constant reconstruction of a reflectance function. The Light Stage samples light source positions on a sphere, and thus the reconstruction of the reflectance function occurs preferably on a sphere to avoid reparameterization artefacts. Because the sampling of illumination positions is not always uniform on the sphere, the area of influence of each sampled location is determined first by constructing a spherical Voronoi diagram. The angle between two points on the sphere and the center point, is a valid measure of distance on a sphere. This distance can be efficiently computed by taking the dot product. Similar spherical Voronoi diagrams are also used in the context of image-based relighting in Masselus et al. [57]. Once the area of influence of each sample is known, all points in this area of influence are assigned the measured sample intensity. Note that because the angular Voronoi diagram only depends on the sampled locations, and these locations are the same for every reflectance function, the Voronoi diagram needs to be computed only once.

In figure 4.8 the reconstructed reflectance functions of the four selected pixels are shown. Each reflectance function is interpolated from 320 samples. The reconstruction error is visually observable: the reconstructed reflectance functions in figure 4.8 look less smooth than the reference reflectance functions in figure 4.7 which are captured with four times as many samples. Note, that in the case of the Light Stage setup, the light sources are positioned regularly in the latitude-longitude domain. The computation of the angular Voronoi diagram is in this case greatly simplified. Each Voronoi cell is a square in the latitude-longitude parameterization. The technique described in Debevec at al. [20], basically employs a zero-order hold interpolation.

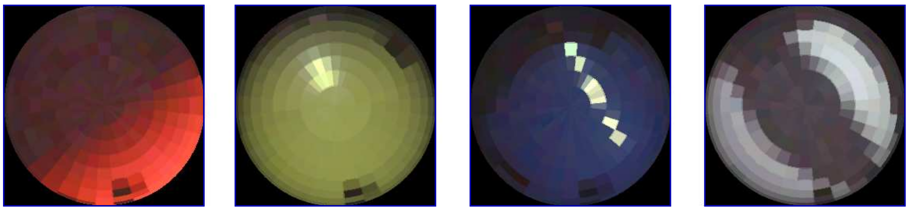


Figure 4.8: The reflectance functions of four selected pixels, reconstructed using the zero-order hold reconstruction method. For each reflectance function only 320 acquired samples are used.

4.3.2 Linear Interpolation

The zero-order hold reconstruction technique does not yield smooth results. In order to improve the smoothness of the reconstructions, linear interpolation can be used. For any given point on the sphere, its linearly interpolated value is determined by the spherical barycentric weighting of the corner points of the spherical triangle containing this point. To ensure a good interpolation, the bounding spherical triangle should be as small as possible, and as regular in shape as possible. A triangulation which satisfies these constraints is a Delaunay triangulation. In our case, a spherical Delaunay triangulation is required, which is the dual of the spherical Voronoi diagram. In Pang et al. [55] an algorithm for constructing Delaunay diagrams on a sphere is detailed. As with the zero-order hold Voronoi diagram, the Delaunay triangulation only depends on the light source positions, and thus can be precomputed.

In figure 4.9 the linearly interpolated reflectance functions of the four selected pixels are shown. These reconstructed reflectance functions are visually much smoother than the piece-wise constant reconstructions. Again, in case a Light Stage setup is used, the samples are regularly spaced in the latitude-longitude domain, and the computation of the Delaunay triangulation becomes trivial.

4.3.3 Distance-weighted Interpolation

Linear interpolation based on Delaunay triangulation, only uses 3 samples for the reconstruction of each point of the reflectance function. Given the fact that reflectance

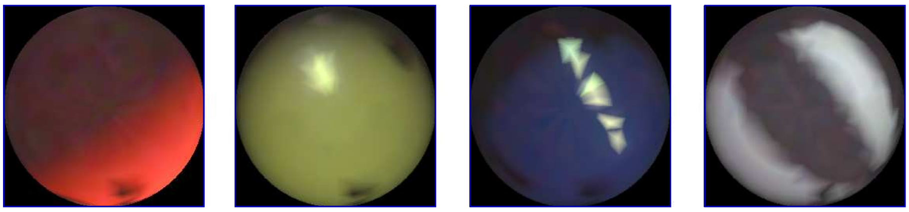


Figure 4.9: The reflectance functions of four selected pixels, reconstructed using linear interpolation on a sphere based on the Delaunay triangulation of the light source positions.

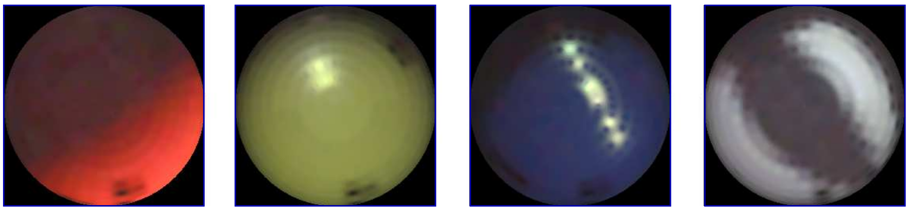


Figure 4.10: The reflectance functions of four selected pixels, reconstructed using inverse distance-weighted interpolation on a sphere. Only the 32 nearest samples are used to interpolate an intensity value.

functions are not band-limited, high frequency "noise" can distort one or more of the three closest samples. In this regard it is sensible to include more than 3 samples. Inverse distance weighted interpolation, also called Sheppard's method, uses all samples, but varies the weights of each sample according to the distance to the reconstructed point. The weights w_i of a point $\mathbf{x}$ with respect to the i-th sample can be defined by:

$$w_i(\mathbf{x}) = \frac{d_i^{-r}(\mathbf{x})}{\sum_j d_j^{-r}(\mathbf{x})},$$

where $d_i(\mathbf{x})$ is the distance function from a point $\mathbf{x}$ to the i-th sample. This distance function can be enhanced by selecting a maximum influence radius, or by imposing a maximum sample limit. The factor r determines the fall-off in function of the distance. In our implementation we set $r = 2$.

In figure 4.10 the inverse distance-weighted interpolated reflectance functions of the four selected pixels are shown. These reconstructed reflectance functions are computed with $r = 2$ and the distance measure is limited to include only the 32 nearest samples. In this case, the weighting only depends on the sampled directions, and thus can also be precomputed.

4.3.4 Spherical Harmonics Interpolation

The Nyquist-Shannon theorem [71, 93], relates the sampling rate to the highest frequency of a signal that can be captured and reconstructed. The frequency response or spectrum of a signal can be computed by decomposing the signal using the Fourier transform (i.e., a decomposition into Fourier basis functions). Spherical harmonics are a natural extension of Fourier basis functions to a spherical domain (see page 675 in [32], volume 2 for a brief overview).

Interpolation can be achieved by fitting spherical harmonics through the sampled data. In case only the hemisphere is captured (due to the Light Stage used in this work), a complete sphere of samples is created by mirroring the data to the other

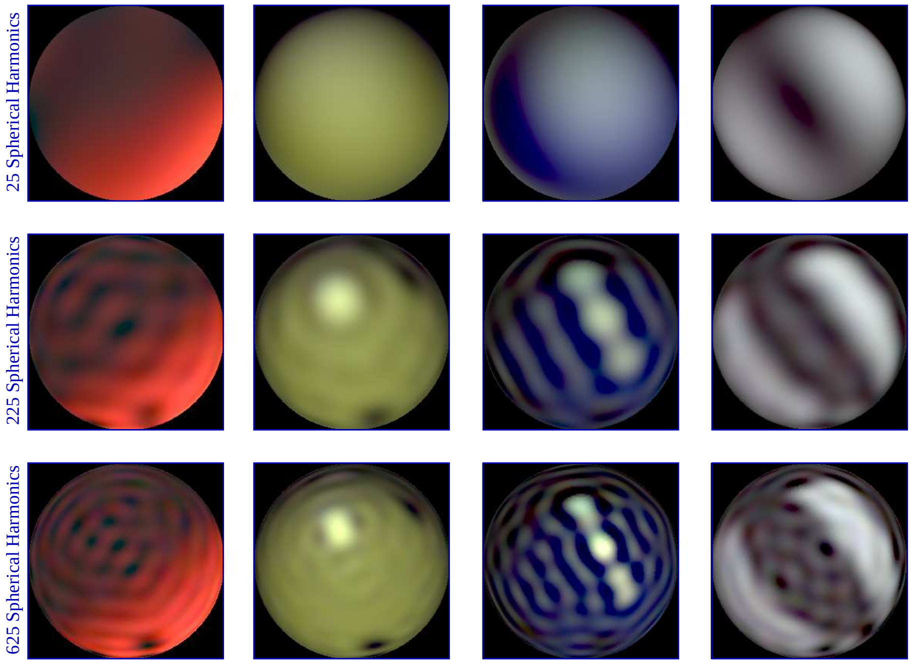


Figure 4.11: The reconstructed reflectance functions, interpolated using 25, 225, and 625 spherical harmonics respectively, of the four selected pixels. The reconstructed reflectance functions exhibit a significant degree of aliasing artefacts.

hemisphere.

Wong et al. [114] applied this technique on sampled reflectance functions of synthetic scenes and used 16, and, 25 spherical harmonics coefficients. This corresponds to four to five bands of the spherical harmonics basis functions respectively. In figure 4.11, first row, the sampled reflectance functions are reconstructed using the 25 first spherical harmonic basis functions. The resulting reflectance function reconstructions are very smooth, but much of the fine detail (high frequency content) has been lost.

In figure 4.11, the reflectance functions reconstructed with 25, 225, and 625 spherical harmonics basis functions are depicted. The reconstructed reflectance functions are able to capture all features to some extent, but suffer from severe Gibbs ringing or aliasing artefacts. Increasing the number of spherical harmonics bands does not make much sense, since the sampling frequency is already reached in the case of the lower row (625 spherical harmonics functions or 25 bands). Spherical harmonic interpolation can result in negative values, which are clamped to zero in figure 4.11.

4.3.5 Wavelet Interpolation

One of the reasons spherical harmonic interpolation does not work well, is that the Light Stage samples in the spatial domain, as opposed to the frequency domain. Therefore, interpolation is more successful in the spatial domain (i.e., zero-order hold, linear interpolation, ...). However, spherical harmonic interpolation generates much smoother results. The question arises if better interpolation results (i.e., smoother), but without the aliasing artefacts of spherical harmonic interpolation can be achieved by using a higher order interpolation scheme. In this regard we consider wavelet interpolation. Wavelets are a versatile mathematical tool, which brought a revolution in many scientific domains, including computer graphics. In this thesis, wavelets will be used on numerous occasions, and a brief introduction can be found in appendix A.

A sampled signal can be reconstructed (i.e., upsampled) using wavelets, by repeatedly inserting additional samples halfway between two sample points. The magnitude of each new sample point is characterized by the scale function of the wavelet. Basically, this comes down to using the scale function's weights as interpolation weights. Another way of looking at wavelet interpolation is to first express the sampled signal in the wavelet domain, and subsequently insert zero magnitude high resolution wavelet coefficients (high frequency features). By transforming the wavelet signal back to the spatial domain, the final upsampled signal is obtained.

A logical choice would be to use spherical wavelets [89]. The highly irregularly spaced samples in the spherical domain, however, are a disadvantage, and require an extra resampling step. In our case, due to the construction of the Light Stage, the recorded samples are regularly spaced in the latitude-longitude parameterization. Therefore, the (normal) wavelet transform is used in the latitude-longitude parameterization. An additional advantage is that common wavelet implementations can be used without much trouble. A disadvantage is that the reparameterization introduces some additional artefacts. However, as will be shown in section 4.4, the impact of these artefacts is minimal.

In figure 4.12 wavelet interpolation of the four reflectance functions is illustrated using three different types of wavelets. The first wavelet used is the popular Haar wavelet. The second wavelet is the $5/3$ LeGall, or Integer $5/3$, which is the shortest symmetrical biorthogonal wavelet with two vanishing moments. Its scaling function is a linear B-Spline. The third wavelet is the well known $9/7$ Daubechies wavelet, which is the shortest symmetrical biorthogonal wavelet of order four, and is by construction a cubic B-Spline. The latter two wavelets are part of the JPEG2000 standard, and are therefore widely implemented. See [105] for a detailed overview on these two wavelets. Normally, these wavelets are mirrored around image boundaries. However, in our implementation we opted for repeating the signal on vertical boundaries, since this fits better to the original spherical domain. We still mirror the wavelets on horizontal boundaries.

Finally, we would like to point out the similarity of the wavelet interpolation meth-

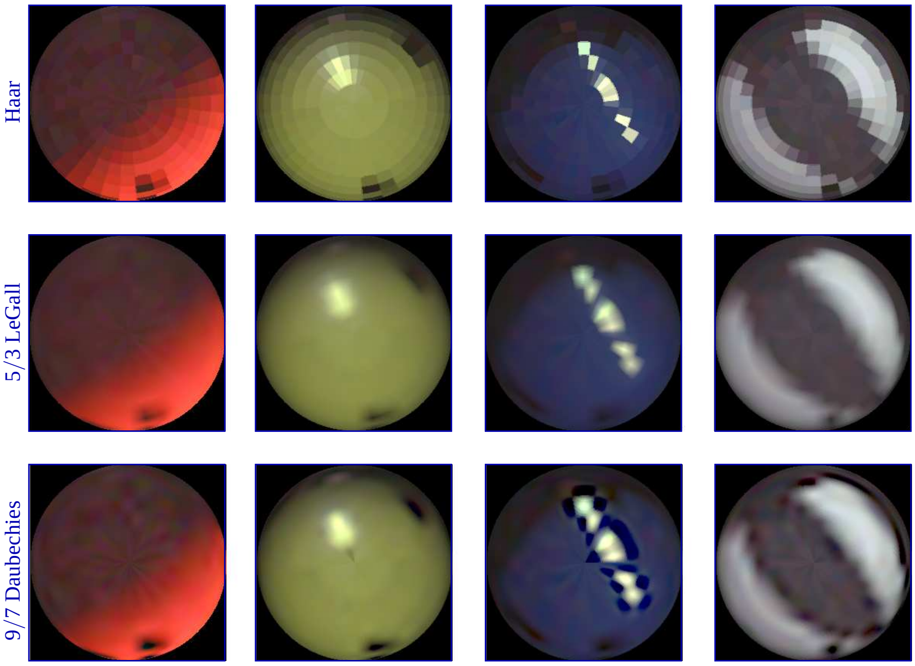


Figure 4.12: The reconstructed reflectance functions, interpolated using the Haar, 5/3 LeGall, and, 9/7 Daubechies wavelets respectively, of the four selected pixels.

ods and some of the previously discussed techniques in case of a regular sampling in the latitude-longitude domain. The Haar wavelet interpolation is identical to the zero-order hold reconstruction. Both methods create a piece-wise constant approximation of the signal. The 5/3 LeGall wavelet interpolation, is similar to the linear interpolation method. Both create a linear interpolation between samples. However, the linear interpolation is computed in the spherical domain, and can result in a slightly different interpolation results near the poles.

4.3.6 Multi-level B-Spline Interpolation

Bicubic B-Splines can be used to create a continuous function. These functions are a good trade-off between smoothness (they are C^2 continuous) and the ability to represent the features in the captured data. B-Splines are not easily defined on a sphere. Similar to wavelets, the sampled data is first reparameterized to a planar domain. in this case the sampled data is represented in the paraboloid map parameterization [38]. This representation offers a continuous projection of a hemisphere, in which the boundaries are identical as on a hemisphere, and the pole is defined in a single point. Furthermore, the solid angle to projected area ratio is close to constant. A disadvantage is the irregular spread of the samples over the domain. However, unlike wavelet interpolation this is not a major problem.

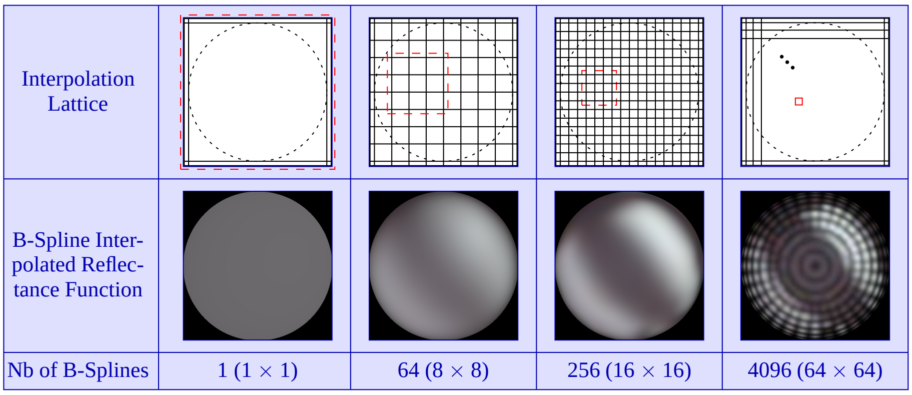


Figure 4.13: Bicubic B-Spline interpolation is illustrated with different resolutions of control points. Each bicubic B-Spline is defined on a 4 × 4 sub-grid. The relative size of each sub-grid is indicated in red.

A bicubic B-Spline can be fitted to the data by creating a 4×4 grid of control points on the projected data, as can be seen in the first column of figure 4.13. The interpolated reflectance function is poorly reconstructed because a single bicubic B-Spline is unable to represent all details. Similar to using a single biquadric polynomial, as is done in Polynomial Texture Maps [54], the obtained interpolation is too smooth. An obvious solution would be to reconstruct the reflectance function using a set of independent B-Splines, each defined on different 4×4 grids of control points, that are defined over a lattice on the projected data. Figure 4.13 demonstrates the influence of the resolution of the lattice and the resulting reflectance function. Using only a few B-Splines results in a good global fit of the data, but with almost no local detail (figure 4.13, first three columns), while using more B-Splines produces a reflectance function with a good local fit but lacking global smoothness (figure 4.13, last column). A similar problem was also noted by Lee et al. [48].

Multilevel B-Splines, introduced by Lee et al. [48], allow to fit a smooth approximation through the projected samples without the problems described above. Multilevel B-Spline interpolation is a hierarchical method that first tries to fit a set of globally smooth B-Splines through the sampled data, each defined on a distinct 4×4 grid of control points with a large coverage. In each successive step the number of control points in the grid of each B-Spline is doubled in each direction and a new set of B-Splines is created on the halved 4×4 grids of control points. The new B-Splines on the smaller grids are fitted through the difference of the sum of the already computed B-Splines and the measured sample values. The method is illustrated in figure 4.14. The hierarchy of B-Spline sets can then be reduced to a single set of B-Splines defined on the 4×4 grids of control points with the smallest coverage used (i.e., highest hierarchy level).

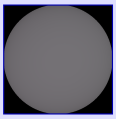
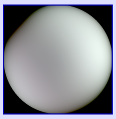
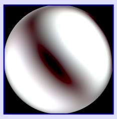
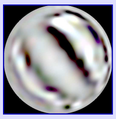
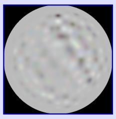
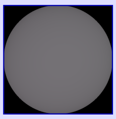
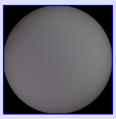
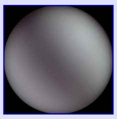
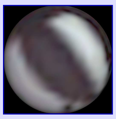
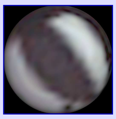
Level	1	2	3	4	5	6
Diff. Samples and Reconstructed Reflectance Function Previous Level						
Reconstructed Reflectance Function						
Nb of B-Splines	4 (2 × 2)	16 (4 × 4)	64 (8 × 8)	256 (16 × 16)	1024 (32 × 32)	4096 (64 × 64)

Figure 4.14: An illustration of the different hierarchy levels of multi-level B-Spline interpolation. For each level, the reconstruction of the samples minus the reconstructed results of the previous levels are shown in the top row. In the lower row, the complete reconstruction is shown.

In figure 4.15 four reflectance functions, reconstructed using the multi-level technique are shown. Starting from four B-Splines in the lowest level upto 64×64 B-Splines, in level 6, are used in the multi-level hierarchy. The number of levels in the hierarchy was empirically determined on the magnitude of differences between the measured values and the values of the sampled directions in the already constructed reflectance function. Using six levels allows to fit the data values, while not fitting noise on the data. This method results in a set of 64×64 B-Splines.

4.4 Discussion

In this section, the different upsampling methods of the previous section are compared. A thorough error analysis is detailed in subsection 4.4.1. In subsection 4.4.2 the similarities and differences between upsampling the reflectance data versus downsampling the incident illumination are discussed.

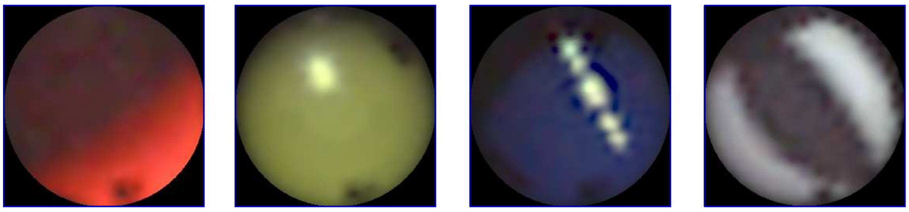


Figure 4.15: Multi-level B-Spline interpolation illustrated on the sampled data. These results are obtained by using 6 levels. Level 1 contains 4 B-Splines (as illustrated in figure 4.14), and level 6 contains 64×64 B-Splines.

4.4.1 Error Analysis

Numerical Error Analysis. In order to compare the upsample techniques of section 4.2, the error of the reconstructed signal versus the real continuous signal needs to be computed. However, the real continuous reflectance function is not known. Therefore, we resort to computing an approximate error, by sampling both the real continuous and the reconstructed reflectance functions at a sampling rate higher than the original sampling rate, and compute an approximate error on the sampled values. In other words, the reflectance function is sampled at a specific sampling rate (determined by the acquisition device). Next, a reflectance function is reconstructed from a subset of the recorded samples (e.g., 25% of the samples). Finally, the relative error is computed between the full set of samples and the reconstructed reflectance functions. More formally:

$$e_i = \frac{||\mathcal{U}(\mathbf{T}_i) - \mathbf{T'}_i||_2}{||\mathbf{T'}_i||_2},$$

where $\mathbf{T}$ is a lower resolution reflectance field (a subset of the samples), and $\mathbf{T}'$ is a high resolution reference reflectance field. The division by the total energy in the reference reflectance function $\mathbf{T'}_i$ ensures differences in albedo between different reflectance functions are eliminated, such that the errors over different reflect reflectance functions can be compared in a sensible way.

The quality of each of the reconstruction techniques is tested on the Formula One toy car scene. Each pixel's reflectance function is upsampled from 320 samples (32×10), and compared to 1280 measured samples (64×20). However, the error on a reflectance function will greatly vary depending not only on the upsampling method used, but also on the underlying material properties. Most of the reconstruction techniques will perform well if a pixel represents a diffuse unoccluded surface, thus a low frequency function, and consequently result in a low error. However, the error will be significant for some reconstruction methods when the reflectance function features a highlight, or complex self-shadowing.

Instead of comparing errors of individual reflectance functions, the average error over a set of pixels with similar occlusion and material properties is computed. We compare these errors for different reconstruction techniques. We distinguish four sets of pixels:

- a set of pixels representing the complete Formula One toy car (excluding the ground plate) (figure 4.16.a),
- pixels representing diffuse unoccluded surfaces (figure 4.16.b),
- pixels representing diffuse surfaces with complex self-shadowing (figure 4.16.c),
- and, pixels located on glossy, and specular materials (figure 4.16.d).

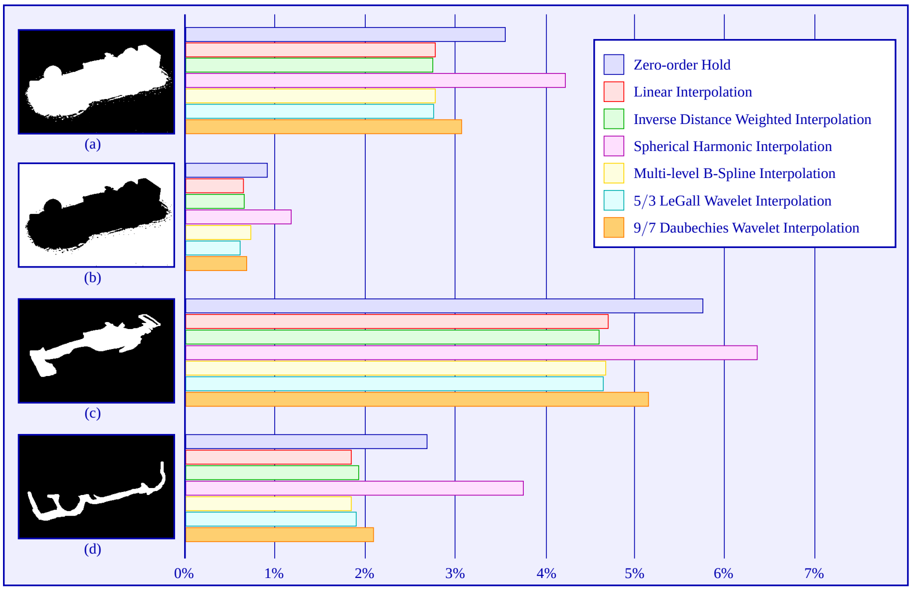


Figure 4.16: The average error of different reconstruction methods over four selected sets of pixels for the Formula One toy car. The four pixel sets represent (from top to bottom): the complete object excluding underground, the diffuse unoccluded underground, glossy and specular materials, and the occluded diffuse underground.

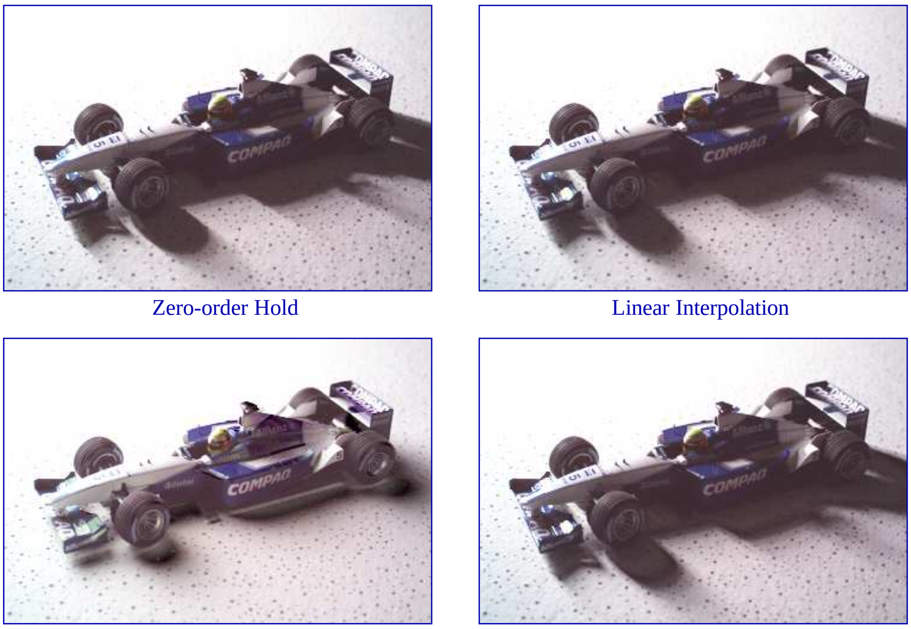
The average relative errors of these four sets are plotted in figure 4.16. The four sets are visualized as false color images in which the relevant pixels are highlighted.

In general, the errors for unoccluded diffuse reflectance functions are low, while reflectance functions featuring high frequency details result in larger errors. Ramamoorthi and Hanrahan [81] noted that diffuse unblocked reflectance functions can be represented by the first three spherical harmonics bands, yielding an error of less than 1%. From the graph in figure 4.16, a similar conclusion can be drawn. Note that the error on the spherical harmonic interpolation uses 5 spherical harmonics bands, but on noisy real data (including occlusion by the camera).

In terms of error, the linear interpolation schemes (including 5/3 LeGall wavelet interpolation), and the multi-level B-Spline interpolation produced upsampled reflectance functions most faithfully. 9/7 Daubechies interpolation also yields good results, albeit with a slightly higher error. The standard deviation on the calculated errors is small for pixels representing diffuse or occluded surfaces. Pixels representing a specular surface produced a large standard deviation which was expected due to the high frequency features in the reflectance functions.

Visual Error Analysis. Although the above error comparisons give a good indication of the accuracy of the different reconstruction methods, it does not guarantee good relit images. In other words, it is the effect of incident illumination on the object that is most important, not the error on the upsampled reflectance function itself. In figure 4.17, the Formula One toy car is illuminated by a vertical sliver of light. Using a sliver of illumination results in images containing both low frequency illumination features such as long soft shadows (along the direction of the line of illumination), and high frequency illumination features such as very short soft shadows and highlights (orthogonal to the line). 1280 sampled directions are used when upsampling the reflectance functions to the resolution of the incident illumination.

Except for spherical harmonic interpolation, the still images in figure 4.17 all look very similar. However, when animating the incident illumination (e.g., rotating the line of light by shifting it along the longitudinal direction), significant differences can be observed in the series of animated relit images. Figure 4.18 illustrates the effect of animated incident illumination by stacking a single vertical scanline of the relit images as columns, in a new image. Thus, the resulting image contains in a single horizontal scanline, the evolution of a single pixel over time. The animated incident illumination in figure 4.18 is a sliver of light rotating over the hemisphere. The sliver of light is oriented along the latitudinal direction.



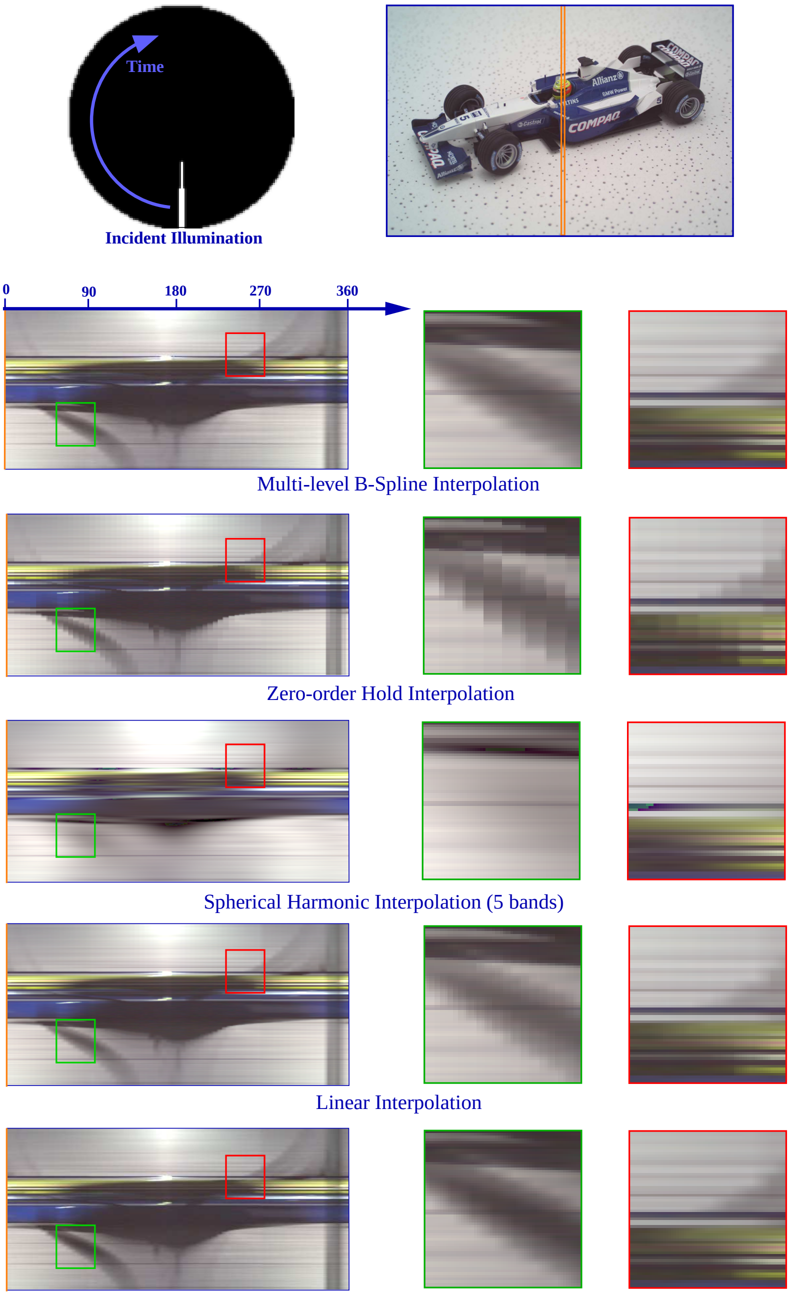
Zero-order Hold

Linear Interpolation

Spherical Harmonic Interpolation

Multi-level B-Spline Interpolation

Figure 4.17: The Formula One toy car, upsampled using different reconstruction techniques, and relit using incident illumination consisting of a single (latitudinal) sliver of light.



Multi-level B-Spline Interpolation

Zero-order Hold Interpolation

Spherical Harmonic Interpolation (5 bands)

Linear Interpolation

9/7 Daubechies Wavelet Interpolation

Figure 4.18: Illustrating the quality of the various reconstruction techniques under time-varying incident illumination. A selected column from the Formula One toy car example, is relit with time-varying incident illumination. The different columns are stacked into a single image, in which each scanline represents a relit pixel value over time.

Using a zero-order hold interpolation still results in a pleasing image, however, the shadows and highlights move in a jittered way when the illumination is changed, visible as blocky features in figure 4.18. Spherical harmonic interpolation does not result in a satisfying image either. The shadows, especially close to the object, are distorted, and the highlights are completely lost. However, this method requires very few coefficients to represent the data which is of importance for real time relighting. Linear interpolation, multi-level B-Spline interpolation, and wavelet interpolation ($5/3$ LeGall and $9/7$ Daubechies) techniques deliver good visual results. The shadows are faithfully recreated, as are the highlights. Multilevel B-Spline interpolation, yields slightly smoother relit animated images. However, this observation is highly subjective.

4.4.2 Upsampling versus Downsampling

When introducing the zero-order hold interpolation, the similarity to downsampling the environment maps as in [20] was noted, hinting at a broader relationship between upsampling reflectance functions and downsampling incident illumination. Furthermore, for a number of reconstruction techniques, the interpolation structure could be precomputed. This raises the question whether this precomputation can be applied to other upsample methods.

First, write the upsampling operator $\mathcal{U}(\cdot)$ as a matrix multiplication with an “upsample”-matrix **U**:

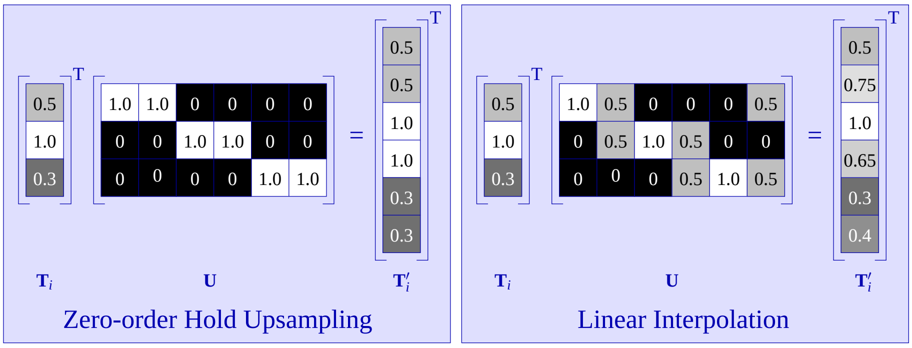
$$\mathcal{U}(\mathbf{T}) = \mathbf{T}\mathbf{U}. \tag{4.6}$$

The upsample-matrix **U**, interpolates each row in **T**. In figure 4.19, zero-order hold and linear interpolation are illustrated on a single vector, that is upsampled to a vector with double resolution. In fact, most of interpolation methods, introduced above, can be represented as a matrix multiplication. In general, any “linear” upsample method can be represented in this way. Interpolation methods which depend non-linearly on the sampled values cannot be represented by a simple matrix multiplication.

Applying equation (4.6) to image-based relighting (equation (3.1)), yields:

$$\begin{aligned} C' &= \mathbf{T}'L' \\ &= (\mathbf{T}\mathbf{U})L' \\ &= \mathbf{T}(\mathbf{U}L') \\ &= \mathbf{T}L, \end{aligned} \tag{4.7}$$

where $\mathbf{T}'$ is the upsampled version of $\mathbf{T}$ and the incident illumination L' is a high resolution version of $L = \mathbf{U}L'$. Of interest is the fact that both $(\mathbf{T}'L')$ and $(\mathbf{T}L)$ yield exactly



$$\begin{bmatrix} 0.5 \\ 1.0 \\ 0.3 \end{bmatrix}^{T} \begin{bmatrix} 1.0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 1.0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \\ 1.0 \\ 1.0 \\ 0.3 \\ 0.3 \end{bmatrix}^{T}$$

Zero-order Hold Upsampling

$$\begin{bmatrix} 0.5 \\ 1.0 \\ 0.3 \end{bmatrix}^{T} \begin{bmatrix} 1.0 & 0.5 & 0 & 0 & 0 & 0.5 \\ 0 & 0.5 & 1.0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 1.0 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.75 \\ 1.0 \\ 0.65 \\ 0.3 \\ 0.4 \end{bmatrix}^{T}$$

Linear Interpolation

Figure 4.19: An iconic illustration of two examples of a zero-order hold upsample-matrix, and a linear interpolation upsample-matrix. Both matrices upsample a vector of length 3 to a vector of length 6. Note that the linear interpolation is circular, i.e., wraps around the bottom and top edges.

the same relit image. In other words, pre-multiplying with the upsample-matrix is actually a downsampling operator, and can be included in the resample operator $\mathcal{R}_L(\cdot)$.

Equation (4.7) dictates that, in case the upsample operator depends linearly on the sample values (which is true in most cases), an equivalent downsampling operator can be defined on the incident illumination resulting in exactly the same relit images. The only difference between upsampling the reflectance functions and downsampling the incident illumination, is the number of elements which need to be taken into account when computing the resulting relit image. In this regard, it is more efficient to downsample the incident illumination once beforehand, and then compute the relit image directly from the sampled data. The interpolation error analysis of subsection 4.4.1 is also relevant for downscaling. Taking the error analysis in account, linear interpolation (or equivalently $5/3$ LeGall wavelet interpolation) seems to be the best candidate because the downsampling operator can be computed efficiently. Multi-level B-Spline interpolation yielded perceptually smoother results under time-varying incident illumination, but computing the downsampling matrix is, in this case, far from obvious.

Finally, equations (4.1) and (4.7) look similar, but the difference between both equations lies in the size of the matrices **G** and **U** respectively. The matrix **G** is higher than it is wide, whereas **U** is wider than it is high. The effect is that **G** reduces the frequency components of **T**, in other words, downsamples it, whereas **U** upsamples **T**, and vice versa for the incident illumination. In effect, **G** removes high frequency components from the light transport matrix **T**, whereas **U** removes these components from the incident illumination L. Additionally, **U** also upsamples the reflectance functions, for which a thorough error analysis can be done. However, it is important to realize that both methods are not mutually exclusive. It is entirely possible, to capture a band-limited reflectance field by pre-filtering the illumination during acquisition, and during relighting upsample the band-limited reflectance field.

4.5 Conclusion

In this chapter, the upsampling of reflectance functions was discussed in detail. A careful error analysis pointed out that linear interpolation, $5/3$ LeGall wavelet interpolation, and multi-level B-Spline interpolation perform better in terms of reconstruction error, and visual smoothness. Additionally, it has been shown that for most upsampling operators an equivalent downsampling operator on the incident illumination can be defined, which yields identical relit results.

5

Compact Representation of Reflectance Functions

In this chapter the representation of reflectance functions in different bases, especially a wavelet basis, is studied. It will be shown, that representing reflectance functions in a different basis can have significant advantages. In the next chapter these properties are exploited to develop two efficient acquisition algorithms. The ideas presented in this chapter were originally presented in [59], with some refinements in [74] and [75].

5.1 Introduction

Until now, the light transport matrix $\mathbf{T}$ was constructed by copying the different acquired HDR photographs, each corresponding to an illumination condition, into different columns $\mathbf{T}_{\cdot,i}$ of the transport matrix $\mathbf{T}$. This implies that a reflectance function is expressed in the same domain (in a mathematical sense) as the incident illumination (i.e., the spatial domain). In section 4.1, we studied the frequency response of reflectance functions in order to determine if these reflectance functions are band-limited. In fact, studying the frequency response of a function is analogous to a change of basis, more specifically to a Fourier basis.

In this chapter, the effect of expressing reflectance functions in a different basis is investigated (section 5.2). A change of basis, can help to better understand the properties of the reflectance field (e.g., frequency content), can yield a more compact representation (section 5.3) or faster relighting computations (section 5.4), and, can eventually lead to a faster acquisition process (chapter 6).

The reader unfamiliar with basis functions, and in particularly wavelet basis functions, is referred to appendix A for a concise introduction to basis transformations and wavelets.

5.2 Change of Basis

The general goal is to express each reflectance function in a new basis. In other words, to change the basis of each row $\mathbf{T}_i$ of the transport matrix $\mathbf{T}$. Given a set of basis functions $\mathbf{B} = [\mathbf{B}_{\cdot,1}|\ldots|\mathbf{B}_{\cdot,l}]$, a reflectance function $\mathcal{S}^{-1}(\mathbf{T}_i)$ (thus not serialized) can be expressed in this basis by:

$$(\mathcal{S}^{-1}(\mathbf{T}_i))_{\mathbf{B}} = \langle\, \mathcal{S}^{-1}(\mathbf{T}_i) \mid \mathbf{B}\, \rangle.$$

Serializing this transformed reflectance function, yields:

$$\begin{aligned} (\mathbf{T}_i)_{\mathbf{B}} &= \mathcal{S}\left(\langle\, \mathcal{S}^{-1}(\mathbf{T}_i) \mid \mathbf{B}\, \rangle\right) \\ &= \langle\, \mathbf{T}_i \mid [\mathcal{S}(\mathbf{B}_{\cdot,1})|\ldots|\mathcal{S}(\mathbf{B}_{\cdot,l})]\, \rangle. \end{aligned}$$

It is important to note that $[\mathcal{S}(\mathbf{B}_{\cdot,1})\,|\ldots|\,\mathcal{S}(\mathbf{B}_{\cdot,l})]$ is still a valid basis, even after serialization. Spatially neighboring elements, however, may not be neighboring in the serialized version anymore. But the general properties, even spatial ones, of this basis are still maintained. By denoting $[\mathcal{S}(\mathbf{B}_{\cdot,1})|\ldots|\mathcal{S}(\mathbf{B}_{\cdot,l})]$ by $\mathbf{B}_{\mathcal{S}}$, we can rewrite the change of basis compactly as a matrix multiplication:

$$\begin{aligned} (\mathbf{T}_i)_{\mathbf{B}} &= \langle\, \mathbf{T}_i \mid \mathbf{B}_{\mathcal{S}}\, \rangle \\ &= \mathbf{T}_i\mathbf{B}_{\mathcal{S}}. \end{aligned}$$

From this follows that the transport matrix in a basis **B** can be expressed as:

$$\mathbf{T}_{\mathbf{B}} = \mathbf{T}\mathbf{B}_{\mathcal{S}}. \tag{5.1}$$

Inserting equation (5.1) in the general image-based relighting equation (3.1) yields:

$$\begin{aligned} C &= \mathbf{T}L + S \\ &= \mathbf{T}(\mathbf{B}_{\mathcal{S}}\overline{\mathbf{B}}_{\mathcal{S}}^{T})L + S \\ &= \mathbf{T}_{\mathbf{B}}(\overline{\mathbf{B}}_{\mathcal{S}}^{T}L) + S \\ &= \mathbf{T}_{\mathbf{B}}L_{\overline{\mathbf{B}}} + S, \end{aligned} \tag{5.2}$$

where $\overline{\mathbf{B}}_{\mathcal{S}}$ is the dual basis[1] of $\mathbf{B}_{\mathcal{S}}$, and $L_{\overline{\mathbf{B}}}$ is the projection of the incident illumination L onto the dual basis $\overline{\mathbf{B}}_{\mathcal{S}}$. In the case that $\mathbf{B}_{\mathcal{S}}$ is an orthonormal basis: $\overline{\mathbf{B}}_{\mathcal{S}} = \mathbf{B}_{\mathcal{S}}$.

First of all, it is important to note that in equation (5.2), C is expressed in the default spatial basis, and thus the result of equation (5.2) is a *normal* relit image. Similarly, S is still expressed in the same space as C (i.e., a photograph of the scene without any external illumination applied).

Second, the interpretation of $\mathbf{T}_{\mathbf{B}}$ remains the same. Each row $\mathbf{T}_{\mathbf{B}i}$ corresponds to a reflectance function, expressed in a basis $\mathbf{B}_{\mathcal{S}}$. Each column $\mathbf{T}_{\mathbf{B}\cdot,j}$ of the light transport matrix $\mathbf{T}_{\mathbf{B}}$ is a photograph of the result of applying a single impulse at the j-th position

[1] A dual basis $\overline{\mathbf{B}}$ of a basis $\mathbf{B}$ is defined such that $\mathbf{B}\overline{\mathbf{B}}^{T} = I$.

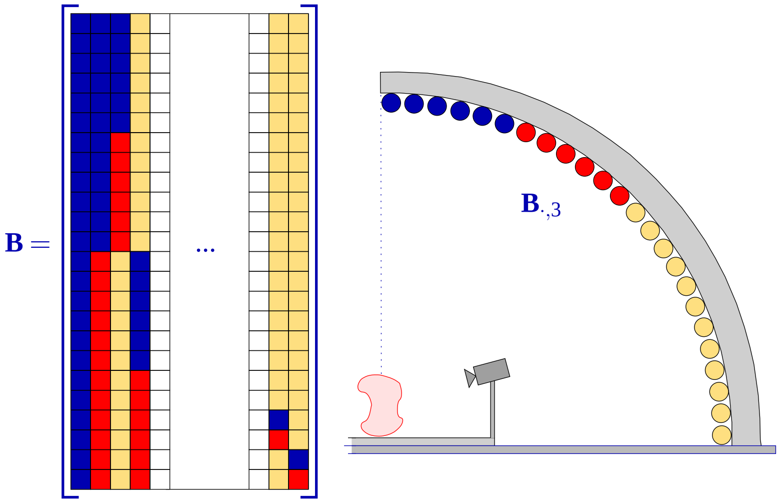


Figure 5.1: An iconic illustration of emitting an illumination condition under a basis transformation **B**. The illumination condition applied in this situation is an incident light field defined by: $L_{\overline{\mathbf{B}}} = [0,0,1,0,\ldots,0]$. Physically, this corresponds, in a Light Stage configuration, to emitting (in this case) the third basis function (i.e., third column) of the basis **B**.

of the incident illumination vector $L_{\overline{\mathbf{B}}}$. The j-th position in the incident illumination vector $L_{\overline{\mathbf{B}}}$, however, does not correspond to the j-th light source position in a Light Stage setup. The effect of the illumination condition corresponding to the j-th element in $L_{\overline{\mathbf{B}}}$, thus $L_{\overline{\mathbf{B}}} = [\delta_{i,j}]_i$, can be inferred as follows:

$$\begin{aligned} C &= \mathbf{T}_{\mathbf{B}} L_{\overline{\mathbf{B}}} + S \\ &= \mathbf{T}_{\mathbf{B}} [\delta_{i,j}]_j + S \\ &= \mathbf{T}\mathbf{B}_{\mathcal{S}} [\delta_{i,j}]_j + S \\ &= \mathbf{T}(\mathbf{B}_{\mathcal{S}} [\delta_{i,j}]_j) + S \\ &= \mathbf{T}\mathbf{B}_{\mathcal{S}\,\cdot,j} + S. \end{aligned}$$

In other words, applying a single impulse at the j-th position in $L_{\overline{\mathbf{B}}}$ equals to emitting $\mathbf{B}_{\mathcal{S}\,\cdot,j}$ in a standard Light Stage configuration. This illumination condition corresponds to the j-th basis function $\mathbf{B}_{\cdot,j}$. This is illustrated in figure 5.1. In fact, the same conclusion holds for the Light Stage setup in normal operation (i.e., without a basis transformation applied). The basis implicitly used in this case corresponds to: $\mathbf{B} = I$, or, $\mathbf{b}_{i,j} = \delta_{i,j}$. In other words, turning on only the j-th light source corresponds to the j-th basis function.

A similar interpretation of $\mathbf{T}_{\mathbf{B}}$, can be obtained from equation (3.3). In this equation, a single element c_i of the exitant illumination C (i.e., a single pixel in a relit image) is the result of the inner-product of the reflectance function $\mathbf{T}_i$, and the incident illumination L. In other words, it is the projection of the reflectance function onto the incident illumination. By selecting the emitted incident illumination to be equal to one of the basis vectors $\mathbf{B}_{\mathcal{S}\cdot,j}$, the reflectance function is in effect projected onto this basis, resulting in an observed coefficient $t_{i,j}$. Repeating this for each basis vector, a vector $[t_{i,j}]_j$ is obtained that describes the complete projection of the reflectance function $\mathbf{T}_i$ onto the basis $\mathbf{B}_{\mathcal{S}}$. Formally:

$$\mathbf{T_B}_i = [t_{i,j}]_j.$$

Thus, $\mathbf{T_B}$ is a set of reflectance functions of a scene, each expressed in a basis $\mathbf{B}$. To compute a valid relit image, the incident illumination must be expressed in such a way that the result of the dot product equals C. This can be done by decomposing the original incident illumination in the dual basis $\overline{\mathbf{B}_{\mathcal{S}}}$.

As mentioned before, both interpretations are similar. Intuitively, the second interpretation follows a more natural derivation of $\mathbf{T_B}$. The first interpretation, however, is more convenient to understand how the incident illumination is influenced (i.e., $L_{\overline{\mathbf{B}}}$).

5.3 Non-linear Approximation

Expressing a reflectance function in a different basis opens a number of interesting possible applications. One of these applications is the compact representation of reflectance functions. For example, suppose a scene, that only contains diffuse unoccluded surfaces, is captured, and is projected onto a Fourier basis. The resulting reflectance functions will be very smooth, and will probably only contain low frequency components. In other words, the coefficients corresponding to the high frequency components will be very small, or even zero. It does not make much sense to store these zero coefficients, since they do not contribute significant information. This is basically the idea behind a number of lossy compression techniques. Other compression techniques, such as: run-length encoding, Lempel-Ziv encoding [118] and variants (e.g., LZW encoding [108]), entropy encoding (e.g., Huffman coding [39], arithmetic coding [112], ...) will not be discussed in this work. These various (lossless) compression techniques can be used in conjunction with the methods discussed in this section.

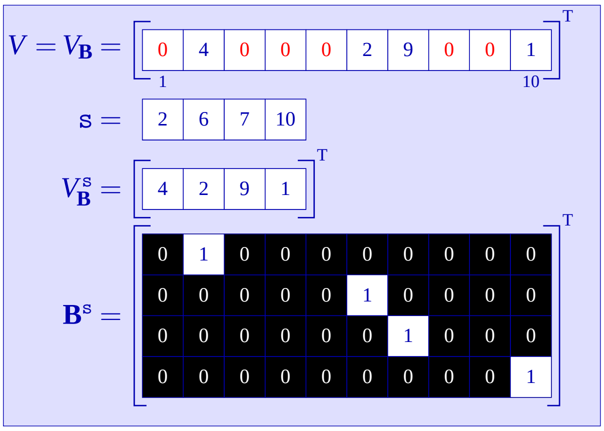
Let's look at this basic idea in a more formal setting. Suppose we want to compress a vector V of length l. First, this vector V is expressed in a basis $\mathbf{B}$:

$$V_{\mathbf{B}}^T = V^T \mathbf{B}.$$

Next, define the set $\mathtt{s}$ by:

$$\mathtt{s} = \{\, i \,|\, i \in \{1,...,l\} \; and \; (v_{\mathbf{B}})_i \neq 0 \,\}.$$

Using this set $\mathtt{s}$, a compressed vector $V_{\mathbf{B}}^{\mathtt{s}}$ can be defined from $V_{\mathbf{B}}$ by:



$$V = V_{\mathbf{B}} = \begin{bmatrix} \begin{array}{cccccccccc} 0 & 4 & 0 & 0 & 0 & 2 & 9 & 0 & 0 & 1 \end{array} \end{bmatrix}^{\mathrm{T}}$$

$$\mathtt{s} = \begin{array}{cccc} 2 & 6 & 7 & 10 \end{array}$$

$$V_{\mathbf{B}}^{\mathtt{s}} = \begin{bmatrix} 4 & 2 & 9 & 1 \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{B}^{\mathtt{s}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$$

Figure 5.2: An illustration of compressing a vector $V_{\mathbf{B}}$ expressed in a basis $\mathbf{B} = I$. The set $\mathtt{s}$ contains the indices of the non-zero elements of $V_{\mathbf{B}}$. The original vector V can be reconstructed from compressed vector $V_{\mathbf{B}}^{\mathtt{s}}$ using the basis transform $\mathbf{B}_{\mathtt{s}} = \overline{\mathbf{B}_{\mathtt{s}}}$.

$$(v_{\mathbf{B}}^{\mathtt{s}})_i = (v_{\mathbf{B}})_{\mathtt{s}[i]}.$$

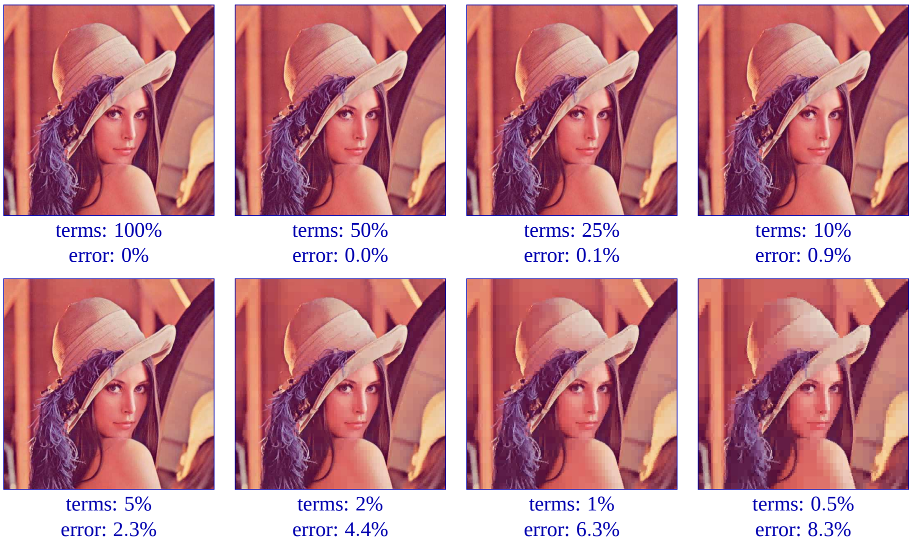
In other words, the vector $V_{\mathbf{B}}^{\mathtt{s}}$ only contains the non-zero coefficients of $V_{\mathbf{B}}$. The length n of this vector $V_{\mathbf{B}}^{\mathtt{s}}$ is less than (or equal to) than the length l of the original vector V. The vector V can be reconstructed *exactly* by defining a reconstruction transformation matrix $\overline{\mathbf{B}^{\mathtt{s}}} = [\overline{\mathbf{B}}_{\cdot,\mathtt{s}[1]}|\ldots|\overline{\mathbf{B}}_{\cdot,\mathtt{s}[n]}]$ with dimensions $n \times l$:

$$V^T = V_{\mathbf{B}}^{\mathtt{s}\,T}\,\overline{\mathbf{B}^{\mathtt{s}}}^T. \tag{5.3}$$

The vector $V_{\mathbf{B}}^{\mathtt{s}}$ can be seen as the vector $V_{\mathbf{B}}$ where the zero elements are *removed*. The original vector V can be obtained from $V_{\mathbf{B}}$ by multiplying it with the dual basis matrix $\overline{\mathbf{B}}$. In this case, the zero elements are multiplied with the corresponding columns in $\overline{\mathbf{B}}$. These columns do not contribute to the final reconstruction, and thus by *removing* these from $\overline{\mathbf{B}}$ (i.e., $\overline{\mathbf{B}^{\mathtt{s}}}$), an identical effect can be obtained by multiplying $V_{\mathbf{B}}^{\mathtt{s}}$ with $\overline{\mathbf{B}^{\mathtt{s}}}$. This is illustrated in figure 5.2.

The probability that a coefficient is exactly zero, however, is very small. Therefore, this technique is mostly used after quantization of the original vector V, increasing the probability of having zero coefficients. Quantization, however, can significantly reduce the accuracy of the reconstruction globally. Another solution is to leave out all unimportant components. This solution also introduces a loss in accuracy, but only locally in the unimportant details. Important components are represented with full accuracy. Formally, this is achieved by defining a set $\mathtt{s}^\tau$:

$$\mathtt{s}^\tau = \{\, i \,|\, i \in \{1, \ldots, l\} \textit{ and } (v_{\mathbf{B}})_i \geq \tau \,\}.$$



terms: 100%
error: 0%

terms: 50%
error: 0.0%

terms: 25%
error: 0.1%

terms: 10%
error: 0.9%

terms: 5%
error: 2.3%

terms: 2%
error: 4.4%

terms: 1%
error: 6.3%

terms: 0.5%
error: 8.3%

Figure 5.3: Non-linear Haar wavelet approximation illustrated on the Lena image. A 512 × 512 image is compressed using a non-linear Haar wavelet approximation. For each example the relative number of terms and the relative error are shown.

As a result, equation (5.3), becomes an approximation:

$$V^T \approx V_{\mathbf{B}}^{\mathtt{s}^{\tau}\,T}\overline{\mathbf{B}^{\mathtt{s}^{\tau}}}^{T}.$$

Therefore, these kinds of compression techniques are called *lossy* approximations. The error on this approximation can be computed by:

$$e = ||V_{\mathbf{B}}^{(\mathtt{s}-\mathtt{s}^{\tau})\,T}\overline{\mathbf{B}^{(\mathtt{s}-\mathtt{s}^{\tau})}}^{T}||_2.$$

Compression can be achieved if the storage requirements of both $\mathtt{s}^{\tau}$ and $V_{\mathbf{B}}^{\mathtt{s}^{\tau}}$ are less than the storage requirements of V. This kind of approximation, is also called a *non-linear approximation* (see [24] for an in depth overview on non-linear approximation methods). The name 'non-linear approximation' comes from the fact that the approximations come from a non-linear manifold (i.e., a union of partially overlapping linear subspaces[2]). The effect of non-linear approximation is illustrated in figure 5.3.

Although non-linear approximation is possible in any basis, we will limit ourself to non-linear approximations using wavelet basis functions. In the next section, non-linear approximation is applied to reflectance functions. As noted before, reflectance

[2] The union of partially overlapping linear subspaces is not necessarily a linear space. For example, consider a $2D$ space, and consider the two linear subspaces $(x,0)$, and $(0,y)$. Except for $(0,0)$, any sum of two elements of both subspaces is not contained in the union, and thus this union is not linear.

functions are not band-limited, and as a consequence, non-linear approximation will not work well with bases that do not localize well in the spatial domain (e.g., Fourier basis functions), or in the frequency domain (i.e., default spatial basis), since these bases yield a large number of coefficients when the functions contains respectively high frequency peaks (e.g., specular highlights), or exhibit a low frequency behavior (e.g., unoccluded diffuse reflections).

5.4 Compression of Reflectance Functions

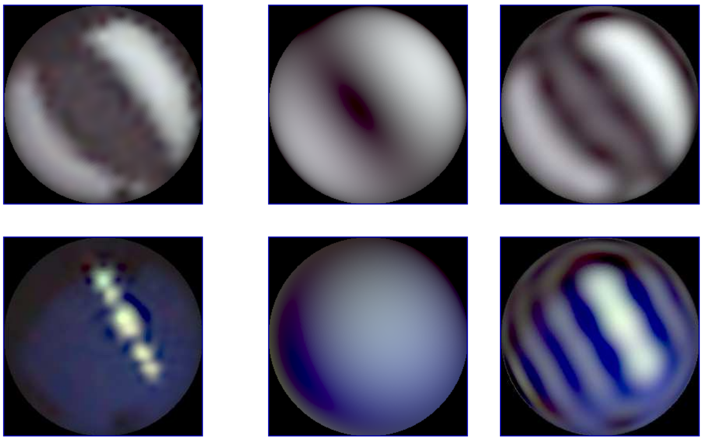
An obvious application of non-linear approximation in the context of image-based relighting is the compact representation (i.e., compression) of individual reflectance functions. An advantage of using a compression scheme based on basis projection, is that a relit image can be computed more quickly. Consider equation (5.2), which states that a relit pixel value can be computed in any basis (given that the incident illumination is expressed in the corresponding dual basis). During relighting computations, coefficients with a zero magnitude can be omitted, since these will not contribute to the final relit pixel value. Non-linear approximation effectively identifies these zero magnitude coefficients, and omits them, and thus the computational cost of relighting is reduced.

In this dissertation, we will only consider the compression of individual reflectance functions. However, the complete reflectance field, can also be compressed in a similar manner using multi-dimensional (i.e., $> 2D$) basis functions. This will, without doubt, result in superior compression ratios, but it will also complicate the relighting computations considerably.

Linear Spherical Harmonics Approximation. Before examining non-linear wavelet approximation, we first look at the approximation of reflectance functions using spherical harmonics. Spherical harmonics are used extensively in, a subdomain related to image-based relighting, *Precomputed Radiance Transfer* [44, 81, 95]. The goal of precomputed radiance transfer is to precompute the light transport in a virtual scene, and re-render it afterwards from any viewpoint under variable complex illumination (i.e., environment maps). Spherical harmonics are used to represent reflectance functions compactly. The main idea is to use only the low frequency bands to represent the data, leaving out high frequency "details". This is motivated by:

- The fact that BRDFs with low frequency responses (e.g., diffuse BRDFs) can be represented accurately using only the 3 lowest frequency bands [81].
- When illuminating a scene with low frequency lighting (i.e., an environment map that only contains low frequency components), only the lower frequency components of a BRDF contribute to the rendered result, regardless of the high frequency content of these BRDFs.

Kautz et al. [44] use the first 15 bands to represent reflection properties of synthetic materials ignoring self-shadowing effects. Sloan et al. [95] extend this technique to in-



Original 25 Coefficients 225 Coefficients

Figure 5.4: Linear spherical harmonics approximation. Two high resolution reflectance functions are approximated by using the 5 and 15 lower frequency bands (25 and 225 coefficients respectively).

clude self-shadowing by using two sets of spherical harmonics, one for the reflectance properties, and one for the self-occlusion features. A key difference in image-based relighting is that the geometry of the scene is not known, as opposed to precomputed radiance transfer where the geometry is known beforehand.

Reflectance functions for image-based relighting that only contain low frequency components can be compactly approximated using only the low frequency spherical harmonics basis functions. Increasing the number of bands allows to add more detail in the approximated reflectance functions. In figure 5.4, two high resolution reflectance functions are represented using 25 and 225 spherical harmonics coefficients, 5 and 15 bands respectively. A significant drawback of spherical harmonics is Gibbs ringing or aliasing which occurs around high frequency features, such as highlights and self-shadowing boundaries. The resulting reflectance functions are similar to the reflectance functions reconstructed directly using spherical harmonics (see chapter 4).

Non-linear Wavelet Approximation. Wavelets are well known for their use in image compression, and have also been used for representing and compressing incident illumination. Ng et al. [67] use a cube-parameterization and a non-linear Haar wavelet approximation on the incident illumination in an image-based relighting context. Using a non-linear approximation of the incident illumination significantly speeds up the relighting computations. However, it does not reduce the storage requirements for the reflectance field. A more logical choice would be to use non-linear wavelet approximation on the reflectance functions themselves, reducing computational and storage requirements. In this thesis, we do not only consider the Haar wavelet, as is done in most of the computer graphics literature, but also higher order

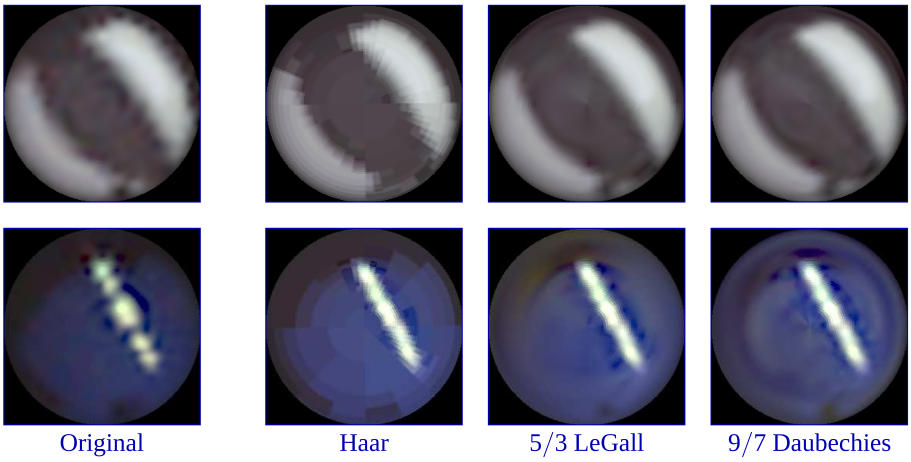


Figure 5.5: Two selected reflectance functions, non-linearly approximated, keeping only 1.5% (i.e., 256) of the total number of wavelet coefficients.

5/3 LeGall, and, 9/7 Daubechies biorthogonal wavelets. These wavelets have better properties with respect to image compression [105] than the Haar wavelet.

We performed a non-linear wavelet approximation in the latitude-longitude parameterization with a resolution of 256×64 pixels. Again, the Formula One toy car example is used. High resolution reflectance functions are obtained by taking 1280 samples, and using a multi-level B-Spline reconstruction technique to interpolate to the desired resolution. An increase in resolution will not alter the compression ratio much, since the increase in information is limited. In figure 5.5 two selected reflectance functions from the Formula One toy car example are depicted, which are non-linearly approximated using the Haar, 5/3 LeGall, and, 9/7 Daubechies wavelets respectively, keeping only the 256 largest coefficients (i.e., 1.5% of the total number of coefficients). Visually, all three wavelets perform well, given the fact that the compression ratio is $\frac{1}{64}$. However, the Haar wavelet results in a noticeably less smooth approximation of the reflectance functions.

A more thorough error analysis was performed on all reflectance functions of the Formula One toy car example. In figure 5.6 the relative Sobolev H^1-error is plotted in function of the number of wavelet coefficients. The Sobolev H^1-norm takes not only the length of the error vectors into account, but also the first derivative. The latter, ensures that differences in smoothness are also taken into account. The Sobolev H^1-norm of a vector V is defined by:

$$||V||_{H^1}^2 = V^T V + (\nabla V)^T (\nabla V),$$

where, ∇V, is a vector defined by:

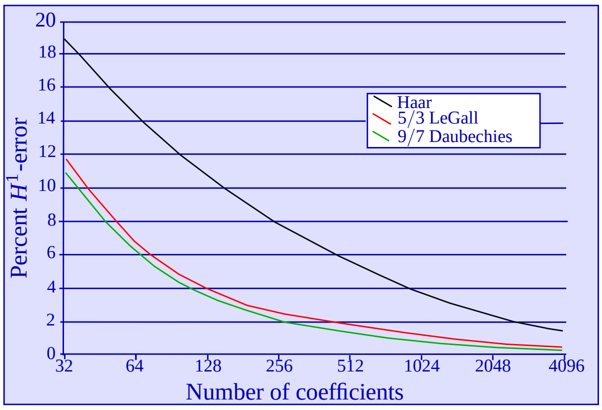


Figure 5.6: The Sobolev H^1-error versus the number of wavelet coefficients used in a non-linear wavelet approximation using three different wavelets: the Haar wavelet, the 5/3 LeGall wavelet, and the 9/7 Daubechies wavelet.

$$\nabla V = [v_i - v_{i+1}]_i \, .$$

The relative L^2-error resulted in a simular graph and is therefore omitted. In general, the Haar wavelet needs more coefficients to achieve the same relative error. For example, 4096 Haar wavelet coefficients are needed for a ±2% error, as opposed to only 256 5/3 LeGall, or, 9/7 Daubechies wavelet coefficients. The 9/7 Daubechies wavelet slightly outperforms the 5/3 LeGall wavelet in terms of error: for 256 coefficients the Haar wavelet results in a 7.05% relative error, the 5/3 LeGall wavelet in a 2.12% relative error, and the 9/7 Daubechies wavelet in a 1.88% relative error.

Using the same number of coefficients for each reflectance function in the reflectance field is not optimal. Some reflectance functions can be compressed using less coefficients while maintaining an acceptable error ratio during relighting. A better approach would be to include a sufficient number of coefficients until the error on the approximation (relative L^2-error between the uncompressed and compressed reflectance function) falls below some error-threshold. This L^2-error is sufficient for determining the number of wavelet coefficients, although it is only an indication of the error during relighting, since relighting is the inner product of the reflectance function, and, the (during compression) *unknown* incident illumination.

The Sobolev H^1-error when a variable number of wavelet coefficients per reflectance function on the Formula One toy car example is shown in figure 5.7. The errors are computed under time-varying illumination. The time-varying illumination used, is the same as used before to create the images in figure 4.18: a sliver of light rotating over the hemisphere. The left graph in figure 5.7 shows the error with respect to the error-threshold. At the top of the graph, the average numbers of wavelet

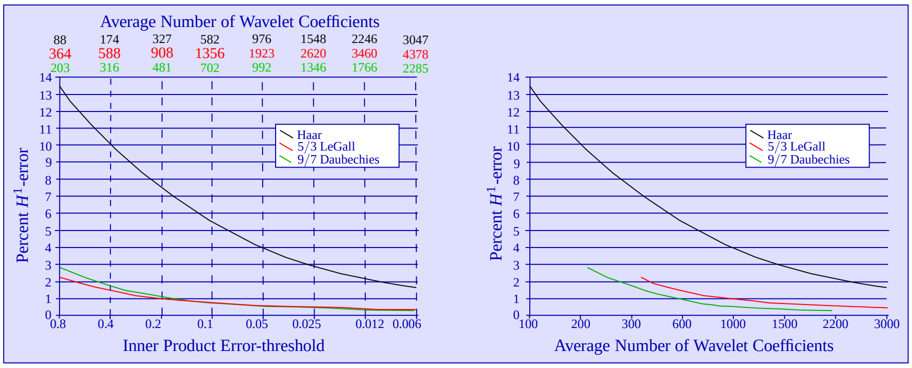
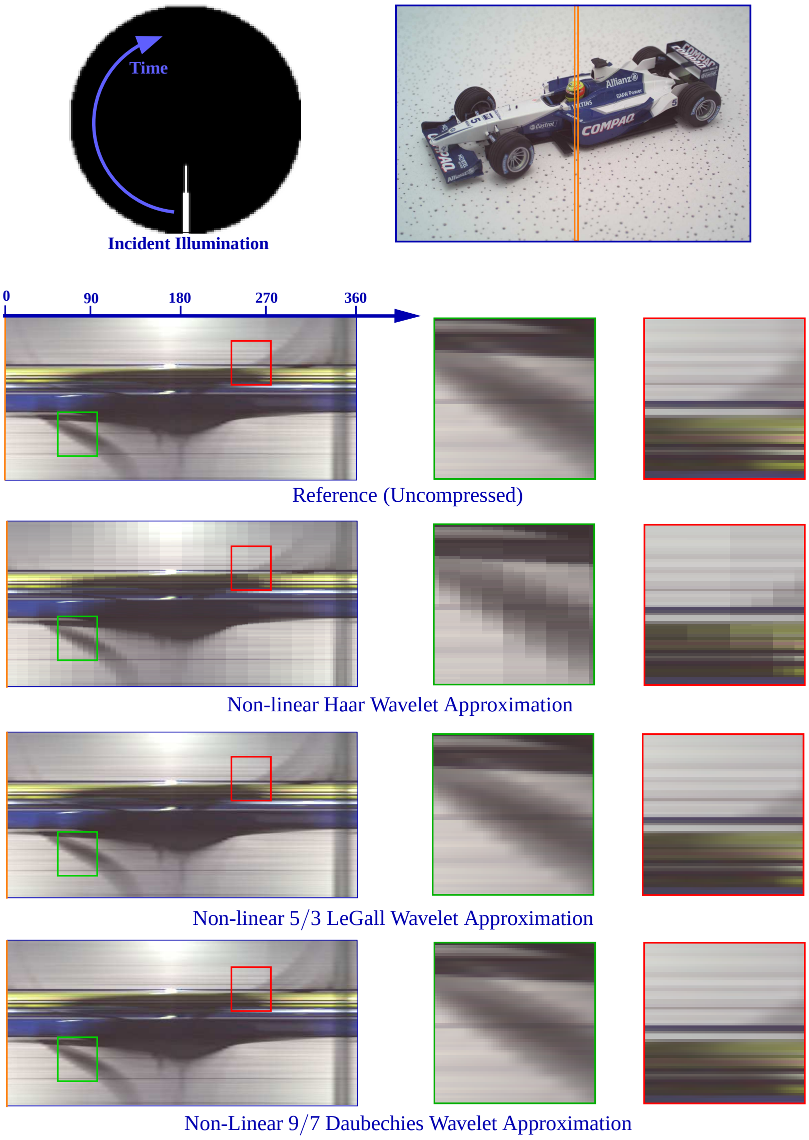


Figure 5.7: Error analysis of non-linear wavelet approximation with a variable number of coefficients per reflectance function. The left graph plots the Sobolev H^1-error versus the error-threshold. The right graph plots error versus the average number of wavelet coefficients per compressed reflectance function.

coefficients per reflectance function are shown. To better illustrate the difference in average number of coefficients and H^1-error, the right graph of figure 5.7 shows the error for each type of wavelet with an equal number of average coefficients per reflectance function. In terms of error-threshold versus H^1-error, both the 5/3 LeGall and 9/7 Daubechies non-linear approximations out-perform non-linear Haar wavelet approximation. The 9/7 Daubechies wavelet out-performs the 5/3 LeGall wavelet when taking into account the average number (and thus total compression ratio) of wavelet coefficients per reflectance function.

It is important that the non-linear approximation, for both a fixed and a variable number of coefficients, maintains as much as possible the smoothness of the original reflectance functions. Introducing additional discontinuities results in visually disturbing artefacts when animating the incident illumination. This can be seen in figure 5.8. The sweeps are created by relighting reflectance functions compressed using a variable number of coefficients (0.1% error-threshold). It is clear from this figure that, although the Haar wavelet achieves very good compression ratios, it fails to maintain smoothness. The reason is the low number of vanishing moments in the Haar wavelet. The 5/3 LeGall and 9/7 Daubechies wavelet perform much better, and still have very good compression ratios.

The correlation between the number of coefficients for a specific wavelet and the error-threshold is difficult to predict. When using a variable number of coefficients, more wavelet coefficients are assigned to reflectance functions containing many details, whereas low-detail functions are compressed using less wavelet coefficients. These details contribute little to the error, hence the small difference in error between compressing reflectance functions using a variable number of wavelet coefficients and using a fixed number of wavelet coefficients.



Reference (Uncompressed)

Non-linear Haar Wavelet Approximation

Non-linear 5/3 LeGall Wavelet Approximation

Non-Linear 9/7 Daubechies Wavelet Approximation

Figure 5.8: Visualizing the effect of time-varying incident illumination on non-linearly approximated reflectance functions. These figures are created in a similar manner as figure 4.18 in chapter 4. The reflectance functions used to create these results are non-linearly approximated using a variable number of wavelet coefficients per reflectance function. An error-threshold of 0.1% is used to determine the number of coefficients per reflectance function.

In terms of error and smoothness, the $9/7$ Daubechies wavelet is preferred (followed by the $5/3$ LeGall wavelet), achieving a compression ratio of $1:34$ for a 0.2% error-threshold, and a H^1-error of less than 1% on 256×64 reflectance functions of the Formula One toy car example. Finally, note that the non-linear Haar wavelet will be better suited to compress reflectance functions that are non-smooth (e.g., upsampled using a zero-order hold interpolation technique).

5.5 Conclusion

In this chapter, the compression of reflectance functions was discussed. It is shown that non-linear wavelet approximation is an excellent tool for compressing reflectance functions. Not only are the storage requirements reduced, but also the relighting computations are accelerated by a factor equal to the compression ratio.

6

Sampling of Wavelet Represented Reflectance Functions

In this chapter, two advanced reflectance field acquisition methods are discussed. These two techniques exploit the properties of a non-linear wavelet approximated representation of reflectance functions, in order to achieve faster acquisition. The work discussed in this chapter was first presented in [74] and [75].

6.1 Introduction

The efficiency of a non-linear wavelet approximation to reduce the storage requirements of a reflectance field is thoroughly discussed in the previous chapter (5). Although the storage requirements are significantly reduced, the time required to acquire a reflectance field remains the same. When acquiring the reflectance field of scenes containing reflectance functions that are composed of only low frequency components, the acquisition timings are still acceptable while maintaining a high level of accuracy. If the reflectance functions contain high frequency components, however, an impractically large number of samples needs to be captured to ensure an accurate reconstruction. Moreover, technical limitations make it nearly impossible to sufficiently sample reflectance functions containing very high frequency components (chapter 4). These high frequency reflectance functions arise in situations where a scene contains specular or transparent objects, such as mirrors, glasses, ...

Taking into account the fact that a non-linear wavelet approximation of a reflectance function is very compact arises a very important question: can this sparseness be exploited during the acquisition process, such that only the significant parts of a reflectance field are sampled. The main difficulty is how to select the best part to sample of an unknown function. Non-linear wavelet approximation requires a-priori knowledge to create this order of importance. This a-priori knowledge is unavailable when the whole reflectance function is unknown beforehand (i.e., during acquisition).

Due to mechanical limitations a Light Stage apparatus will not be practical to capture very high frequency reflectance functions. However, there exists a sub-domain in computer graphics which can handle specular and transparent materials: *environment* *matting* [12, 119]. When image-based relighting and environment matting were introduced, it was not quite clear that these two methods solve basically the same problem. In [63, 73], environment matting was first considered in an image-based relighting context. Although Debevec et al. [20] also considered combining the Light Stage and environment matting, they still regarded it, however, as a mere matting technique. Because of its relevance, we will discuss environment matting in greater detail in section 6.2.

The remainder of this chapter is structured as follows. First a general overview of matting and environment matting is given in section 6.2. Then, the specific details and calibration of a new setup for the acquisition of high frequency reflectance fields is discussed in section 6.3. Next, *tree approximations with wavelets*, a hierarchical variant of non-linear wavelet approximation is introduced (section 6.4). This hierarchical non-linear approximation forms the basis of a *progressive tree approximation* algorithm (section 6.5), that is used to develop two novel acquisition methods. The first method explicitly samples the reflectance field selectively in the wavelet domain (section 6.6). The second method uses an advanced implicit sampling method, which works on individual reflectance functions (section 6.7). This chapter concludes with some final remarks in section 6.8.

6.2 Environment Matting

Environment matting is an extension of the conventional matting process [79, 96]. Originally, environment matting was mainly intended to handle refractive materials, such as glass, in a convincing manner. However, in subsequent environment matting techniques, the scope of the matting process was further extended to include glossy and diffuse reflections from the surrounding environment. Before discussing the different environment matting techniques in detail, we first explore some of the conventional matting techniques briefly.

Conventional Matting. Conventional matting has been around for many years. One of the first publications on this topic in computer graphics is by Porter and Duff in 1984 [79], in which they describe the need for alpha-matting and define basic operators needed to create a multitude of effects. An alpha-matte itself is already an extension to regular matting: i.e., the separation of an image in foreground and background elements. A problem with regular matting is that the boundaries of a foreground element are not necessary along the pixel boundaries and can for example cover only half of a pixel (see figure 6.1.a). When compositing such a foreground element on a new background, aliasing occurs, and the transition between foreground and the new background seems unnatural. Alpha-matting solves this problem not only by considering a pixel to be exclusively background or foreground, but a weighting of foreground and background. This weighting is numerically represented by an α-value. For example, a pixel which is half covered by a foreground element, and half by a background element, will have an α-value of 0.5 (see figure 6.1.b). Alpha-matting is governed by the

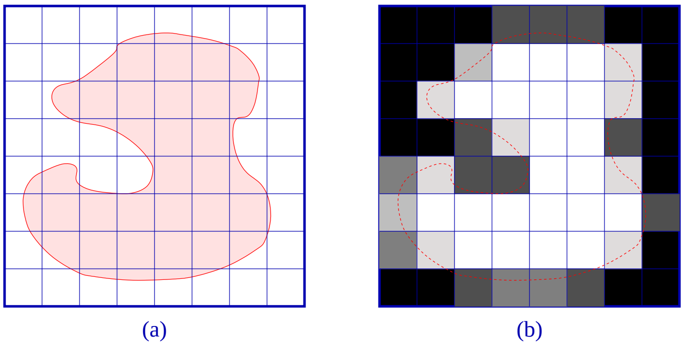


Figure 6.1: A conceptual illustration of an alpha matte. (a) A red foreground element on an 8×8 pixel grid. Some pixels are only partially covered by the foreground element. (b) The corresponding alpha matte. Pixels partially covered have an opacity value of less than 1. The exact value of the opacity (alpha) value corresponds to the fraction of the pixel covered by the foreground element.

following equation:

$$C = \alpha F + (1 - \alpha)B, \tag{6.1}$$

where C is the resulting composited image, F the foreground image, and B the background image. All three images have the same size. Note, α is in this case a matrix of α-values, one for each pixel in C. The α-value is sometimes called: *opacity-value*, since it indicates how transparent a foreground pixel is ($\alpha = 1$: the foreground pixel is completely opaque. $\alpha = 0$: the foreground pixel is completely transparent). Both compositing and extracting of an alpha-matte is governed by equation (6.1). In the case of compositing, both F, B, and α are known. Computing this equation yields the composited image. In the case of alpha-matte (α) extraction, C is known. Depending on the matting technique used, some additional information regarding F or B is also known.

Although the matting equation (equation (6.1)) is rather straightforward, the implications and practical problems are, even today, significant. In 1996, Smith and Blinn [96], discuss the problems associated with the acquisition of alpha-mattes of real objects against a single constant colored background (i.e., blue-screen matting). Surprisingly, the problem is deemed mathematically unsolvable, which is particularly surprising since blue-screen matting is in fact a commonly used matting technique in the film and video industry. However, Smith and Blinn show that an exact solution is possible when the scene is shot against two completely different backdrops, that differ at each point in color in the camera image.

In practice multiple backdrops are not always possible and a number of solutions have been presented. One of the more interesting approaches is natural image matting [11, 85, 100]. Natural image matting, starts from a single photograph of a scene

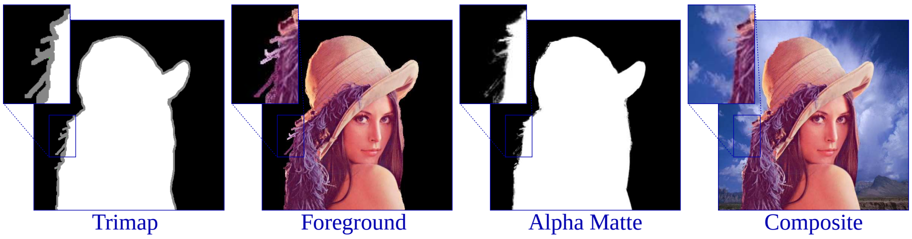


Figure 6.2: Natural image matting applied on the Lena image. Given a user-specified trimap, the foreground and alpha-matte is computed. The user-specified trimap indicates which part is definitely foreground (white), definitely background (black), or which has to be computed (gray). Afterwards, the foreground can be composited using equation (6.1) on a novel background.

against a random background and a trimap. This trimap is a user-specified map which indicates which of the pixels in the photograph are definitely foreground, definitely background, or a mix of both. The α-value of this last category is inferred from the properties of the foreground and background marked pixels. In figure 6.2 an example of natural image matting is shown.

Other solutions to the matting problem use a more elaborate setup, such as an additional infrared camera [106] or multiple cameras [64], a flash and non-flash photograph pair [101], or polarized light and appropriate filters [5].

Environment Matting. Conventional matting does not work well with transparent foreground objects. At first this might seem a bit strange, since an alpha-matte encodes how transparent a pixel of the foreground is. The problem lies in the fact that a transparent material refracts light, in effect deforming the background image before being transmitted (alpha-matted) through the foreground element. In other words, it is not necessarily the background pixel directly behind the foreground pixel that is partially visible, but it can be any background pixel. This is illustrated in figure 6.3.

Environment matting and compositing, was first presented by Zongker et al. [119] and later extended by Chuang et al. [12]. Unlike conventional matting, an environment matte does not only represent the opacity of a pixel, but it also includes the reflection and refraction effects of the backdrop through the scene. To create an environment matte, a scene is photographed from a single vantage point against a series of known background patterns. Usually a CRT monitor, positioned behind the scene, is used to emit these different backdrop patterns onto the scene. Using the information from the recorded photographs, an approximation of the light transport from the background through the scene into the camera is computed for all camera-pixels. The main difference between different methods lies in which background patterns are used, and how light transport is computed and represented.

In the approach of Zongker et al. [119] horizontal and vertical stripe patterns are

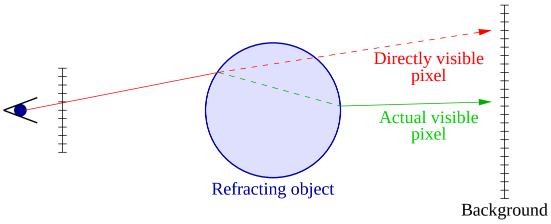


Figure 6.3: The refraction problem. A transparency value per pixel is not sufficient to create convincing composited images of objects containing refracting materials, since the image of the background can be deformed through the object. The red line illustrates the situation where refraction is not taken into account (i.e., only a transparency value). The green line represents the real refracted path.

emitted onto a scene. For each emitted pattern a photograph of the scene is recorded from a fixed viewpoint. The environment matte, which encodes the reflection and refraction properties of the scene, is represented for each pixel by a single reflection coefficient and a normalized box filter on a rectangular support area on the backdrop. A least squares optimization procedure is used to extract the support areas and reflection coefficients from the recorded photographs. Compositing (i.e., applying a novel backdrop) is performed by filtering, for each pixel, the novel background over the support area and scaling the result by the reflection coefficient. The method itself is elegant, and requires few photographs to be recorded.

This approach, however, has a few limitations, as pointed out by Chuang et al. [12]. A single rectangular support area and a single reflection coefficient per pixel are not sufficient to capture the complex reflection and refraction effects of dielectrics or rough materials. In addition, the choice of a rectangular support area can cause excessive blurring in the final image. To address these problems, Chuang et al. sweep different oriented Gaussian stripes across the background to capture the environment matte. This resembles the space-time analysis used in 3D range scanning [15, 43]. The environment matte is approximated by a limited number of oriented elliptical Gaussian filters, each with a single reflection coefficient. Non-linear optimization is necessary to compute the oriented elliptical Gaussian filters. Compositing is performed similarly to Zongker et al., except that contributions from multiple filters for a pixel on the backdrop are added together. Chuang et al. [12] also presented an environment matting method for acquiring environment mattes in real-time, that uses a single color gradient as backdrop pattern. This method, however, is limited to perfect specular materials that do not modulate the emitted color. In this case, the environment matte is reduced to warping the background image before compositing.

The representational power of these filters is very limited, and directly related to the size of the filter footprint. Therefore accurate mattes can only be extracted by these environment matting methods for objects containing specular, and, in the case of the first method of Chuang et al. [12], highly glossy material properties.

Wexler et al. [111] presented an environment matting extension that is able to work without knowledge of the exact form of the backdrop images used. It relies on having enough background samples, or sufficiently rich backdrop images (e.g., by moving a backdrop image behind the scene), to successfully extract an environment matte. During processing, a free-form filter, defined by different weights for individual pixels on the background, is computed. The number of required background samples per pixel ($\sim$ number of photographs that need to be recorded) is directly proportional to the size of the filter. This makes this method only suited for high frequency reflections (i.e., small footprint filters). Therefore, Wexler et al. only demonstrate their technique on specular materials.

Zhu and Yang [117] model light transport using a similar representation as [111]. Time-varying cosine wave patterns, with a predetermined time frequency per pixel, are used as input. The main advantage of frequency based patterns is the robustness with respect to measurement noise. Using a different frequency per pixel would result in a large number of required illumination patterns. Therefore two series of patterns are used, each with a constant frequency for each row or column respectively. The method is biased towards elliptical Gaussian responses, due to the separation in horizontal and vertical patterns, and is therefore only suited for computing reflectance functions with a compact footprint, i.e., specular and glossy reflections.

Matusik et al. [61] use known natural illumination (i.e., photographs) as input. The reflectance function of each pixel is represented by a summation of weighted box filters, which are inferred from the effect of the input illumination on the scene by a progressive algorithm. This algorithm starts with a coarse approximation of the reflectance function (i.e., few box filters), and subsequently splits one of the box filters in two, such that the error is minimized. For this purpose, in each iteration of this progressive algorithm, a constrained linear system is solved using quadratic programming. A spatial correction is used to further enhance the results. The splitting of filters, enables to obtain very complexly shaped, small and large, reflectance function approximations. Matusik et al. demonstrate their technique on a number of objects containing specular, glossy, and diffuse surfaces. It is not clear from [61] how many input images are required, or what the constraints are on the input illumination.

Environment Matting versus Image-based Relighting. Although environment matting found its roots in a different graphics subdomain, it essentially tries to solve the same problem as image-based relighting: to model the effects of incident illumination on the observed scene from a single viewpoint. Typically for most environment matting methods is that they try to model light transport through the scene by a set of simple filters. Using filters to represent light transport bears some similarity to BRDFs. However, the filters used are more general and require no information of the underlying geometry, but they are still restricted by their limited representational power.

There are a number of problems with environment matting:

- **The error of the environment matte approximation is unknown**, as is the error on the composited images. This error depends on the scene properties, the filters used (e.g., a box filter versus an elliptical Gaussian filter), the illumination patterns used during the recording process, and the background image itself used during compositing. All these parameters make it impossible to predict (a reasonable upper-bound on) the error exactly without explicitely computing it for each separate composited image.

- **Diffuse surfaces are still problematic**, because an elliptical Gaussian and most other filters are not sufficient to capture the effects of diffuse reflections. A notable exception is Matusik et al. [61]. Diffuse materials have a large area of support which can be irregularly shaped because of occlusion and self-shadowing. These irregularly shaped support areas are difficult to approximate accurately with a limited number of elliptical Gaussian filters. A more general model is needed with a greater degree of flexibility.

- Most environment matting methods rely on **non-linear optimization procedures**, that require a significant amount of post-processing time, to compute the final environment matte approximation. Such methods usually depend in a non-trivial manner on a number of parameters (e.g., error-thresholds) that greatly affect the quality of acquired results. Non-linear optimization procedures also require a significant amount of processing time. Increasing accuracy by using better filters or more approximation terms, would increase post-processing time even more. Again, Matusik et al. [61] forms a notable exception in this regard.

- **Unclear relation between input stimuli and final optimized environment matte**. Without a clear relation, it is hard to predict what the error and the effect is of recording more photographs (under different illumination conditions). There is no guarantee that an increase in the number of illumination conditions results in a more accurate environment matte.

Environment matting also raises some interesting questions with respect to image-based relighting:

- **High frequency reflectance functions.** Can we use similar techniques (as environment matting) to handle high frequency reflectance functions, such as specular reflections and refractive materials?

- **Decoupling incident illumination resolution and acquisition time.** In environment matting, the number of illumination patterns, and thus the duration of the acquisition process, is insignificant versus the resolution of incident illumination. The number of illumination patterns in environment matting techniques varies from 1 to 2000. This low number of patterns is due to the non-linear dependency of the filter parameters on the information gained from the illumination patterns. Can similar ratios be obtained in a linear image-based relighting setting?

The setup used in most environment matting papers (i.e., a CRT monitor) as a spatially controllable light source, is very interesting. However, due to the dense packing of controllable illumination elements (i.e., pixels on the surface of the screen), naively applying a similar acquisition process as the Light Stage, results in some practical problems. A typical CRT monitor has approximately a resolution of 1024×1024 pixels. If a Light Stage acquisition approach would be taken (i.e., for each light source, record a high dynamic range photograph of the scene illuminated by this light source), the same number of HDR photographs needs to be recorded. Assuming that each HDR photograph takes 1 second to record, this would result in 12 days of continuous acquisition. Not only the duration of the acquisition is problematic, also the storage of all recorded photographs. Assuming that each HDR photograph can be losslessly compressed to 50Kb, this would still result in a total storage requirement of 50Gb. This also implies that the same amount of data needs to be processed before a relit image can be computed.

In this chapter, two new image-based relighting techniques are introduced, that were developed using the theoretical framework presented in chapter 3, and that use a CRT monitor as source of incident illumination, but do not suffer from the aforementioned problems. A CRT monitor as controllable light source potentially enables to capture high frequency reflectance functions accurately. The goal is to develop efficient acquisition methods for capturing the light transport accurately and efficiently through a scene, without placing restrictions on the materials present in the scene.

6.3 Setup

We use a similar setup as in the environment matting techniques discussed in the previous section. An object is placed in front of an emitter that is capable of displaying structured patterns (e.g., a plasma screen, or a CRT monitor). In our setup we use a CRT monitor (figure 6.4). A series of illumination patterns is emitted from the CRT monitor. The resulting effect of each illumination pattern (on the object) is captured by means of a digital camera.

Using a CRT monitor as source of incident illumination has a number of advantages:

- **High frequency incident illumination.** Due to the tight spacing of individual pixels on a CRT monitor, specular and glossy materials can be illuminated without introducing visual aliasing.
- **Complex illumination patterns.** Each pixel can emit illumination independently of other pixels. This allows to use complex illumination patters (i.e., more than a single light source, and each with different intensities or colors).
- **Reference images.** Because a CRT can emit complex illumination patterns, it can also be used to create a reference image. In this case, the incident illumination used to create a relit image, can also be emitted onto the scene from a

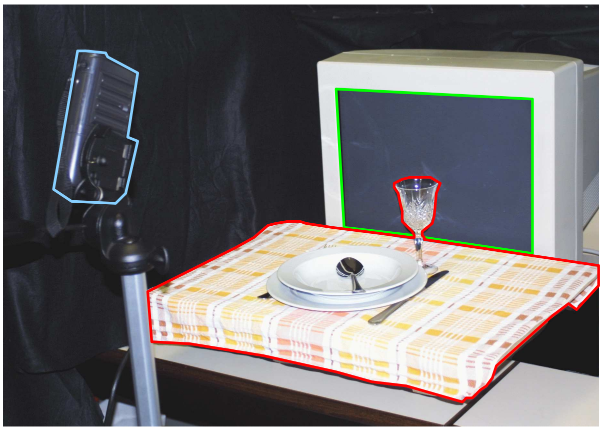


Figure 6.4: The scene is highlighted in red, the camera in blue, and the emitter in green.

CRT monitor. In theory, the relit image and reference image should be indistinguishable. Except for the Light Stage v3 [21], this is not possible to achieve (in a single photograph) with a Light Stage design.

Calibration. Both the CRT monitor and the camera need to be radiometrically calibrated such that both devices behave linearly in terms of exitant and incident illumination respectively. The camera response curve is determined using the technique of [22], and each recorded photograph is converted to a high dynamic range image (see [83], chapter 4 for an in depth discussion on high dynamic range photography). A minimal shutter time of 1 second is used to avoid synchronization problems with the refresh rate of the CRT monitor. The gamma curve of a CRT monitor is measured by recording high dynamic range photographs of the CRT monitor while emitting 256 different intensity images. To avoid additional errors, we opted for not fitting an analytical gamma curve through the measured intensity values, but use the discrete representation (i.e., lookup table) directly. The discrete gamma curve is inversely applied before emitting an illumination pattern from the CRT monitor, such that the intensity ratios between the pixels of the patterns and the emitted illumination are the same.

In chapter 3, the assumption was made that all computations could be performed directly in RGB space. This implies that all three color channels can be processed in parallel. As a result, only monochromatic illumination patterns (i.e., grayscale) need to be emitted from the CRT monitor. And thus, no color calibration, except white balancing, is required.

Recording a reference image. When creating a reference image, care has to be taken that the illumination used to create a relit image is the same as is used to create a reference image.

Using the gamma-curve, the non-linear mapping of intensities can be undone. However, when using colored incident illumination, a color calibration is required. Color calibrating a CRT monitor is a very difficult problem, since the color mixing is not necessarily linear, nor does it corresponds with the RGB color space of the camera (i.e., cross-talk). Since it usually concerns only a few incident illumination patterns (to create a reference image), a different approach can be taken. Taking an HDR photograph of the CRT monitor while emitting the reference illumination pattern, yields the actual emitted illumination. Using this HDR photograph, as incident illumination during relighting, would in theory result in an exact relit image.

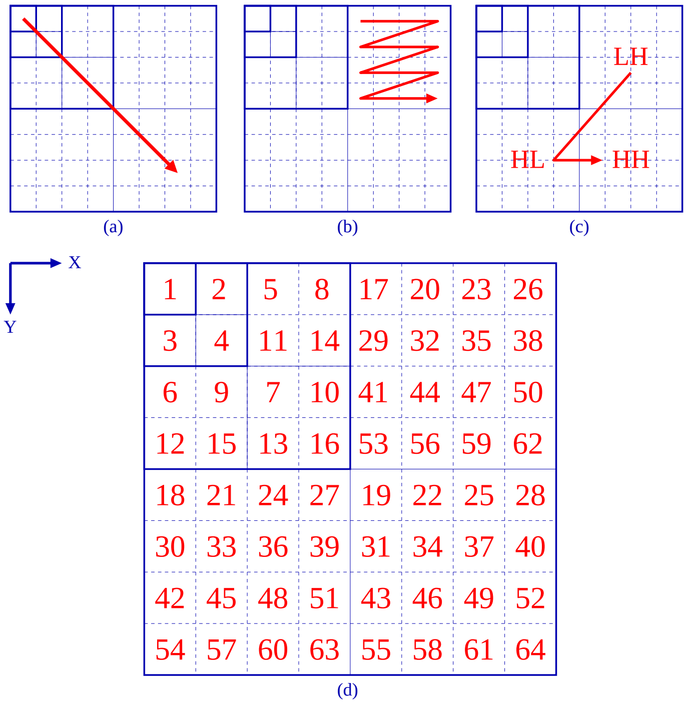
There is, however, a slight practical problem with this approach: the geometrical distortions caused by the intrinsic and extrinsic camera settings (i.e., lens distortion and camera position) and the curvature of the CRT surface. To find the correspondence between CRT pixels and camera pixels, we turn to structured light techniques in computer vision for inspiration. For an overview on structured light methods we refer the interested reader to [4, 7, 86]. To create a homography between the camera and the CRT monitor, we emit horizontal and vertical stripe patterns, each containing approximately 64 stripes, and record an HDR photograph for each emitted pattern. Next, the edges of the individual stripes are detected in the resulting HDR photographs. Since the exact positions of the stripe edges are now known in both CRT screen space as in camera space, a homography between both can be easily interpolated. This homography is accurate enough to undo the geometrical deformations in the recorded HDR photographs of the incident illumination (emitted from the CRT monitor), and thus this photograph can subsequently be used to compute a relit image.

6.4 Lossy Wavelet Approximations

As noted before, a brute force, Light Stage like, acquisition process is impractical due to the high number of illumination elements on a CRT monitor. Before developing the novel acquisition methods, using a CRT monitor as acquisition device, a review of lossy wavelet approximation is made (section 6.4). This is done in order to evaluate if these methods can be adapted to optimize the acquisition process, i.e., evaluating their usability to accelerate the acquisition of reflectance fields (i.e., online compression).

Linear Wavelet Approximation. Linear wavelet approximation compacts a function by keeping the first *n* coefficients of function expressed in the wavelet domain. This implies an a-priori ordering of the coefficients, and thus basis functions. For example, in the case of linear wavelet approximation the following ordering of the ($2D$) basis functions can be used:

1. Low detail levels come before high detail levels (figure 6.5.a).



(a) (b) (c)

1	2	5	8	17	20	23	26
3	4	11	14	29	32	35	38
6	9	7	10	41	44	47	50
12	15	13	16	53	56	59	62
18	21	24	27	19	22	25	28
30	33	36	39	31	34	37	40
42	45	48	51	43	46	49	52
54	57	60	63	55	58	61	64

(d)

Figure 6.5: Linear order in a 2*D* wavelet domain. (a) Low detail levels come before high detail levels. (b) Within a level, a scanline ordering is used. (c) Ordering over directions. (d) The complete ordering illustrated on a 3-level wavelet domain (i.e., 8×8 coefficients).

2. Within a level, the scan order is along the x-coordinate (i.e., the y-coordinate is incremented when the x-coordinate reaches its maximum value and is subsequently reset to zero) (figure 6.5.b).

3. For each coordinate (i.e., (x, y)-coordinate), the different directions are scanned as: LH, HL, HH (where L and H stand for the low- and high-pass filters over either x or y direction) (figure 6.5.c).

Figure 6.5.d illustrates this linear order. An advantage of linear wavelet approximations is that no additional information needs to be stored along with the wavelet coefficients to reconstruct the functions, since the “location” of the coefficient is implicitly known, due to the ordering, in the wavelet domain. The approximation error, however, greatly depends on the behavior of the function in the ordered wavelet domain. For example, the above ordering works well for smooth functions (e.g., in a Besov space) since the wavelet coefficients decay with increasing level. Singularities, on the other hand, will not compress well, since the wavelet coefficients are localized in each wavelet level.

Non-linear Wavelet Approximation. Non-linear wavelet approximation keeps the *n* largest coefficients. This assumes an a-posteriori ordering (with respect to wavelet decomposition of the function) of the wavelet coefficients. The approximation error is guaranteed to be minimal for the number of wavelet coefficients. A disadvantage is that additional information, a coordinate denoting the "location" in the wavelet domain, is required per wavelet coefficient to reconstruct the function.

Tree Approximation with Wavelets. A tree approximation [3, 13] of *n* wavelet coefficients, is the set of coefficients for which the approximation error is minimal, and each ancestor of an included wavelet coefficient is also included in the approximation. The notion of an ancestor of a wavelet comes naturally from the hierarchical nature of the wavelet domain. An advantage of a tree approximation is that less additional information per coefficient is required to store and reconstruct a tree approximated function than a non-linear approximation. For each coefficient, a single bit per child is needed to indicate if the tree approximation includes these children. A non-linear wavelet approximation on the other hand requires an index, usually a coordinate, per coefficient to successfully reconstruct the original function. More information, however, is required than in a linear approximation. Surprisingly, the performance of a tree approximation is close to that of a non-linear approximation and superior to that of a linear approximation for a large class of functions. For example, smooth functions have wavelet coefficients that decay as the wavelet level increases, and thus the parents of included nodes have a larger coefficient and is therefore also included. The wavelet coefficient of a singularity have large wavelet coefficients organized along the branches of a wavelet tree. Both these types of functions can be represented by a tree approximation without loss of efficiency compared to a non-linear wavelet approximation. To create an optimal tree approximation, the original function must be known beforehand.

Comparison. Linear and non-linear approximations are each others opposites in the following sense:

- **Linear Approximation** is sequential (i.e., linear) over a "spatial" ordering, but random over a wavelet coefficient magnitude ordering.
- **Non-linear Approximation** is random (i.e., non-linear) over "spatial" ordering, but sequential over wavelet coefficient magnitude ordering.

Practically, this implies that linear approximation requires no a-priori knowledge of the function itself. An approximation is obtained by taking the first *n* coefficients, regardless of their magnitude (i.e., importance). Non-linear approximation on the other hand, requires full knowledge of the function itself. An approximation is obtained by sorting all coefficients from high to low importance, and taking the first *n* coefficients (i.e., *n* most important ones) from this sorted list. A tree approximation has a strong correlation in both "spatial" and coefficient magnitude ordering. However, it still requires a-priori knowledge of the shape of the function to create an approximation.

Usability. During acquisition, the exact form of the reflectance field, and thus individual reflectance functions, is not yet known. This implies that non-linear wavelet approximation, and tree approximation, cannot be used to reduce acquisition duration. Linear approximation, on the other hand can be used, but there is no guarantee that a good approximation is obtained. Ideally, we would like to have a spatial ordering which is also linear in coefficient magnitude ordering. However, such an universal ordering does not exist for general functions expressed in a wavelet basis.

6.5 Progressive Tree Approximations

In this section a progressive tree approximation algorithm is presented. This progressive algorithm will form the basis of the two novel acquisition methods introduced in sections 6.6 and 6.7.

The lossy approximation techniques of the previous section are specifically geared towards data compression. In such a case, the function is known beforehand, and the storage requirements are minimized as much as possible. Progressive tree approximation starts from a different situation, i.e., the function is not completely known beforehand, but individual coefficients can be queried at an unknown cost (and should therefore be avoided as much as possible).

Two variants of progressive tree approximation are presented. *Greedy progressive tree approximation* tries to encode all queried coefficients. It assumes that a coefficient-query is a costly operation in comparison to the cost for storage. *Optimal progressive tree approximation* tries to minimize storage requirements, while minimizing the number of coefficient-queries. It assumes that the cost for a query and storage are in balance. Both methods are based on tree approximation, and have a sub-linear compacting complexity in terms of the original function size, and a linear complexity in terms of stored coefficients.

A key assumption is that, if the magnitude of a wavelet coefficient is large, then there is a large likelihood that its children's coefficients are also large. We define the children of a wavelet as all the wavelet functions of the next level (more detail) that have an overlap with the parent wavelet's (spatial) footprint. It is obvious that this key assumption is not always valid (for example: a function that equals a single (high level) wavelet function). However, in the cases in which this is valid (or at least almost valid) a progressive tree approximation algorithm can be applied. This key idea is supported by the observation that for a large class of functions, a tree approximation is quasi-optimal [3]. Further support is given by DeVore et al. [25], who noted that for natural images (i.e., photographs of real scenes) the wavelet coefficients decay, and that the coefficients tend to cluster in a tree-like manner. This implies that our key idea would be valid for a large class of functions, including natural images.

Greedy Progressive Tree Approximation. A greedy progressive tree approximation can be described as follows. The approximation is initialized by the lowest resolution wavelet coefficients. Next, the approximation is progressively expanded by adding the children of the largest coefficient in the current approximation for which the children have not been added yet (i.e., the leaf nodes in the current tree approximation). This is repeated until the approximation contains enough elements or the magnitude of the largest leaf node in the tree approximation falls below some threshold.

Mathematically, greedy progressive tree approximation of a vector V can be described in a similar manner as non-linear wavelet approximation by using a (progressive) set $\mathtt{s}_s$ to define a wavelet basis transformation $\psi_{\mathtt{s}_s}$:

$$\begin{aligned} \mathtt{s}_1 &= \{Index(\varphi)\} \\ \mathtt{s}_s &= \mathtt{s}_{s-1} \cup Child\left(\arg\max_{i\in(\mathtt{s}_{s-1}-Parent(\mathtt{s}_{s-1}))} ||(v_\psi)_i||_2\right), \end{aligned} \tag{6.2}$$

where $Index(\varphi)$ is the index of the scale function in the wavelet basis ψ (usually the first column), $Parent(\cdot)$ returns the indices of the ancestors, and, $Child(\cdot)$ the indeces of the children. Note that $(\mathtt{s}_{s-1} - Parent(\mathtt{s}_{s-1}))$ is the set of leaf nodes in $\mathtt{s}_{s-1}$ (i.e., all the nodes that don't have child nodes). Furthermore, note that the definition of $\mathtt{s}_s$ is a recursive definition. Each additional element in $\mathtt{s}_s$ depends on the $s-1$ previous elements. Each queried coefficient $(v_\psi)_i$ is included in the set $\mathtt{s}_s$, and thus in the tree approximation, hence the name "greedy". This reflects the assumption that a query of a coefficient is significantly more expensive than the cost of storage.

The order of processing of a greedy progressive tree approximation can be visualized by considering the tree of wavelet coefficients that connects a wavelet coefficient with its direct ancestor's coefficient. Greedy progressive tree approximation starts at the root of this tree, and subsequently refines (i.e., computes the children's coefficients) of the leaf node with the largest magnitude. Note, that for each refinement step, multiple wavelet coefficients are computed. The number of computed wavelet coefficients at each step, depend on the dimensionality of the signal and on the size of the wavelet filter. For example, for $1D$ functions and using a Haar wavelet basis, each wavelet has 2 children, thus 2 wavelet coefficients need to be computed in each refinement step. When using, for example, a Daubechies $D4$ wavelet basis on a $1D$ signal, each wavelet has 4 children, and thus 4 coefficients need to be computed (at most). The order of processing in a greedy progressive tree approximation using the Haar wavelet is illustrated iconically in figure 6.6. In this example, the approximation is initialized by including the root element $(4/1)$. Next, the largest leaf element is searched for in the current approximation. Since only the root is present, this element is further refined, and the elements $2/4$ and $3/2$ are added to the approximation. Again, the largest leaf node is searched for. In this case this is $3/2$, which is further refined, and $3/3$ and $1/6$ are added. And so on...

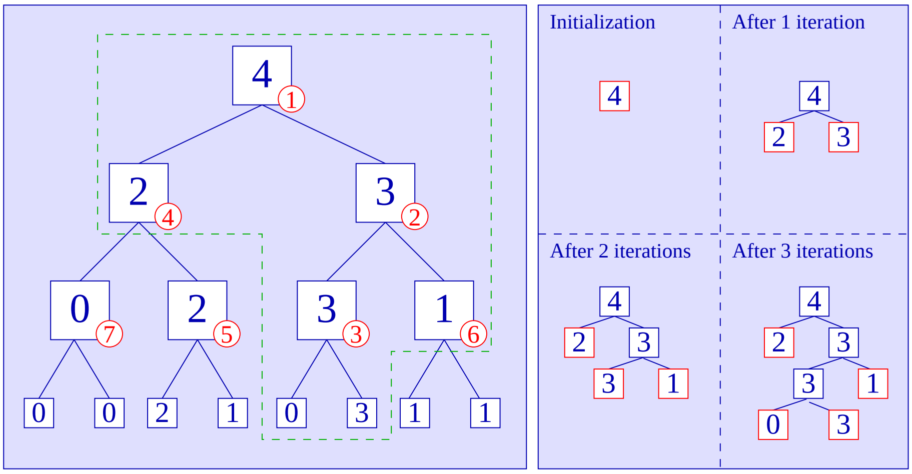
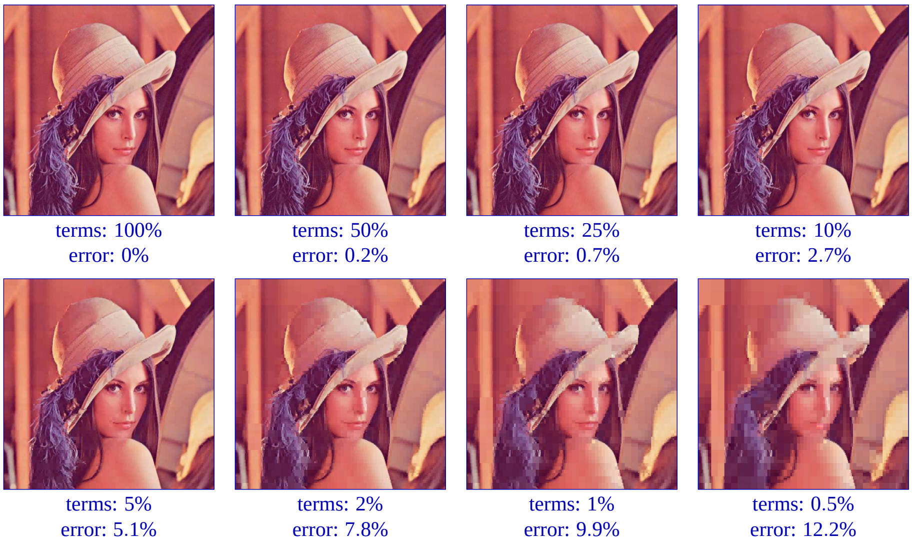


Figure 6.6: Order of processing in a greedy progressive tree approximation using the Haar wavelet. Left: A complete wavelet-tree of a $1D$ signal, discretized in 16 elements is shown. The wavelet coefficients are noted in **blue** in each node. The order of refinement is denoted in **red**. At each refinement step, two new wavelet coefficients (the children's) are computed. The **green** set includes the wavelet coefficients computed after 3 refinement steps (i.e., 7 coefficients). Note, that 2 elements are not yet refined (order-number 4 and 6) after 3 refinement steps. Right: For the first 3 refinement steps, the tree approximations are shown (without order number). The red marked elements are the leaf nodes (of the tree approximation) from which the largest element is selected for refinement in each iteration step.

Figure 6.7 illustrates greedy progressive tree approximation applied to the Lena image. At high bit rates greedy progressive tree approximation does not perform well in comparison to non-linear wavelet approximation (figure 5.3). The main reason for this is that non-zero coefficients outside a tree structure are (almost) never included, resulting in some lower bound on the approximation error. At low bit rates (i.e., less than 0.5%) the approximation is about 3% less efficient in terms of error compared to non-linear wavelet approximation.

Optimal Progressive Tree Approximation. An optimal progressive tree approximation is more selective as to which coefficients are added to the approximation than the greedy approach. The basic idea is to add the largest of the leaf node's children. The greedy approximation added **all** children of the largest leaf node, whereas the optimal approximation only adds a **single** coefficient that is the largest of all the children of all the leaf nodes.

Mathematically, optimal progressive tree approximation of a vector V can be described in a similar manner as greedy progressive tree approximation by using a progressive set $\mathfrak{s}_s$ to define a wavelet basis transformation $\psi_{\mathfrak{s}_s}$:



terms: 100%
error: 0%

terms: 50%
error: 0.2%

terms: 25%
error: 0.7%

terms: 10%
error: 2.7%

terms: 5%
error: 5.1%

terms: 2%
error: 7.8%

terms: 1%
error: 9.9%

terms: 0.5%
error: 12.2%

Figure 6.7: Greedy progressive Haar wavelet tree approximation applied to the Lena image. A 512 × 512 image is compressed using a greedy progressive Haar wavelet tree approximation. For each example the relative number of terms and the relative error are shown.

$$\begin{aligned} \mathtt{s}_1 &= \{Index(\varphi)\} \\ \mathtt{s}_s &= \mathtt{s}_{s-1} \cup \arg \max_{i \in (Child(\mathtt{s}_{s-1}) - \mathtt{s}_{s-1})} ||(v_\psi)_i||_2. \end{aligned} \tag{6.3}$$

Note the different location of the operator $Child(\cdot)$ compared to equation (6.2). $(Child(\mathtt{s}_{s-1}) - \mathtt{s}_{s-1})$ is the set of all the children of all the leaf nodes. The order of processing is illustrated in figure 6.8. In this example, the tree approximation is initialized by the root node $4/1$. Next, the largest of all leaf nodes $\{2/5, 3/2\}$ of the current approximation is added, $3/2$ in this case. At this point, the list of candidate leaf nodes contains: $\{$2/5,3/3,1/8$\}$. From this set, the largest node (i.e., $3/3$) is added to the tree approximation. This is repeated until some stop criterion is reached.

An important difference between equations (6.2) and (6.3) is that, in the case of the greedy approximation the maximum is computed over coefficient magnitudes $||(v_\psi)_i||_2$ that are already in the tree approximation, while for the optimal approximation they are not. The assumption was made that a query for a coefficient $(v_\psi)_i$ is a relatively expensive operation. Therefore, it is important to minimize the number of queries, and to cache queried coefficients as much as possible. For the greedy algorithm this implies that the maximum is computed on coefficients already in the approximation, and a query is only performed when adding new children coefficients to the tree approximation. In case of the optimal progressive tree approximation, an explicit cache needs to be maintained. This can be achieved by introducing a priority queue $\mathtt{p}$, and changing equation (6.3) as follows:

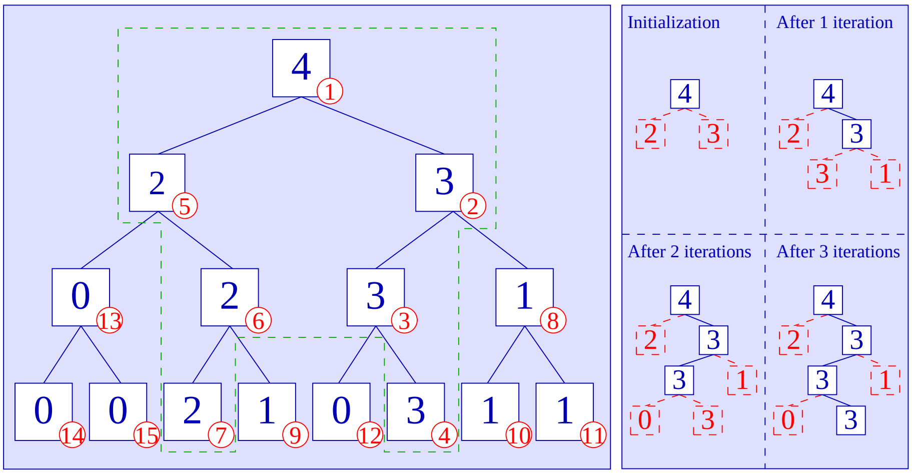
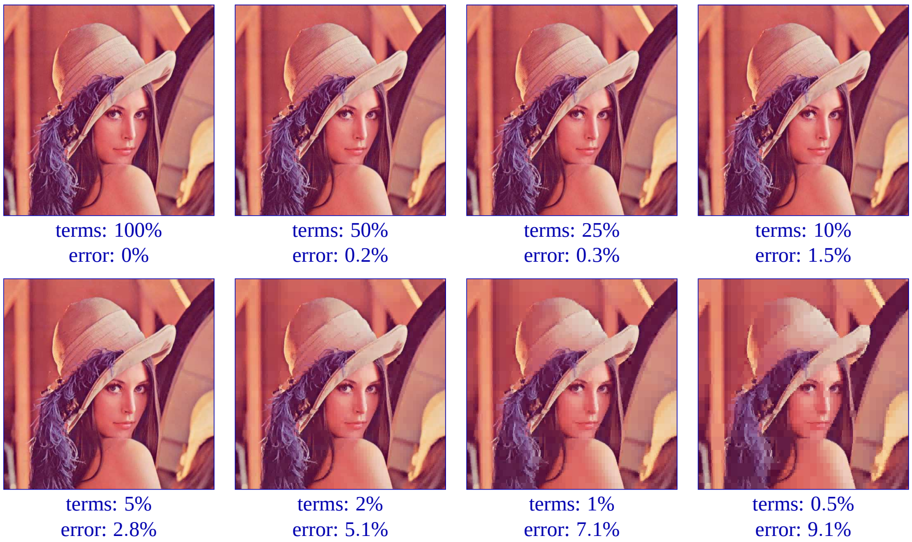


Figure 6.8: Order of processing in an optimal progressive tree approximation using the Haar wavelet. Left: A complete wavelet-tree of a $1D$ signal, discretized in 16 elements is shown. The wavelet coefficients are noted in **blue** in each node. The order of refinement is denoted in **red**. At each refinement step, a single new wavelet coefficient is added. The **green** set are the wavelet coefficients computed after 7 refinement steps. The green set is equal is size then the marked set in figure 6.6. However, the approximation error is less than in the greedy approach. Right: For the first 3 refinement steps, the tree approximations are shown. For each approximation, the list of candidates nodes, from which a coefficient is added in the next refinement step, are shown in **red**. These red candidate nodes are **not** part of the optimal tree approximation.

$$
\begin{aligned}
\mathfrak{p}_0 &= \{Index(\varphi)\} \\
\mathfrak{s}_0 &= \emptyset \\
\mathfrak{p}_s &= \mathfrak{p}_{s-1} \cup Child\left(\arg\max_{i\in\mathfrak{p}_{s-1}} ||(v_\psi)_i||_2\right) - \arg\max_{i\in\mathfrak{p}_{s-1}} ||(v_\psi)_i||_2 \\
\mathfrak{s}_s &= \mathfrak{s}_{s-1} \cup \arg\max_{i\in\mathfrak{p}_{s-1}} ||(v_\psi)_i||_2 .
\end{aligned}
\tag{6.4}
$$

The priority queue contains all the children of the leaf nodes of the tree approximation. By only performing a query operation when adding children to the priority queue, the number of queries is minimized. To refine the set $\mathfrak{s}_{s-1}$, no new queries are made, but the requested coefficient is retrieved directly from the priority queue. Figure 6.9 illustrates optimal progressive tree approximation applied to the Lena image. In general, the optimal progressive tree approximation performs better than greedy progressive tree approximation (figure 6.7). At high bit rates optimal progressive tree approximation does not perform well in comparison to non-linear wavelet approximation (figure 5.3). The main reason for this is that non-zero coefficients with a zero parent coefficient are (almost) never included, resulting in some lower bound on the approximation error. At low bit rates (i.e., less than 0.5%) the approximation is about $\sim 0.7\%$ less efficient in terms of error compared to a non-linear approximation. This proves



terms: 100%
error: 0%

terms: 50%
error: 0.2%

terms: 25%
error: 0.3%

terms: 10%
error: 1.5%

terms: 5%
error: 2.8%

terms: 2%
error: 5.1%

terms: 1%
error: 7.1%

terms: 0.5%
error: 9.1%

Figure 6.9: Optimal progressive Haar wavelet tree approximation applied to the Lena image. A 512×512 image is compressed using a optimal progressive Haar wavelet tree approximation. For each example the relative number of terms and the relative error are shown.

empirically that the efficiency of the optimal progressive tree approximation is a close match with that of a non-linear wavelet approximation for low bit rates.

Approximation Error In general the approximation quality of a progressive tree approximation will be similar to a normal tree approximation. Furthermore, the quality of an optimal progressive tree approximation will be better than that of a greedy progressive tree approximation. The quality of the approximation is superior to linear approximation, and can for a large number of function classes approach the quality of a non-linear approximation. To obtain an equal quality as in a non-linear wavelet approximation (assuming an optimal progressive tree approximation), the input vector must meet the following condition:

$$||(v_\psi)_i||_2 \leq ||(v_\psi)_{Parent(i)}||_2,$$

for any $i \in \mathtt{s}_s$. Or in words, the wavelet coefficients should be monotonically decreasing along the branches of the wavelet tree. Smooth functions (e.g., in a Besov space) and natural images [25] meet this condition, and thus can be approximated without loss of quality by an optimal progressive tree approximation.

A large class of functions satisfy to a related condition:

$$weight(Level(\psi_{\cdot,i})) \cdot ||(v_\psi)_i||_2 \leq weight(Level(\psi_{\cdot,i}) - 1) \cdot ||(v_\psi)_{Parent(i)}||_2, \quad (6.5)$$

where *weight*$(\cdot)$ is a weighting function taking a wavelet level as parameter. An example of a class of functions that follow this rule are non-oscillating[1] amplitude limited functions (i.e., $-c \leq \max f(x) \leq c$). An efficient approximation, with respect to a **weighted** norm, is possible in such a case using an optimal progressive tree approximation. Note that general reflectance functions fall in this latter category because of the law of energy conservation (i.e., a reflectance function cannot reflect more light than it receives).

6.6 Explicit Sampling

In this section we introduce a novel method to acquire an approximation of the reflectance field of a scene in a sub-linear complexity based on a greedy progressive tree approximation. This method was first presented in [74]. This section is structured as follows: first the basic idea is detailed on a single reflectance function (subsection 6.6.1). Next, this is expanded to include the acquisition of the complete reflectance field in subsection 6.6.2. The complete acquisition process is finally presented in subsection 6.6.3. Some practical considerations are mentioned in subsection 6.6.4. Finally, some results (subsection 6.6.5) and the method itself are discussed in subsection 6.6.6.

6.6.1 Basic Idea

Let's for the moment consider only a single reflectance function, and try to extend the conclusions afterwards to a complete reflectance field. As in chapter 5, given a progressive tree approximation set s_s, the resulting basis ψ_{s_s} can be used to compress a reflectance function $\mathbf{T}_i$:

$$\mathbf{T}_i \quad \approx \quad (\mathbf{T}_i \psi_{\mathsf{s}^s}) \overline{\psi_{\mathsf{s}^s}}^T .$$

In the case that $s = l$ (the resolution of the incident light field) then the basis transformation ψ_{s_l} is equivalent to ψ, but in which the columns are permutated. However, as mentioned before, if the size of the incident illumination becomes too large, capturing the whole reflectance field becomes impractical. Therefore, we prefer the size of the final set s_s to be significantly smaller than the size of the incident illumination. We will use a progressive tree approximation to achieve this goal.

Due to the progressive nature of greedy progressive tree approximation, a basic algorithm (see figure 6.10) can be formulated based on the hierarchical set defined in equation (6.2). This algorithm works as follows: First, the lowest resolution wavelet basis function is emitted, from the CRT monitor, onto the scene and the pixel response is observed (i.e., by taking an HDR photograph and isolating the relevant pixel). This observed wavelet coefficient is added to the current tree approximation of the pixel's reflectance function. Next, from the current approximation, the subsequent wavelet

[1] With *non-oscillating functions* we refer to functions that do not contain isolated wavelet-like structures.

```
Init:   $s = 1$, $\mathsf{s}_s = \{Index(\varphi)\}$, wavelet basis $\psi$

while($s <$ maximum number of coefficients)
{
    Emit: the wavelet basis function $\psi_{\cdot,\mathsf{s}[s]}$
    Observe: $(c^{(s)})_i = \mathbf{T}_i \cdot \psi_{\cdot,\mathsf{s}[s]}$
        (i.e., i-th pixel in an HDR photograph)
    Compute: $\mathsf{s}_{s+1}$ based on $\left\{ (c^{(1)})_i, \ldots, (c^{(s)})_i \right\}$
    $s \leftarrow s+1$
}

Output: tree approximation $\left[ (c^{(1)})_i \,|\, \ldots \,|\, (c^{(s)})_i \right]$ of $\mathbf{T}_{\psi^{\mathsf{s}_s}}{}_i$
```

Figure 6.10: Greedy progressive tree approximation on a single reflectance function $\mathbf{T}_i$.

basis functions to emit are determined. This is achieved by identifying the wavelet basis function that resulted in the largest observed wavelet coefficient magnitude among all leaf nodes (i.e., non-refined wavelet basis functions) in the current tree approximation. Of this wavelet basis function, the children wavelet basis functions are emitted, and the observed coefficients are added to the current tree approximation. This is repeated until some stop criterion is met.

This algorithm can be efficiently implemented by explicitly constructing a tree as in figure 6.6, or by using a priority queue (i.e., the observed wavelet coefficients $(c^{(j)})_i$ (under the j-th wavelet illumination condition) are placed on a priority queue. At each iteration, the top of the priority queue is removed. The children of this top-element are the elements to query subsequently).

6.6.2 Extension to Reflectance Fields

There are two problems that need to be solved before the algorithm given in the previous subsection can be practically used:

1. A reflectance function corresponds to a single camera pixel. However, a camera records multiple pixels in parallel in a single photograph, and thus gains information from multiple reflectance functions at once. Therefore, the question arises: "how to extend this algorithm to capture the complete reflectance field, and not only a single reflectance function".
2. How to ensure that the greedy progressive tree approximation is well-behaved such that it converges to an as optimal as possible approximation?

Reflectance fields (1). Reconsider the basis equation (3.1): $C = \mathbf{T}L + S$. This equation describes what happens if some form of incident illumination is applied to a scene. In the algorithm described above, different wavelet basis functions are emitted onto the scene, and appropriate action is taken with respect to the observed wavelet coefficients. When capturing an HDR photograph of the scene illuminated by a progressively selected wavelet basis function, not only the wavelet coefficient of that particular reflectance function (on which the progressive order is computed) is observed, but also the wavelet coefficients of all other reflectance functions with respect to the emitted wavelet basis function.

It is important to realize, that the set $\mathtt{s}_s$ of the tree approximation is only optimal for a single reflectance function, and not for *all* reflectance functions. It is obvious that creating the set $\mathtt{s}_s$ with respect to a single reflectance function, is not the best way to create a progressive tree approximation for the whole reflectance field. Instead it would be better to base the choice of the subsequent emitted wavelet basis function on the previously observed wavelet coefficients of all reflectance functions. The p-norm of the exitant illumination is a good candidate to base this decision on. The p-norm of a vector V is defined as:

$$||V||_p = \left(\sum_{i=1}^{l} |v_i|^p \right)^{\frac{1}{p}} .$$

A p-norm of the observed image can play a similar role as the weight of a single coefficient magnitude in algorithm 6.10. A refined acquisition algorithm using the p-norm is given in figure 6.11. This algorithm works in a similar manner as algorithm 6.10, but instead of observing a single pixel's coefficient, the wavelet coefficients of all pixels are observed in parallel. The decision to what wavelet basis functions to emit next, is now based on the p-norm of the observed images.

Conditions (2). In section 6.5 the approximation errors for a progressive tree approximation are discussed. A critical condition for a well-behaved progressive refinement method is the decay of the wavelet coefficients along the branches of the wavelet tree. As the algorithm is currently formulated (figure 6.11), there is no guarantee that this is the case. Given the fact that reflectance functions are amplitude-limited, and thus the condition in equation (6.5) holds, a suitable weighting must be defined. There are two possible solutions to achieve this:

1. **Weighting of the observed wavelet coefficient images.** The wavelet coefficients (images) could be weighted, such that the weighted wavelet coefficients decay with advancing wavelet level. However, the effect on the complete system must be evaluated carefully. The obtained progressive tree approximation is obviously not optimal anymore with respect to the L_2-error, but to the weighted norm.

2. **Scaling of the wavelet basis functions.** By selecting a different normalization on the wavelet basis functions a similar effect as above can be obtained. It is important to realize, that this also implicitly results in a different error norm.

```
Init:   $s = 1$, $\mathtt{s}_s = \{Index(\varphi)\}$, wavelet basis $\psi$

while($s <$ maximum number of coefficients)
{
    Emit: the wavelet basis function $\psi_{\cdot,\mathtt{s}[s]}$
    Observe: $C^{(s)} = \mathbf{T}\psi_{\cdot,\mathtt{s}[s]}$
        (i.e., recording an HDR photograph)
    Compute: $\mathtt{s}_{s+1}$ based on $\left\{||C^{(1)}||_p, ..., ||C^{(s)}||_p\right\}$
    $s \leftarrow s+1$
}

Output: tree approximation $\left[C^{(1)}|...|C^{(s)}\right]$ of $\mathbf{T}_{\psi^{\mathtt{s}_s}}$
```

Figure 6.11: Greedy progressive tree approximation on the complete reflectance field.

For this technique we will use the first method (weighting of the observed wavelet coefficient images) to ensure a well-behaved progressive tree approximation. We show that by restricting the properties of the incident illumination used afterwards for relighting, a perfect progressive tree approximation with respect to the L_2-error on the relit results is obtained, regardless of the non-optimal error on the reflectance field due to the weighting. Furthermore, we will show that both approaches, weighting and scaling the basis functions, are identical.

As mentioned before, DeVore et al.[25] noted that for natural images (i.e., photographs of real scenes) wavelet coefficients decay, and that this decay is dependent on the level v or resolution of the wavelet, the local order of continuity t of the image, and the number of dual vanishing moments[2] d of the wavelet:

$$decay \sim 2^{-v \max(t,d)}.$$

Intuitively, this implies that the energy content decays over different wavelet levels, and thus the importance of details in a natural image is directly proportional to the spatial size of the detail. It is reasonable to assume that incident illumination also behaves as a natural image. Dror et al. [28] investigated the statistics of real-world illumination, and conclude that the statistics are very similar to that of natural images, with the exception of bright point light sources such as the sun. This assumption allows us to define an upperbound for the wavelet coefficients of any incident illumination in function of the wavelet level v:

$$weight(v) = c_r\, 2^{-v \times c_s}, \tag{6.6}$$

[2] the order of polynomials that can be approximated by the *dual* scaling functions of the wavelet.

where c_s is a constant (≥ 1) indicating the general smoothness of the incident illumination and of the specific wavelet basis used. c_r is a constant (> 0) related to the dynamic range of the considered incident illumination. In this work we set $c_s = 1$, however, if it is a-priori known that the incident illumination or the wavelet basis are smooth, then a larger c_s could be used. Selecting a larger constant c_s favors wavelet basis functions with low level (low frequency wavelet functions) over wavelet basis functions with a high level (high frequency wavelet functions).

Given a wavelet basis ψ, a weighting vector W can be defined, in which each element is determined by $w_i = weight(Level(\psi_{\cdot,i}))$. Since L is similar to a natural image, equation (6.6) holds, and thus $w_i \geq ||\overline{\psi}^T L||_2$. The upperbound on the p-norm of any exitant illumination (i.e., camera image) can be derived from the general relighting equation which relates any incident illumination L to the observed exitant illumination C:

$$\begin{aligned} ||C||_p &= ||\mathbf{T}L||_p \\ &= ||\mathbf{T}\psi(\overline{\psi}^T L)||_p \\ &\leq ||\mathbf{T}\psi W||_p \\ &\leq \sum_i ||w_i(\mathbf{T}\psi)_{\cdot,i}||_p \\ &= \sum_i ||w_i \mathbf{T}\psi_{\cdot,i}||_p \\ &= \sum_i w_i ||\mathbf{T}\psi_{\cdot,i}||_p. \end{aligned} \tag{6.7}$$

In other words, the p-norm of any relit image is bounded by the sum of weighted p-norms of the observed effects of emitting the wavelet basis vectors onto the scene.

As noted before, weighting the p-norms of the observed exitant illumination under wavelet incident illumination instead of using the p-norm directly yields a well-behaved progressive tree approximation algorithm because of equation (6.5), and the error on the reflectance field is optimized to the weighted p-norm. However, from equation (6.7) it follows that the error on the relit image (under natural illumination) is optimal under any p-norm (including the L_2-norm), since this error is bounded by the sum of weighted p-norms. The goal is to create relit images, therefore the first (i.e., error on the reflectance field) is of lesser consequence than the latter (i.e., error on the relit images).

Now consider the relation between weighting the p-norm and using a different wavelet normalization. There are two kinds of common wavelet normalizations:

1. **Constant energy.** All wavelets have the same energy. (high-pass Nyquist gain = $\sqrt{2}$, and low-pass DC gain = $\sqrt{2}$). This normalization is required if an orthogonal wavelet basis is needed (assuming that the shape of the wavelet allows this).

```
Init:   $s = 1$, $\mathtt{s}_s = \{Index(\varphi)\}$, wavelet basis $\psi$

while($s <$ maximum number of coefficients)
{
    Emit: the wavelet basis function $\psi_{\cdot,\mathtt{s}[s]}$
    Observe: $C^{(s)} = \mathbf{T}\psi_{\cdot,\mathtt{s}[s]}$
        (i.e., recording an HDR photograph)
    Compute: $\mathtt{s}_{s+1}$ based on $\left\{ ||C^{(1)}||_p^w, \ldots, ||C^{(s)}||_p^w \right\}$
    $s \leftarrow s + 1$
}

Output: tree approximation $\left[ C^{(1)} | \ldots | C^{(s)} \right]$ of $\mathbf{T}_{\psi^{\mathtt{s}_s}}$
```

Figure 6.12: Explicit sampling of a wavelet reflectance field using a greedy progressive tree approximation algorithm and a weighted *p*-norm.

2. **Constant amplitude.** All wavelets have approximately the same amplitude. (high-pass Nyquist gain = 2, and low-pass DC gain = 1).

The observations of DeVore et al. [25] regarding the decay of the wavelet coefficients of natural images are made under a constant energy normalization. The relation between a constant energy and a constant amplitude normalization is identical to equation (6.6) in which c_s and c_r are equal to 1. Note that $c_s = 1$ is the worst case, and c_r is a constant for all wavelet level and does not affect the ordering of the observed wavelet coefficients or *p*-norms. This shows that using a constant amplitude normalization is equivalent (although a bit more restrictive) than using a weighting function.

From the *constant amplitude* normalization is it very easy to prove the decay of the observed wavelet coefficients. First, consider the fact that a single reflectance function cannot reflect more light than it receives. This follows directly from the conservation of energy. This implies that the energy in a reflectance function is bounded in some interval[3]. Furthermore, this implies that elements in the reflectance functions are also bound, and thus consequently, given the fact that the wavelet basis functions are normalized to a constant amplitude, the wavelet coefficients are also bounded.

6.6.3 Feedback Loop

As mentioned before, we will use a weighting of the p-norms of the observed images C. This is denoted by: $||\cdot||_p^w$. The final algorithm can now be seen in figure 6.12. This algorithm works as follows: First, low frequency wavelet basis functions are emitted (from a CRT monitor) onto the scene, and HDR photographs are recorded. Depending

[3] the constant c_r of the weighting function (equation (6.6)) actually reflects this, and is the upperbound on the energy content

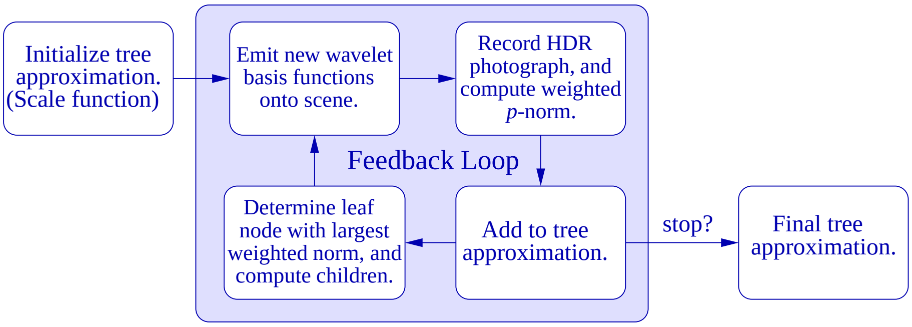

Initialize tree approximation. (Scale function)
Emit new wavelet basis functions onto scene.
Record HDR photograph, and compute weighted p-norm.
Feedback Loop
Determine leaf node with largest weighted norm, and compute children.
Add to tree approximation.
stop?
Final tree approximation.

Figure 6.13: A flow-chart of the complete explicit sampling acquisition process.

on the weighted *p*-norm of the observed images in the current tree approximation, the subsequent wavelet basis functions to emit are determined. This decision is made by selecting the children of the wavelet basis functions (from the leaf nodes of the current tree approximation) that has the largest weighted *p*-norm. This process is repeated until some stop criterion is reached. Possible stop criteria are:

- **Acquisition time limit.** Each additional recorded HDR photograph requires extra acquisition time. In some cases it might be advantageous to set a time-limit for the acquisition.

- **Maximum number of wavelet coefficients**. Related to the previous, an upper limit on the number of emitted wavelet illumination patterns can be set. This also implies that, during relighting, the incident illumination is only approximated by the same set of wavelet basis functions.

- **Weighted p-norm threshold.** Instead of limiting the acquisition time or the number of wavelet coefficients, the acquisition process can be stopped when the weighted p-norm falls below some threshold. This weighted p-norm is an indication of the error on the approximation of the reflectance field.

In practice, the algorithm in figure 6.12 uses a feedback loop. It iteratively emits a wavelet basis function $\psi_{\cdot,\mathbf{s}[i]}$, records an HDR photograph $C^{(i)}$, analyses this photograph, and decides what wavelet basis to emit subsequently. Recording an HDR photograph $C^{(i)}$ corresponds to a coefficient query operation in the progressive tree approximation algorithm of section 6.5. This is a very expensive operation in comparison to storing an HDR photograph $C^{(i)}$, because an HDR photograph needs to be recorded, which can take multiple seconds. Therefore a greedy progressive tree approximation algorithm is used as a basis for this algorithm rather than an optimal progressive tree approximation algorithm. Figure 6.13 illustrates the complete acquisition process .

6.6.4 Practical Considerations

There are some practical issues that need to be considered before the feedback loop algorithm of the previous subsection can be implemented.

P-norm. Any value can be used in theory for p when computing the p-norm. In our implementation p is set to 2. In other words, we use the squared norm L_2. The L_2-norm weights low radiance values (which are more susceptible to noise) less. This improves the stability of the feedback loop with respect to measurement noise.

2D wavelet basis functions. In the feedback loop $2D$ wavelet patterns are emitted. This creates some ambiguity in the definition of a parent/child of a wavelet. Previously, we defined the children of a wavelet as the wavelets of the next level that have a footprint that overlaps with the footprint of the parent wavelet. $2D$ wavelet basis functions also have an "orientation", i.e., different $1D$ scale, and, $1D$ wavelet basis function combinations. There are 3 orientations associated with each wavelet "location". In our implementation, a weighted p-norm is associated with a wavelet location. For each location, the contributions (weighted p-norms) of the orientations are summed. This means that at each feedback step, when the coefficients of a wavelet location are queried, all 3 orientations need to be emitted, and their responses acquired.

Emitting Wavelet Illumination Patterns. The dynamic range of a wavelet illumination pattern usually does not fit within the range of the emitter, nor is the range of the emitter linear in radiance space. Scaling the wavelet patterns (and doing the inverse scaling on the resulting observed photographs) solves the first problem, whereas calibrating the emitter solves the second.

Before emitting a pattern, the gamma curve of the emitter needs to be inversely applied in order to transform the non-linear range of the emitter to a linear range in radiance space. This was already discussed in 6.3.

Also, wavelets have positive and negative values. Therefore we need to map each wavelet illumination pattern to a completely positive range, since emitting negative light is not possible. Let's assume that the scaled wavelet $\psi_{\cdot,i}$ has a range of $[-1,+1]$ and the range of the calibrated emitter is linear in radiance space $[0,1]$. Two mappings are possible:

1. Translating the wavelet and scaling it: $\psi'_{\cdot,i} = \frac{\psi_{\cdot,i}+1}{2}$,
2. Splitting it into two patterns $\psi_{p_{\cdot,i}}$ and $\psi_{n_{\cdot,i}}$ that contain respectively the positive and negative part of $\psi_{\cdot,i}$.

The resulting photograph $C_{\psi_{\cdot,i}}$ when illuminating the scene by the wavelet illumination pattern $\psi_{\cdot,i}$ can be reconstructed as follows:

1. $C_{\psi_{\cdot,i}} = 2C_{\psi'_{\cdot,i}} - C_1$, where C_1 is the observed result of illuminating the scene by emitting a full white pattern from the CRT monitor.

2. or, respectively, $C_{\psi_{\cdot,i}} = C_{\psi_{p\cdot,i}} - C_{\psi_{n\cdot,i}}$.

The latter approach, which we use, has some practical advantages. First, exposing a CRT monitor for a long period with the same color introduces significant extra noise caused by the afterglow from these pixels. Second, the dynamic range of an emitted wavelet pattern is doubled at the cost of an extra HDR photograph.

Directly Visible Pixels. Directly visible backdrop pixels from the emitter should not be included in the computation of $||C_{\psi_{\cdot,i}}||_p^w$, because these pixels will behave as a perfect specular material, and will influence the performance of the acquisition negatively (see also subsection 6.6.6, discussion on the acquisition of specular materials). An alpha-matte is computed in order to exclude these directly visible elements. This alpha-matte is constructed using the method proposed by Zongker et al. [119]. The overhead of recording these extra photographs of progressively finer stripe patterns is minimal. Uncovered pixels in the matte are replaced in the final image by the corresponding incident illumination values, by using the correspondence between CRT pixels and camera pixel (see section 6.3).

Stop Criterion. In our experiments, we use an acquisition time limit of 12 hours for each scene as the stop criterion in the feedback loop. The total recording time could be improved by using a better synchronization between the digital camera (Canon EOS D30) and the feedback loop. The use of a digital video camera can reduce the time to capture a reflectance field even more.

6.6.5 Results

The result of the acquisition process is a set of recorded HDR photographs $C^{(i)} = \mathbf{T}_{\psi^{\mathtt{s}_s}{}_{\cdot,i}}$, and a set $\mathtt{s}_s$. A relit image can be easily computed by using equation (5.2) combined with equation (5.3):

$$C = \mathbf{T}_{\psi^{\mathtt{s}_s}} L_{\overline{\psi^{\mathtt{s}_s}}} + S.$$

In other words: First, the novel incident illumination L is decomposed into the wavelet basis $\overline{\psi}$. Next, the observed images $C^{(i)}$ are weighted by the $\mathtt{s}_s[i]$-th element of the decomposed incident illumination $L_{\overline{\psi}}$, and summed to form the relit image C. This is very similar to the relighting process used by Debevec et al. [20], described in section 2.3.

In contrast to other environment matting methods, using a similar setup, the presented method can handle diffuse surfaces. This can be seen in figure 6.14 where four different colored cubes are placed on a diffuse surface, illuminated by a low frequency incident light field. Approximately 400 wavelet illumination patterns are used to capture the reflectance field of the depicted scene. In figure 6.15, the same scene is shown, illuminated from a small square located respectively left, middle, and right behind the cubes. As can be seen, the shadows are faithfully reproduced, something that would

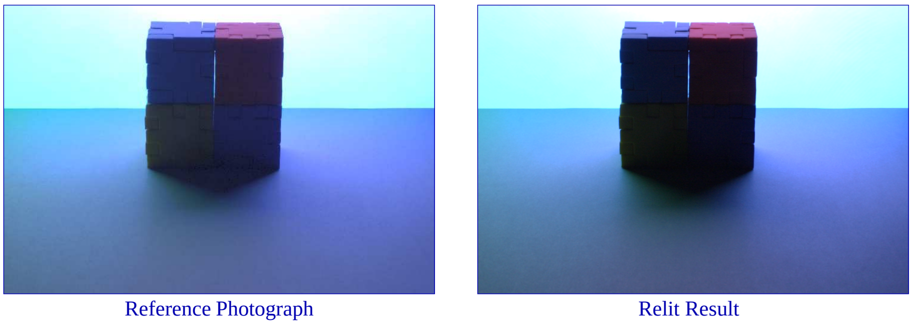


Figure 6.14: A scene containing diffuse surfaces, acquired by emitting only 400 Haar wavelet illumination patterns. On the right, the scene is relit using a low frequency incident illumination field. On the left a reference photograph of the scene under the same illumination emitted from the CRT monitor.

have been impossible to achieve using simple filter functions as in other environment matting methods.

In figures 6.16 and 6.17, a scene containing a glass candy jar filled with little candy bears is depicted, relit by a photograph of a river and a street as incident illumination. The environment matte is captured using 2400 HDR photographs (or 1200 Haar wavelet illumination patterns split in a negative and positive part). In both examples, a reference figure is shown on the left. The slight mismatch in colors between the reference photograph and the relit result is caused by a non-optimal color calibration of the CRT monitor. Figure 6.18 shows a small number of observed responses of the glass bear scene under wavelet illumination. Positive values are indicated in blue, while negative values are marked in yellow. Figure 6.19 illustrates the effect of capturing the reflectance field with fewer terms (i.e., taking fewer photographs and thus using fewer wavelet illumination patterns). The glass bear scene is captured using, from left to right, 100, 300, 600, and, 900 wavelet illumination patterns respectively. To give a better idea of the impact of using fewer terms, the directly visible pixels are not replaced by the correct pixels from the incident illumination nor is the ambient illumination term S added.

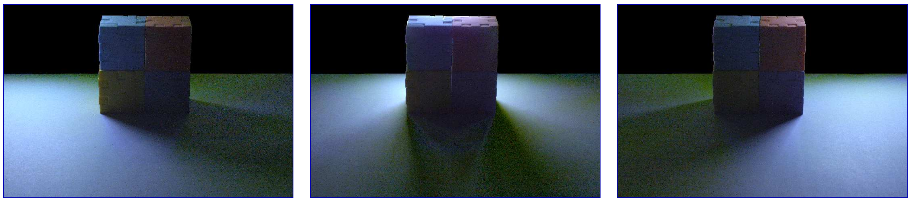


Figure 6.15: Relit images illustrating faithful reproduction of shadows. The incident illumination consists of a square light source positioned respectively left, middle, and right behind the colored cubes.

Figures 6.20 and 6.21 show the effect of using a different wavelet than the Haar wavelet. Using a higher order wavelet, e.g., the 9/7 Daubechies wavelet, can result in a smoother approximation. Using these smooth wavelets yield more pleasing results if the number of photographs is very limited with respect to the resolution of the incident illumination, as opposed to the Haar wavelet which yields blocky results. The advantages of these higher order wavelets become less obvious when the number of recorded wavelet illumination patterns increases. The reason is that these higher order wavelets require more coefficients to represent high frequency details, and thus require to observe more high resolution wavelets to represent these fine details. This number of (high resolution) wavelet patterns quadruples with every increase in level. Again, the slight mismatch in colors in these results between the reference photograph and the relit results is caused by a non-optimal color calibration of the CRT monitor.

The acquired examples require on average 2.5GB to store all photographs (RLE compressed). Using more advanced compression algorithms (e.g., JPEG2000) could further reduce the required storage.

6.6.6 Discussion

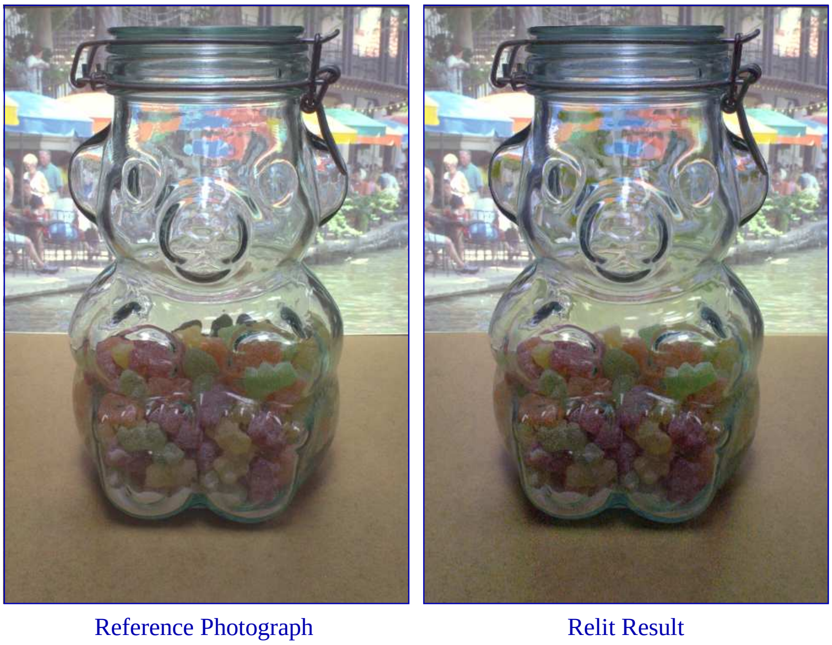
We conclude this section on explicit sampling of wavelet represented reflectance functions with an analysis of the error bound on the relit images, and a review of the limitations of this method.

Error bound. There exists an interesting relation between the p-norm of $C_{\psi_{\cdot,i}}$ and of $\psi_{\cdot,i}$:

$$||C_{\psi_{\cdot,i}}||_p \leq c\,||\psi_{\cdot,i}||_p.$$

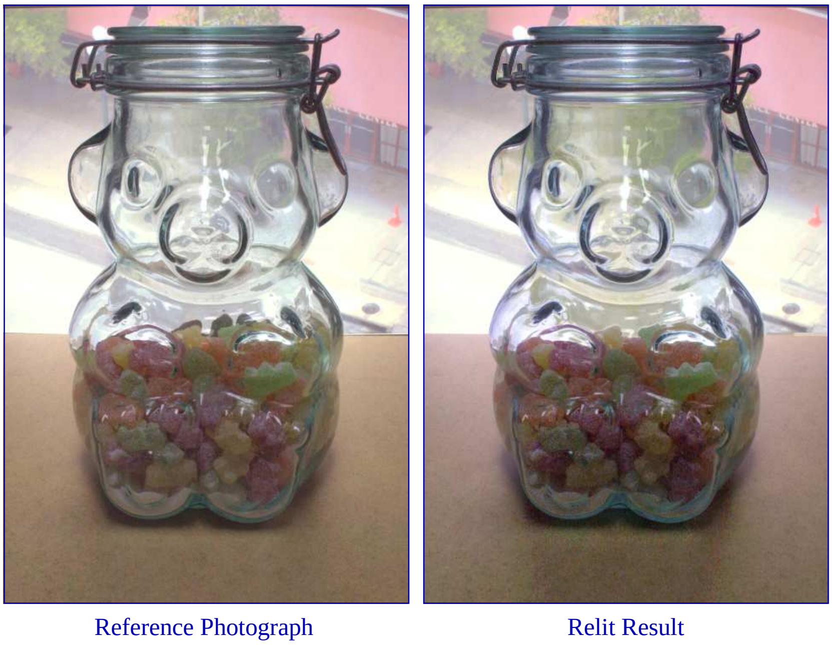
This means that the p-norm on an observed image of **any** scene under wavelet illumination, is bounded by the energy (according to the p-norm) of the wavelet illumination (times a constant c). This is a direct result of the fact that a material cannot reflect more light than it receives (the constant c expresses the relation between the amount of light emitted and the magnitude of a single element in the incident light field vector L). Together with equation (6.7) this results in an important observation: the error on the relit image C is bounded by the error on the approximation of the incident illumination L. In other words, it is an upper-boundary for the error on C and is, in general, an overestimation of the real error. It also implies that increasing the number of emitted wavelet basis functions will have a positive effect on the error of C, and in the limit this error will vanish.

Limitations. The number of wavelet illumination patterns required to obtain an acceptable approximation of the reflectance field of a scene is in general limited (e.g., 400). However, the number of required wavelet illumination patterns, and thus HDR



Reference Photograph Relit Result

Figure 6.16: A glass candy jar filled with candy bears. Left: Reference photograph. Right: Relit result of the scene captured with 1200 Haar wavelet illumination patterns.



Reference Photograph Relit Result

Figure 6.17: Identical to figure 6.16, but relit with a different incident light field.

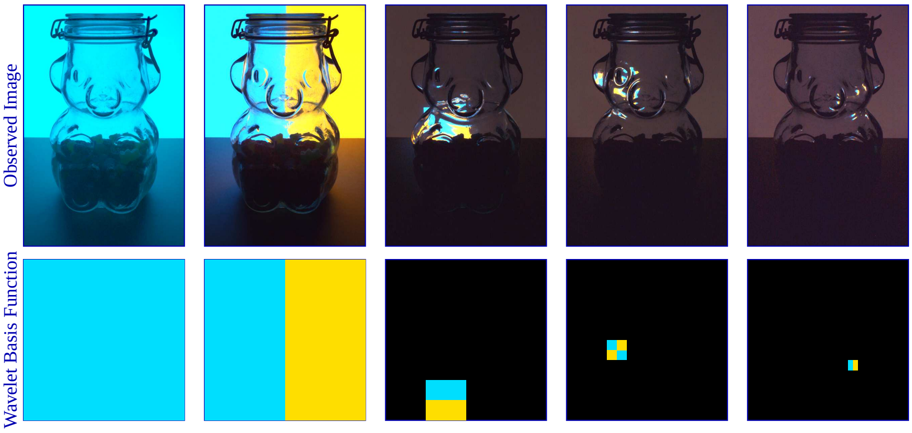


Figure 6.18: An small subset of observed responses under wavelet illumination patterns. Positive values are given in blue, while yellow indicates a negative value. For each example, the emitted wavelet basis function is shown below each of the observed responses.

photographs which need to be recorded, does not scale well when the scene contains specular objects. A small footprint wavelet illumination pattern contributes significantly to only a few (specular) pixels in the recorded photograph, and thus yields a limited gain in information for the reflectance field approximation. This is visible in figure 6.18, where fewer pixels are lit in the image illuminated by high frequency wavelet basis functions. Furthermore, the amount of light reflected from the object into the camera is closely related to the footprint size of the wavelet basis function,

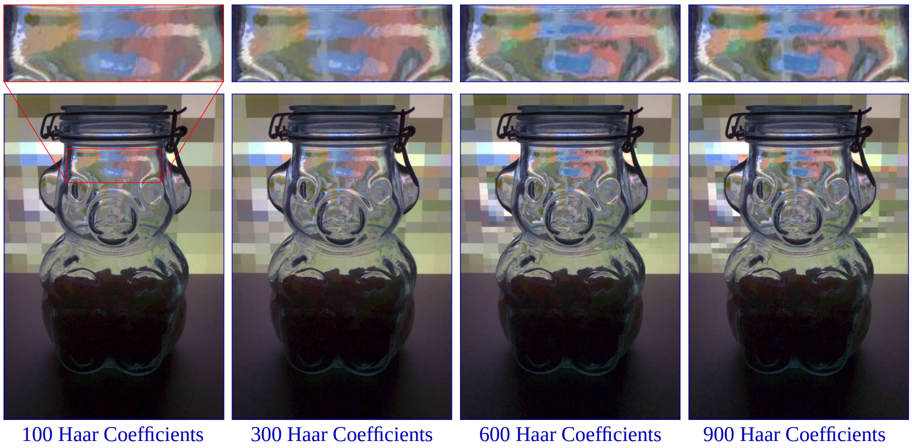
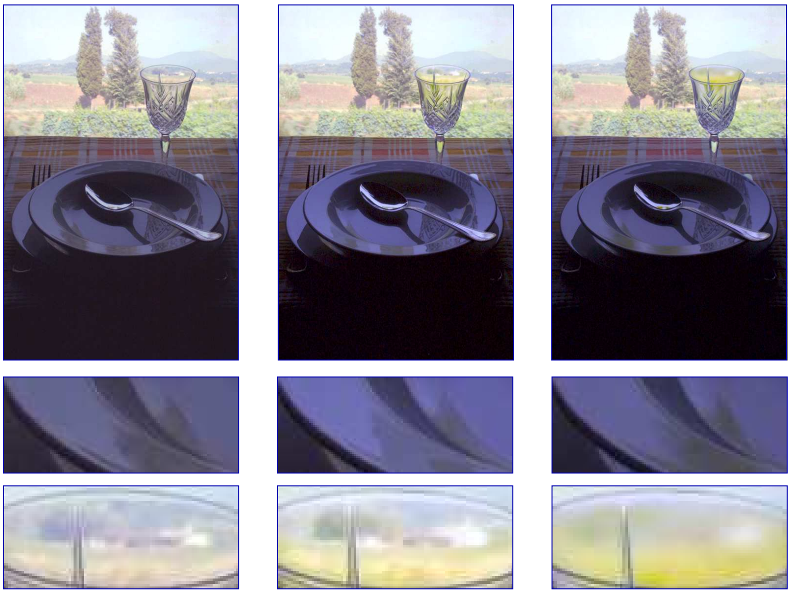


Figure 6.19: The effect of increasing the number of wavelet coefficients. The scene is captured using, from left to right, 100, 300, 600, and, 900 Haar wavelet illumination patterns. During relighting, directly visible pixels on the CRT monitor are not replaced, nor is the self-emittance term *S* added.

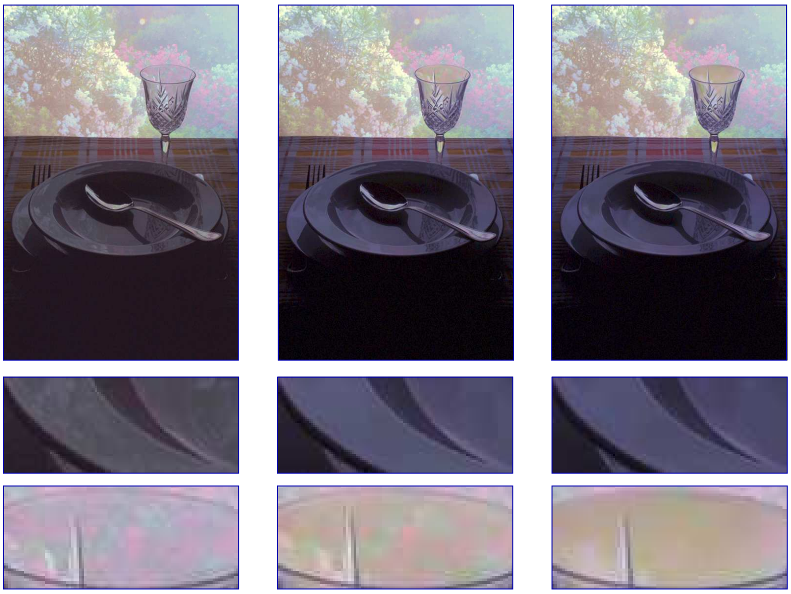


Reference Photograph

Relit Result (Haar Wavelet)

Relit Result (9/7 Daubechies Wavelet)

Figure 6.20: The effect of using different wavelet types during acquisition of a dinner scene. Left: reference photograph. Middle: Haar wavelet. Right: 9/7 Daubechies wavelet.



Reference Photograph

Relit Result (Haar Wavelet)

Relit Result (9/7 Daubechies Wavelet)

Figure 6.21: Identical to figure 6.20, but relit with different incident illumination.

requiring a large dynamic range to accurately capture both small and large footprint wavelet patterns.

6.7 Implicit Sampling

In this section, an implicit wavelet sampling algorithm for the acquisition of reflectance fields is introduced. This algorithm avoids the disadvantages of the explicit wavelet sampling algorithm of the previous section. Furthermore, it has a constant acquisition complexity, and a linear (in terms of desired number of terms) post-processing complexity. This method was first introduced in [75]. This section is structured as follows. First the basic idea is outlined in subsection 6.7.1. Next, we detail the idea of wavelet noise, and explain how it can be used in an image-based relighting context (subsection 6.7.2). In subsection 6.7.3, we show how an optimal progressive tree approximation can be inferred from the responses of the scene under wavelet noise illumination. Practical considerations are detailed in subsection 6.7.4. Finally, the results are discussed in subsection 6.7.5.

6.7.1 Basic Idea

From the previous section (6.6), it is clear that, although a quasi-optimal sampling for a single reflectance function can be defined, it is very unlikely that this sampling is optimal for all reflectance functions in a reflectance field. This implies, that an online sampling algorithm (i.e., using a feedback loop), will never achieve an optimal sampling for the complete reflectance field, since the acquisition process is inherently parallel with respect to all reflectance functions. Therefore, a different strategy is devised in this section. Instead of selectively sampling the reflectance field during acquisition, it would be better to capture as much information as possible from the reflectance field beforehand and afterwards reconstruct the reflectance field from this data.

Key to our technique is the use of wavelet noise illumination patterns. For each emitted wavelet noise pattern a high dynamic range photograph of the scene is recorded. Next, an optimal progressive tree approximation is inferred offline from the recorded data. By decoupling the sampling procedure from the acquisition, approximations for each pixel's reflectance function can be computed separately and more accurately.

A wavelet noise pattern is defined as the sum of weighted wavelet basis functions, and these weights are distributed according to a normal distribution. An example of wavelet noise is shown in figure 6.22. Using wavelet noise as illumination patterns has a number of advantages:

- It is possible to generate any number of unique wavelet noise patterns. Together with a progressive computation of the reflectance functions, this enables to trade off between the number of illumination patterns and the quality of the final approximation.

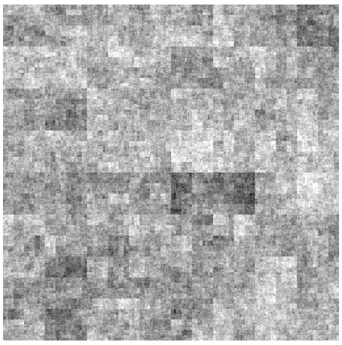


Figure 6.22: An example of Haar wavelet noise.

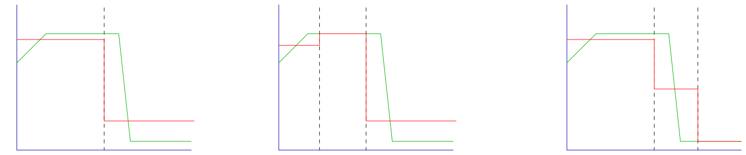
- Because the wavelet noise patterns are densely defined in both space and frequency, each observed pixel gives a response, independently of the underlying reflectance function, when illuminated by a wavelet noise pattern. Thus, each additional photograph contributes new information to the computation of a reflectance function for every pixel.

- It is possible to fix the average of each wavelet noise pattern, minimizing the dynamic range that needs to be captured.

A progressive algorithm is used to infer the reflectance function for each pixel separately from the responses of the wavelet noise patterns. The number of wavelet coefficients in the final approximation can be either user-defined or dynamically determined depending on the number of input illumination patterns. In order to progressively refine the approximation of the reflectance functions, a refinement oracle based on optimal progressive tree approximation with wavelets is used. A similar problem is addressed in the previous section (6.6), and in Matusik et al. [61]. Both use a progressive algorithm:

- In the previous section a feedback loop is used to determine online which subsequent wavelet illumination patterns are important for the approximation of the wavelet represented reflectance field. The selection of these subsequent illumination patterns is based on information from already acquired photographs. The children of the wavelet, that resulted in the largest contribution to the reflectance field (i.e., photograph with the largest weighted L_2-norm), are selected to be processed in the subsequent iteration. Note, that the selection of the wavelet illumination patterns is based on information gathered from all pixels in parallel, which can result in a non-optimal choice for individual pixels. Furthermore, it is possible that a child of another (less) important wavelet is more significant than the proposed wavelet illumination patterns.

- Matusik et al. [61] use a method in which reflectance functions are approximated by a collection of non-overlapping box filters, that are split progressively



(a) Coarse approximation. **(b)** Split according to energy. **(c)** Split according to entropy.

Figure 6.23: (a) An approximation (red) of a $1D$ reflectance function (green). (b) Refinement according to energy content. A better refinement criterion in terms of error, would be according to entropy (c).

> based on the current approximation for each pixel separately. The splitting criterion, however, is not optimal as demonstrated in figure 6.23. Sub-figure 6.23.a shows an approximation (red) of a $1D$ representation of a reflectance function (green). In sub-figure 6.23.b the approximation is refined by splitting the box filter with highest energy content (as is done in [61]). However, in sub-figure 6.23.c a more optimal refinement, according to entropy, is shown.

For this method we will use a different refinement oracle based on optimal progressive tree approximation that avoids the disadvantages of both methods.

6.7.2 Wavelet Noise

An important tool in this novel acquisition technique is wavelet noise. We define a wavelet noise pattern $\mathbf{M}_{\cdot,i}$ as the sum of normal distributed randomly weighted wavelet basis functions:

$$\mathbf{M}_{\cdot,i} = \psi\, \mathbf{W}_{\cdot,i},$$

where $\mathbf{W}_{\cdot,i}$ is a vector of l normal distributed weights.

The observed exitant illumination under incident illumination $\mathbf{M}_{\cdot,i}$ is (using equation (3.1)):

$$\begin{aligned} C &= \mathbf{T}\, \mathbf{M}_{\cdot,i} \\ &= \mathbf{T}_{\psi}\, \overline{\psi}^{T}\, \mathbf{M}_{\cdot,i} \\ &= \mathbf{T}_{\psi}\, \overline{\psi}^{T}\, \psi\, \mathbf{W}_{\cdot,i} \\ &= \mathbf{T}_{\psi}\, \mathbf{W}_{\cdot,i}, \end{aligned}$$

and thus the observed response c_p of a single pixel p is:

$$c_p^{(i)} = \mathbf{T}_{\psi_p} \cdot \mathbf{W}_{\cdot,i}.$$

Denote $\mathbf{W} = [\mathbf{W}_{\cdot,1}|\ldots|\mathbf{W}_{\cdot,n}]$, where each $\mathbf{W}_{\cdot,i}$ is a column in $\mathbf{W}$ (a set of wavelet noise patterns' weights). We would like to minimize the number of required illumination patterns, and thus the number of photographs that need to be captured, such that the reflectance function $\mathbf{T}_{\psi_p}$ can still be accurately determined.

Since the reflectance function $\mathbf{T}_{\psi_p}$ contains l elements, a brute force approach would require at least l wavelet noise patterns $\mathbf{W}_{\cdot,i}$, and requires a linear system to be solved (for the reflectance function $\mathbf{T}_{\psi_p}$) for each pixel p:

$$P = \mathbf{T}_{\psi_p} \mathbf{W}, \tag{6.8}$$

where P is a vector containing the observed (known) pixel values $c_p^{(i)}$ for each emitted wavelet noise pattern $\mathbf{W}_{\cdot,i}$. However, in chapter 5 we noted that each reflectance function $\mathbf{T}_{\psi_p}$ can be compactly represented by a non-linear wavelet approximation. Now suppose that each $\mathbf{T}_{\psi_p}$ can be sufficiently approximated by at most $m << l$ coefficients. If we know which m coefficients are significant, then an accurate estimate of the magnitude of these coefficients is possible from emitting at least m wavelet noise patterns[4] $\mathbf{M}_{\cdot,i}$. Denote the set of these m wavelet coefficients as $\mathtt{s}_m$, yielding a approximative wavelet basis $\psi^{\mathtt{s}_m}$, and a reflectance function approximation $\mathbf{T}_{\psi^{\mathtt{s}_m}p}$. Thus:

$$P \approx \mathbf{T}_{\psi^{\mathtt{s}_m}p} \mathbf{W}^{\mathtt{s}_m}, \tag{6.9}$$

where $\mathbf{W}^{\mathtt{s}_m}$ is a matrix that contains the random wavelet coefficients corresponding to the wavelet coefficients in $\mathbf{T}_{\psi^{\mathtt{s}_m}p}$.

If $\mathbf{W}^{\mathtt{s}_m}$, P, and the set of m significant coefficients are known, then the reflectance function $\mathbf{T}_{\psi^{\mathtt{s}_m}p}$ can be efficiently computed using a linear least squares minimization. Both $\mathbf{W}$ and P are known by either construction or acquisition. However, it is unknown which m coefficients are significant for a pixel p. Furthermore, each pixel p has a different set of m significant coefficients.

6.7.3 Progressive Refinement

To compute which m coefficients are significant for an approximation of the reflectance function, a progressive algorithm is used. This algorithm is based on optimal progressive tree approximation (section 6.5). However, before detailing the progressive algorithm we first show that it is possible to use a progressive algorithm to infer the reflectance functions. Next, a refinement criterion for progressively computing a reflectance function is derived.

[4] Generally, it is better to use more than m wavelet noise patterns, since it is possible that the m patterns are not linearly independent for the subspace spanned by the m non-zero wavelet coefficients.

Validity of a Progressive Algorithm. A progressive algorithm is only possible if a reliable estimate can be made of the magnitude of a subset of the significant wavelet coefficients. In other words, we need to ensure that the least squares error when estimating a subset (containing less than m, and less than l) of the wavelet coefficients is well behaved. Therefore, reconsider equation (6.9), and extend it with an error term $E_p^{(i)}$:

$$P = \mathbf{T}_{\psi^{\mathtt{s}_i}\, p}\, \mathbf{W}^{\mathtt{s}_i} + E_p^{(i)},$$

where $1 \leq i \leq m$, and $\mathtt{s}_i = \{\mathtt{s}_m[1], ..., \mathtt{s}_m[i]\}$. Combining with equation (6.8) we can write $E_p^{(i)}$ as:

$$\begin{aligned} E_p^{(i)} &= \left(\mathbf{T}_{\psi_p}\, \mathbf{W}\right) - \left(\mathbf{T}_{\psi^{\mathtt{s}_i}\, p}\, \mathbf{W}^{\mathtt{s}_i}\right) \\ &= R_p^{(i)}\, \mathbf{W}, \end{aligned}$$

where $R_p^{(i)}$ is the residue, containing the wavelet coefficients zeroed out in $\mathbf{T}_{\psi_p}$ to obtain $\mathbf{T}_{\psi^{\mathtt{s}_i}\, p}$. Note that $||R_p^{(i)}||_2$ decreases as i increases. Furthermore $||R_p^{(m)}||_2 \approx 0$.

An important observation is that the magnitudes of the elements in $E_p^{(i)}$ are distributed according to a normal distribution, since **W** follows a normal distribution by construction, and a weighted sum of normal distributions is a normal distribution itself. The mean of all elements in $E_p^{(i)}$ is zero. The variance is determined by the magnitude of the elements in the residue $R_p^{(i)}$.

Consequently, the i wavelet coefficients of $\mathbf{T}_{\psi^{\mathtt{s}_i}\, p}$ can be estimated from the observed pixel responses **P** using a linear least squares minimization (equation (6.9)). The error on the estimates decreases if the remaining most significant wavelet coefficients are added with increasing i. Therefore, the reflection function's wavelet approximation $\mathbf{T}_{\psi^{\mathtt{s}_i}\, p}$ should contain the i largest coefficients, in order to minimize the error on the estimates.

This proves the validity of a progressive algorithm, because it enables to make a reliable estimate of the wavelet coefficients of a subset of any size (less than m). Furthermore, this implies, that, by carefully extending the set $\mathtt{s}_i$, intermediate decisions can be made based on the current estimates of the wavelet coefficient $\mathbf{T}_{\psi^{\mathtt{s}_i}\, p}$. Finally, with each additional increase of the set $\mathtt{s}_i$, the quality of the estimates increases.

Progressive Refinement. The previous argument shows that it is indeed possible to use a progressive algorithm to extract an approximation of a reflectance function from the observed pixel values under wavelet noise illumination. The estimates of the i largest wavelet coefficients, can in turn be used to drive the progressive algorithm,

```
Data:  priorityQueue processOrder
       list treeApproximation

Init:  add the scale function to processOrder
       treeApproximation is empty

while(maximum number of coefficients not reached and
       processOrder is not empty and
       processOrder.top $>$ threshold)
{
    newWavelet $=$ processOrder.top
    add newWavelet to treeApproximation
    estimate the energy in each of the direct children which
         overlap the footprint of newWavelet (Least Squares)
    insert children in processOrder
}
Re-estimate all selected wavelet coefficients (Least Squares)
    in treeApproximation to arrive at the final solution.
```

Figure 6.24: A high level description of the algorithm used to compute a reflectance function approximation for each pixel.

based on an optimal progressive tree approximation.

An algorithm based on optimal progressive tree approximation can be used for each reflectance function separately as follows: The set $\mathtt{s}_i$, with $i = 1$, of important coefficients is initialized to the root of the wavelet tree. The corresponding wavelet coefficients of the reflectance function $\mathbf{T}_{\psi^{\mathtt{s}_i}\,p}$ can be estimated by a linear least squares optimization on equation (6.9). Using these estimates, the set $\mathtt{s}_i$ is expanded using the progressive refinement rule of equation (6.3), and the process is repeated until a sufficient accurate approximation of the reflectance function is obtained. This algorithm is similar to the one presented in the previous section (6.6), except that the HDR photographs are recorded beforehand, thus the cost of a query operation is much less expensive (i.e., only a linear least squares system needs to be solved, which can be computed relatively fast).

A problem with this approach is that each time the set $\mathtt{s}_i$ is extended, all wavelet coefficients in the current approximation and all their children's coefficients need to be re-estimated. Re-estimating all wavelet coefficients at each iteration requires a significant overhead, not only in terms of computation, but also in terms of required number of illumination patterns, since the number of children grows linearly with the length of the current approximation. A solution to this dilemma is to only estimate the children of the wavelet that has been added most recently, and keeping the estimates of the children of previously added wavelets without re-evaluating them. The rationale is that the relative error on large estimates is small, and thus the estimate will not change

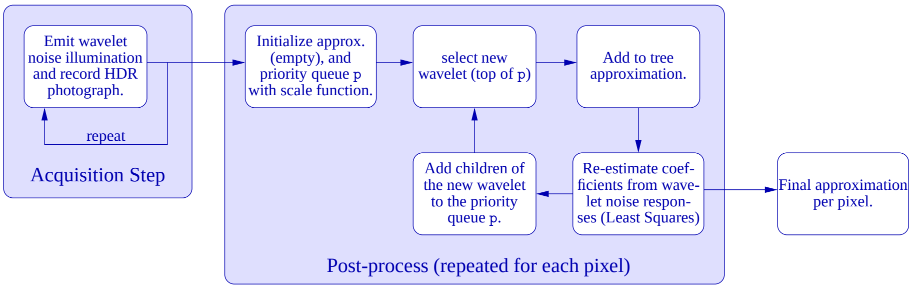

Emit wavelet noise illumination and record HDR photograph.
repeat
Acquisition Step
Initialize approx. (empty), and priority queue p with scale function.
select new wavelet (top of p)
Add to tree approximation.
Add children of the new wavelet to the priority queue p.
Re-estimate coefficients from wavelet noise responses (Least Squares)
Final approximation per pixel.
Post-process (repeated for each pixel)

Figure 6.25: A flow-chart of the implicit sampling algorithm (see also figure 6.24)

much in magnitude. The relative error on small estimates, however, can be large, but this is not really a problem since we are mainly interested in large coefficients. Once the final set $\mathtt{s}_m$ is found, all coefficients are re-estimated to maximize accuracy.

This can be efficiently implemented using a priority queue (similar to equation (6.4)), that contains the estimated children's wavelet coefficients from which a single coefficient is selected in each iteration step. The resulting algorithm is shown in figure 6.24. There are three wavelet orientations associated with each location in space. We store the total magnitude of all three wavelets as a single entry in the priority queue. When a specific wavelet location is retrieved from the queue, we estimate the direct children in the footprint of all three wavelets (i.e., 12 children for the Haar wavelet). Therefore, the priority queue is sorted according to intensity, and all three color channels are processed in parallel. The threshold in the algorithm in figure 6.24 ensures that we are not modeling camera noise. A flow-chart of this algorithm is given is figure 6.25.

6.7.4 Practical Considerations

The algorithm in figure 6.24 requires a least squares minimization for each iteration during the computation of a reflectance function. A least squares minimization requires $O(m^2e)$ operations, where m is the number of coefficients to be computed, and e is the number of emitted wavelet noise patterns, (e.g., using QR factorization), this is repeated m times. Thus the total operation count for the computation of a single reflectance function is $O(m^3e)$. This computation has to be repeated for each pixel's reflectance function (i.e., typically 1000×1000 times). However, this approach is not optimal, since a large number of operations is repeated between consecutive least squares minimizations. In figure 6.26 a sequential variant on the QR factorization algorithm is given for computing the reflectance function in $O(m^2e)$ time-complexity, instead of $O(m^3e)$. The algorithm is based on [16] (also in [97], p339), where a QR factorization is updated for the case a column is added to the original matrix. In our algorithm this corresponds to adding the wavelet noise coefficients associated with a specific wavelet "location" (i.e., $\mathbf{W}^{\mathtt{s}_i}$ is extended in equation (6.9)). The most expensive step in the algorithm is (6), the back-substitution. To minimize the computational

```
Input:   $\mathbf{Q}^{(i-1)}$, $\mathbf{R}^{(i-1)}_{\cdot,1:i-1}$, $B^{(i-1)}$, and $\mathbf{A}_{\cdot,i}$
         with $\mathbf{Q}^{(0)} = \mathbf{I}$, $\mathbf{R}^{(0)}_{\cdot,1:0}$ = empty, and, $B^{(0)} = B$
         $(\mathbf{Q}^{(i-1)}\mathbf{R}^{(i-1)}_{\cdot,1:i-1} = \mathbf{A}_{\cdot,1:i-1})$
Output:  The solution to $\mathbf{A}_{\cdot,1:i}\, X = B$

(1)  $V = {\mathbf{Q}^{(i-1)}}^T\ \mathbf{A}_{\cdot,i}$
(2)  compute the Householder reflection $\mathbf{H}^{(i)}$ using $V$
(3)  $\mathbf{Q}^{(i)} = \mathbf{Q}^{(i-1)}\ \mathbf{H}^{(i)}$
(4)  $\mathbf{R}^{(i)}_{\cdot,1:i} = \left[\, \mathbf{R}^{(i-1)}_{\cdot,1:i-1} \mid \mathbf{H}^{(i)}\ V \,\right]$
(5)  $B^{(i)} = \mathbf{H}^{(i)}\ B^{(i-1)}$
(6)  back-substitute $\mathbf{R}^{(i)}_{\cdot,1:i}\ X = B^{(i)}$ for $X$
```

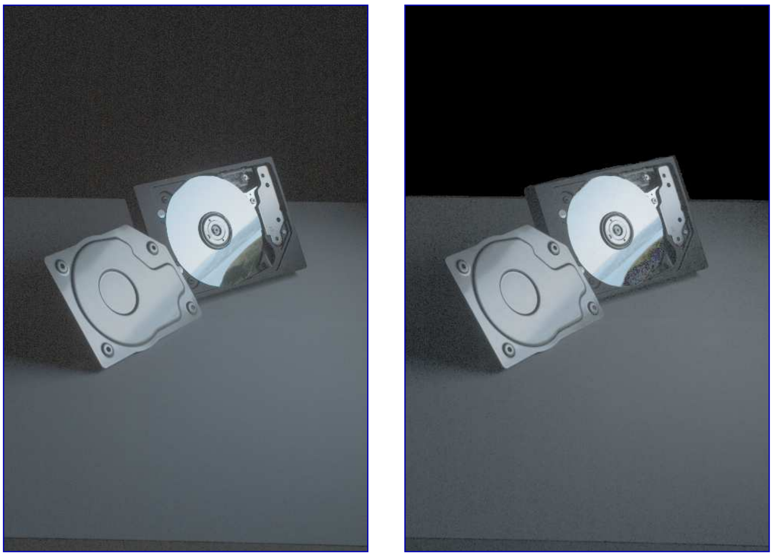
Figure 6.26: A sequential QR factorization algorithm for computing a least squares solution of $\mathbf{A}_{\cdot,1:i}\ X = B$ for each iteration of the algorithm in figure 6.24.

cost, the back-substitution should be postponed as long as possible, i.e., only back-substitute the elements that need to be estimated in the current iteration. Alternatively, incremental singular value decomposition [10] can be used instead of a sequential QR factorization.

This all fits in the progressive algorithm 6.24 as follows: When adding a new wavelet coefficient (from the priority queue) to the approximation, the corresponding vector of wavelet noise coefficient is added (this corresponds to $\mathbf{A}_{\cdot,i}$ in algorithm 6.24), and the $\mathbf{Q}$ and $\mathbf{R}$ factors are updated using the algorithm 6.26 (excluding the back-substitution step (6)). Next the wavelet children of this wavelet are determined, and added to the priority queue, Again, the corresponding vector of wavelet noise coefficients are (temporarily) added, and $\mathbf{Q}$ and $\mathbf{R}$ are (temporarily) updated according to algorithm 6.26. However, instead of computing the complete back-substitution (step (6)), we only compute a partial back-substitution for the coefficients of the child wavelets. Next, these coefficients are added to the priority queue, and $\mathbf{Q}$, $\mathbf{R}$, and, $\mathbf{A}$ are restored to their original state, and the whole process is repeated. Once the complete approximation is found, a complete back-substitution (step (6)) is computed to maximize accuracy.

6.7.5 Results and Discussion

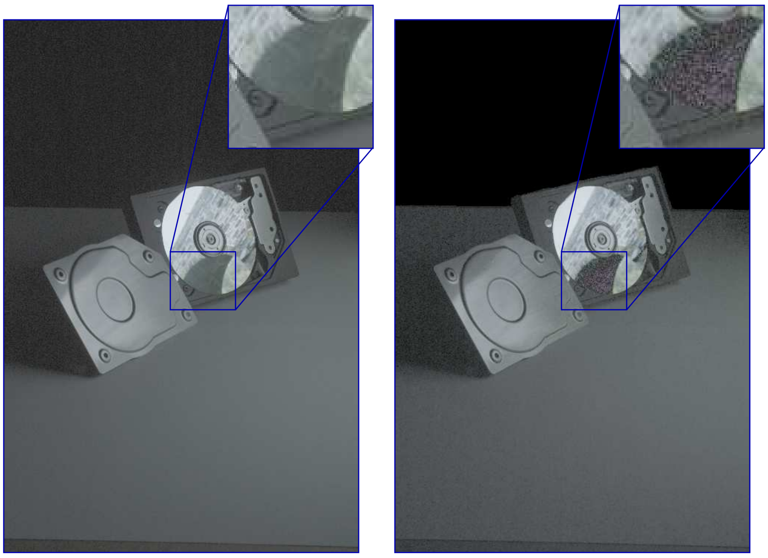
The result of the acquisition process is a set of reflectance functions $\mathbf{T}_{\psi^{\mathtt{s}^{(p)}_m}\, p}$, expressed in a wavelet basis ψ, and non-linearly compressed using a set $\mathtt{s}^{(p)}_m$. Note that this set $\mathtt{s}^{(p)}_m$ is different for each pixel p, and thus for each reflectance function. A relit image is created as follows: First, the novel incident illumination L is expressed in the dual wavelet basis $\overline{\psi}$. Then, a relit pixel value c_p is computed for every pixel p by taking



Reference Photograph

Relit Result

Figure 6.27: A hard disk illuminated from the right side by a photograph of a landscape. The scene contains specular (disk), glossy (cover), and diffuse (underground) materials. A reference photograph is shown on the left, and a computed relit image on the right. The reflectance field is computed from 256 wavelet noise patterns and each reflectance function is approximated by 64 Haar wavelet coefficients.



Reference Photograph

Relit Result

Figure 6.28: Identical to figure 6.27, but a photograph of an old stone bridge is used as incident illumination. A detail of the specular reflection showing the underside of the bridge is shown to illustrate the presence of stochastic noise on relit results.

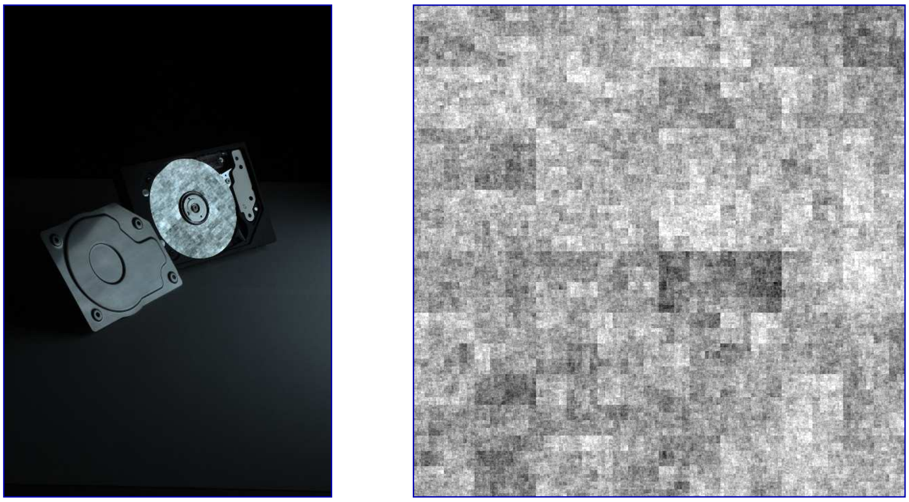


Figure 6.29: A photograph of a hard disk (left) under wavelet noise illumination (right).

the dot-product of the non-linearly approximated reflectance function $\mathbf{T}_{\psi^{\mathsf{s}_m^{(p)}}\, p}$ and the incident illumination $L_{\overline{\Psi}}$.

In figures 6.27, 6.28, 6.31, 6.32, and 6.33 some results of this technique are given. A reference photograph is shown, for each pair, on the left, while the computed relit image is depicted on the right. The reflectance functions for all examples, except those in figure 6.33, are approximated by 64 coefficients computed from 256 photographs. The reflectance functions in figure 6.33, are approximated using 128 coefficients computed from 512 photographs. The resolution of the incident illumination is 512×512. We would like to stress at this point, that each reflectance function uses a different set of 64 (or 128) Haar wavelet basis functions, as opposed to the technique of section 6.6 where all reflectance functions used the same set of wavelet basis functions.

In figure 6.27 and figure 6.28, a hard disk is shown, illuminated from the right side by two different photographs. The scene contains specular (disk), glossy (cover), and diffuse (underground) material properties. The final computed reflectance field is 68MB LZW compressed (158MB uncompressed). An example of this scene under a wavelet noise illumination pattern is shown in figure 6.29.

A detail of this scene, containing a balanced selection of different material properties, is used to test the influence of the number of coefficients versus the number of photographs. The results are depicted in figure 6.30. The red graph shows the error with respect to the reference image as a function the number of coefficients computed using 256 illumination patterns. A minimum is reached for 64 coefficients. Adding additional wavelet coefficients allows to approximate the reflectance function more

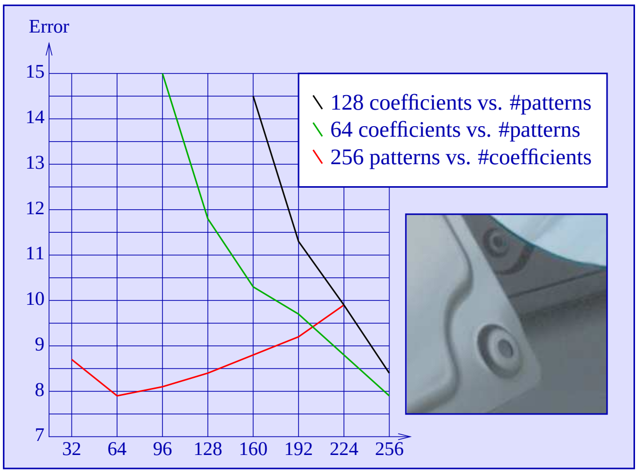


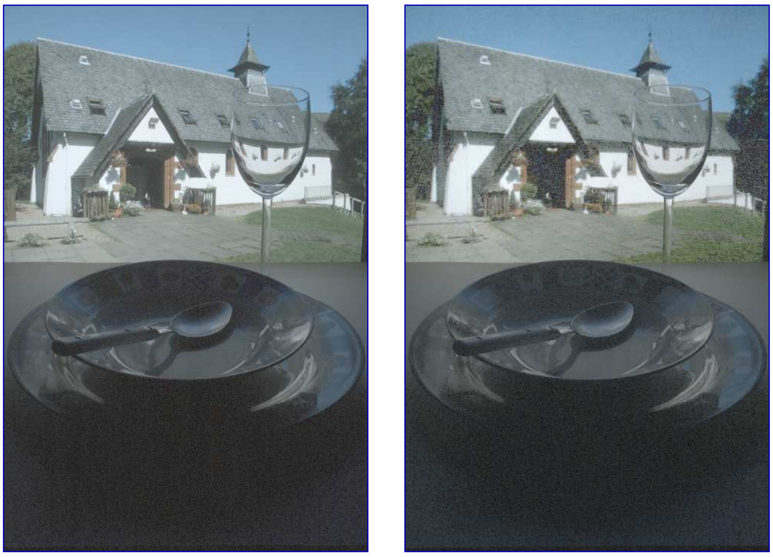
Figure 6.30: The error on the approximation and a reference photograph in terms of the number of coefficients and the number of photographs. The error is computed on a detail of the scene from figure 6.27, and contains a balanced selection of diffuse, glossy, and specular material properties. The red graph shows the error with respect to the number of coefficients when using 256 illumination patterns. The green and the black graph shows the error on respectively 64 and 128 coefficients for a variable number of illumination patterns.

accurately (e.g., from 32 to 64 coefficients), however, only a limited amount of information is available to estimate the coefficients, and thus, with each increase in the number of coefficients, the accuracy of the estimates decreases. At a certain point, the error on the estimates outweighs the addition in wavelet coefficients (e.g., from 64 to 96 coefficients). The green and black graphs show the error for respectively 64 and 128 coefficients as a function of the number of illumination patterns. As expected the error decreases with each additional illumination pattern.

The processing time required depends on the number of coefficients, the number of photographs, the underlying material properties, and the noise threshold. We tested a number of settings on the detail shown in figure 6.30. The timings for each setting ranged from 0.005 to 0.075 seconds per pixel on average, on a 3GHz Pentium 4 with 1GB of memory.

In figure 6.31 and figure 6.32 a scene is shown, similar to figure 6.20 and figure 6.21. Unlike the previous method (section 6.6), we also computed the reflectance functions of directly visible pixels on the CRT monitor. The final computed reflectance field is 139MB LZW compressed (329MB uncompressed).

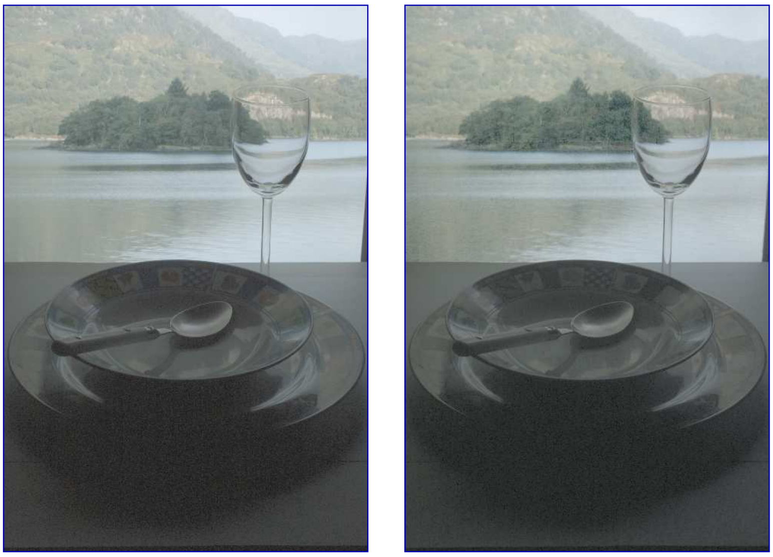
To compare the quality of the obtained results of this technique versus the technique of the previous section consider figure 6.33. This scene contains the glass candy bear jar which is also used in figures 6.16 and 6.17. The reflectance functions of this scene are approximated using 128 Haar wavelet coefficients, that are inferred from



Reference Photograph

Relit Result

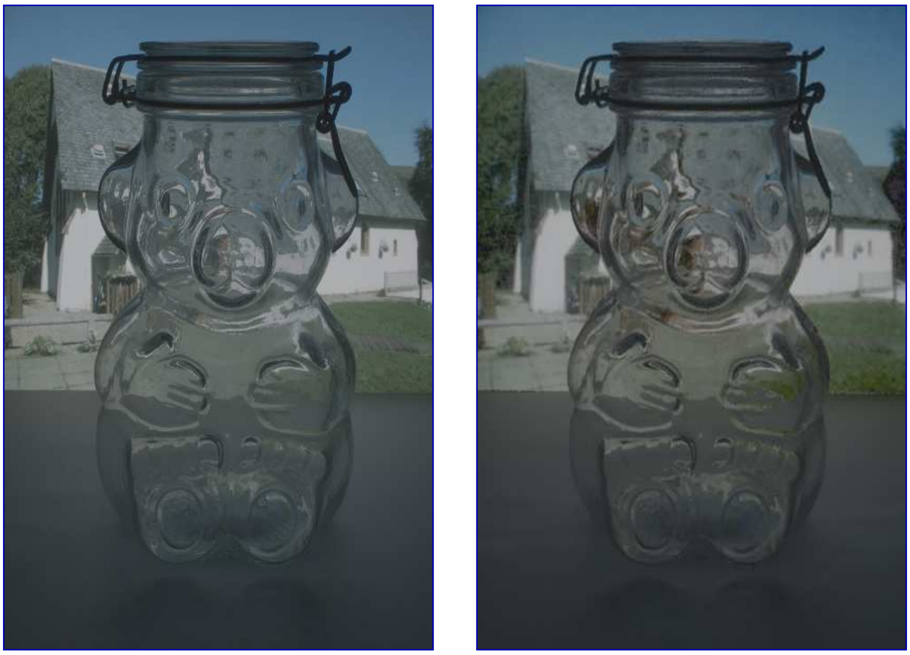
Figure 6.31: A dinner scene, similar to figures 6.20 and 6.21, captured using 256 wavelet noise patterns. Each reflectance function is approximated using 64 Haar wavelet coefficients. A photograph of a small church is used as incident illumination.



Reference Photograph

Relit Result

Figure 6.32: Identical to figure 6.31, but relit using a photograph of a small island as incident illumination.



Reference Photograph Relit Result

Figure 6.33: The glass candy bear scene, which also featured in figures 6.16 and 6.17, captured using 512 wavelet noise patterns. Each reflectance function is approximated using 128 Haar wavelet coefficients.

the effects on the scene of 512 wavelet noise patterns. Note that although the number of recorded HDR photographs is only 25% of the number used for the result in figures 6.16 and 6.17, the results are visually more pleasing.

All three results show some noise in dark areas. There are two sources of noise. First, there is measurement noise, noticeable in regions with low reflectance. Second, there is some stochastic noise, due to the stochastic nature of the illumination patterns and estimation process (e.g., visible in the detail of figure 6.28, under the arch of the bridge). This stochastic noise shows some structure related to the (Haar) wavelet used. The amount of noise should decrease if the number of photographs increases.

The methods of section 6.6 (also in Peers and Dutré [74]) and Matusik et al. [61] are closest related to the presented method. What follows is a short comparison of the presented methods and [61]. Both [61] and this method have a straightforward data acquisition process, which is complicated in [74] by the feedback loop. The number of distinct wavelets in the computed reflectance functions is approximately 262000, which is almost equal to the total number of possible wavelets (given the resolution of the incident illumination). This gives an idea on how many wavelets have to be emitted to achieve similar results with the technique of [74]. Matusik et al. [61]

solve a constrained linear least squares problem, which is more complex than solving an unconstrained linear least squares problem. Furthermore, to further enhance the results [61] requires a spatial correction, which can fail if the scene contains many high frequency features. The presented method performs better when compared to the results of [61] without spatial correction. We use a more optimal refinement criterion compared to both previous methods. Furthermore, it is unclear what the constraints are on the natural illumination used in [61], whereas the wavelet noise patterns are well defined. A key difference is that the wavelet noise patterns are dense in space and frequency, whereas this is not guaranteed for natural images.

6.8 Conclusion

In this chapter, two progressive tree approximation algorithms with wavelets have been detailed. These two algorithm form the basis for two new efficient acquisition methods for reflectance fields. Both methods use a CRT monitor to emit structured illumination upon the scene, and both use a progressive algorithm to approximate the reflectance field. The first method, based on greedy progressive tree approximation, uses a feedback loop and selects, during acquisition, which wavelet basis functions to use. The second algorithm, based on optimal progressive tree approximation, uses a fixed set of wavelet noise patterns. From the scene's response to the wavelet noise illumination, each reflectance function is progressively inferred.

7

Sampling $4D$ Reflectance Functions

This chapter discusses the acquisition of 4*D* reflectance functions. A novel acquisition device based on the Light Stage is presented. It is shown how this all fits in the framework developed in chapter 3. Finally, an acquisition acceleration method is detailed that reduces the acquisition complexity from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^3)$. The work presented in this chapter was first published in [58].

7.1 Introduction

In chapter 2, 2*D* approximations of 4*D* incident illumination were discussed. The motivation behind approximating the incident illumination by a lower dimensional representation is that due to the direct relation between the dimensionality of incident illumination and that of the reflectance functions, it is sufficient to capture a lower dimensional approximation of each reflectance function. Capturing 2*D* reflectance functions, and thus a 4*D* reflectance field, is less complex than acquiring the full 6*D* reflectance field.

The relit results under 2*D* incident illumination look realistic and convincing. However, there are situation in which 2*D* incident illumination does not suffice. An illustration of such a situation is shown in figure 7.1. In this example a paper and a toy soccer ball are shown, illuminated by the sun shining through some Venetian blinds. The light coming from the sun is highly directional, while the Venetian blinds cast a spatially varying illumination pattern on the scene. When capturing an environment map (2*D* incident illumination) of this scene will yield a different result depending on the location of the light probe, e.g., the blue and red marked spots in figure 7.1. For the blue marked spot, the sun will be visible in the environment map. For the red marked spot, this will not be the case.

It is obvious that certain illumination conditions require 4*D* incident illumination. However, it is not clear if this also implies the need for 4*D* reflectance functions. A common misconception is that if the geometry of an object is known, and for each surface point a 2*D* reflectance function is also known, then the effects of 4*D* reflectance functions can be exactly simulated. The idea is that at each surface point, a 2*D* slice of the 4*D* incident light field is extracted, and used in a 2*D* relighting computation.

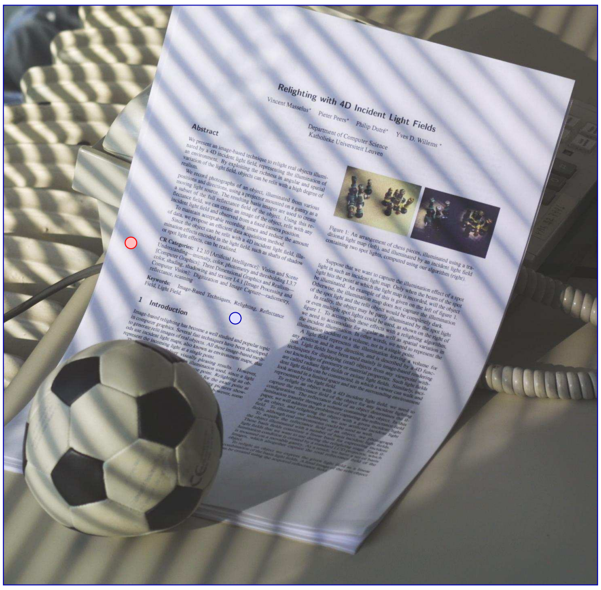


Figure 7.1: A photograph illustrating the importance of 4*D* incident illumination. A paper and a toy soccer ball lit by the sun shining through some Venetian blinds. The Venetian blinds cast spatially varying illumination onto the scene.

In figure 7.2, a situation is shown where this will fail. In this figure, an object is illuminated by parallel beams of light. A point (a), not illuminated by a beam, still receives indirect illumination from a point (b) (that is illuminated by a beam). In global illumination [29], such indirect illumination is computed by performing a full light transport simulation through the scene. The relighting computations in image-based relighting are similar to the direct illumination computations in global illumination, and hence, indirect illumination from other surface points is not included in such a case. A notable exception occurs when at each surface point, the incident 2*D* slices of the 4*D* incident light field are identical. This situation corresponds to relighting with

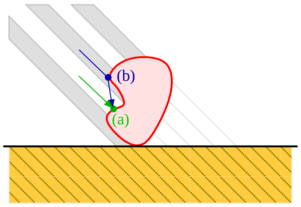


Figure 7.2: An illustration of the effects of 4*D* incident illumination. The 4*D* incident illumination consists of beams of light. Point (a) lies outside a beam, while point (b) is directly illuminated by a beam. The indirect illumination from point (b) to point (a) can only be accurately computed, when using both 4*D* incident illumination and 4*D* reflectance functions.

2*D* incident light fields, as discussed in the previous chapters.

To relight with 4*D* incident light fields, we need to capture a 6*D* reflectance field. Similar to capturing 4*D* incident reflectance fields, this implies that we need to know how these reflectance fields react to incident illumination, 4*D* in this case. Although this might seem trivial, a number of practical problems need to be solved. First, how do we sample this 4*D* space of incident illumination? Second, a 4*D* space also implies an $\mathcal{O}(n^4)$ acquisition complexity. How can we reduce this complexity such that the duration of the acquisition becomes more practical. The first question is answered in section 7.3. In section 7.5 an acquisition acceleration method is introduced. In section 7.2 the implications on our theoretical framework of using 4*D* incident illumination is investigated.

7.2 Relighting with 4*D* Incident Light Fields

In this section, we first study how relighting with 4*D* incident light fields can be expressed in terms of the framework developed in chapter 3. Next, the angular and spatial components of 4*D* incident light fields is introduced. Finally, an informal frequency analysis of the effects on the reflectance field of both the angular and spatial component of 4*D* incident illumination is made.

Framework. The framework presented in chapter 3 is flexible enough to support relighting with 4*D* incident light fields. The light transport is still a linear operation, even with 4*D* incident illumination. As a result the basis relighting equation (3.1) is still applicable:

$$C = \mathbf{T}L + S,$$

however, L now is a serialized version of the 4*D* incident illumination. This implies that a different serialization operator $\mathcal{S}_L$ has to be defined. As in the 2*D* case, a straightforward serialization of an (x,y,z,w) coordinate in a 4*D* space, discretized at a resolution of $n \times m \times l \times k$, can be serialized as $k(l(mx+y)+z)+w$.

As before, each reflectance function, $\mathbf{T}_i$ is a 4*D* function serialized identically as the incident illumination. Each element $\mathbf{t}_{i,j}$ in this reflectance function is the response of the i-th pixel under incident illumination $L^{(j)} = [\delta_{k,j}]_k$. In other words, by setting one of the elements of the incident illumination to one, the corresponding response of the reflectance functions can be observed.

Angular and spatial component. To gain further insight in the nature of 4*D* reflectance fields, consider the illustration of 4*D* incident illumination in figure 7.3. Intuitively, 4*D* incident light fields consist of two components: an *angular* component, and a *spatial* component. Note that the two proposed 2*D* approximations in section 2.2, omit one of these components. Furthermore, note that an environment

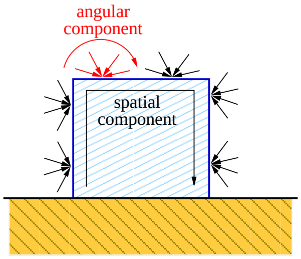


Figure 7.3: An illustration of the angular and spatial component of 4*D* incident light fields.

map is equivalent to the angular component for a specific spatial coordinate.

The Light Stage samples a 4*D* reflectance field (i.e, a collection of 2*D* reflectance functions), by emitting illumination from a point on a bounding sphere. This corresponds to sampling over the spatial component, and emitting illumination over the full angular range at each point. A CRT monitor, used in chapter 6, samples a 4*D* reflectance field densely in a similar manner. Moving from one pixel to another implies a change in spatial coordinate. Each pixel emits light diffusely, and thus over the full angular range.

Frequency analysis. To design an efficient acquisition apparatus for 4*D* reflectance fields, an intuitive frequency analysis of these 4*D* reflectance functions is presented.

The frequency response of spatially sampled 4*D* reflectance functions, is similar to the frequency response of 2*D* reflectance functions. As discussed in chapter 4, these are not band-limited. Furthermore, this component is very closely related to the BRDFs of the underlying materials.

To better understand the frequency response of angularly sampled 4*D* reflectance functions, consider the situation in figure 7.4. In this figure, an object is shown, illuminated angularly from a point on the bounding volume. From this illustration, it is

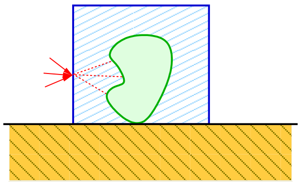


Figure 7.4: An illustration of the effect of angularly emitted incident illumination. Each angularly emitted ray (from a single point on the bounding volume) hits a different surface point on the object.

clear that the different angularly emitted rays hit the object at different surface points. Ignoring subsurface scattering and indirect illumination, this implies that each surface point is at most influenced from a single direction. And thus, the reflectance function is very high frequent. Subsurface scattering will in most cases yield a local response, and thus still result in a fairly high frequency response. Indirect illumination, is the only source of low frequency components.

7.3 Acquisition

As noted in the previous section (7.2), the frequency response of the reflectance functions under the spatial component of 4*D* incident illumination will be very similar to the frequency response of 2*D* reflectance function captured by a Light Stage. In this section, a novel acquisition device is presented that extends a "normal" Light Stage in order to acquire 6*D* reflectance fields.

Extending a Light Stage to capture 4*D* reflectance functions, requires to add the ability to emit illumination angularly at each Light Stage light source position. One could envision a similar sampling strategy (as with the spacial component) to achieve this goal, for example by aiming a light source at different directions. However, previously we noted that the corresponding responses of angularly emitted light have a high frequency behavior. Undersampling this component will yield visually disturbing relit images. In chapter 4 a number of strategies to combat these aliasing effects are discussed. The solutions included: *higher sampling rates*, *pre-filtering of the signal*, and additionally for image-based relighting applications: *pre-filtering the incident illumination*.

Setup description. Although preferred, using a higher sampling rate is not always possible due to mechanical limitations. Obviously, pre-filtering the signal is not possible. That leaves pre-filtering the incident illumination as only alternative. By replacing the light sources by an LCD or DLP projector, this can be easily achieved. A projector allows to emit angularly structured patterns. A photograph of a prototype setup can be seen in figure 7.5. This is a similar Light Stage configuration as the device presented in section 2.3, except that a single light source is moved by a gantry, instead of switching on and off different light sources. Emitting pre-filtered angular incident illumination is straightforward. A large selection of filters are available to pre-filter the incident illumination. In our implementation we use a box filter with a width equal to the sampling frequency. In other words, the projection plane is subdivided in $n \times n$ squares, and each square corresponds to a box-filtered sample. Each square is lit sequentially, and an HDR photograph is recorded. Other, more complex filters can also be used (e.g., Gaussian filters). We refer the interested reader to [56], chapter 6 for an in depth study of different filters to capture 4*D* reflectance functions. Note that only angular filtering is possible using this setup.

Such an HDR photograph corresponds to a 2*D* slice of the 6*D* reflectance field,

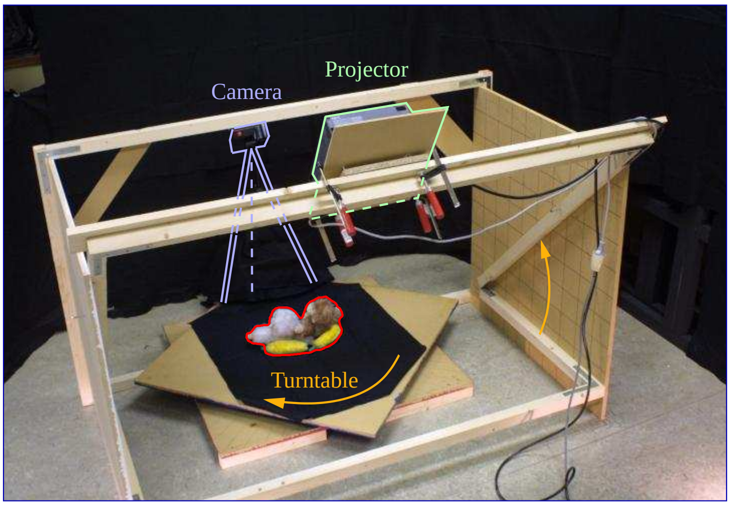


Figure 7.5: A photograph of a prototype acquisition device to capture 4*D* reflectance functions of an object. The object (marked in red) is mounted on a turntable. A camera (marked in blue) is also located on this turntable. A projector (marked in green) mounted on an arm can be moved along the latitudinal direction. Rotating the turntable realizes a (relative) longitudinal rotation of the projector with respect to the scene (and camera).

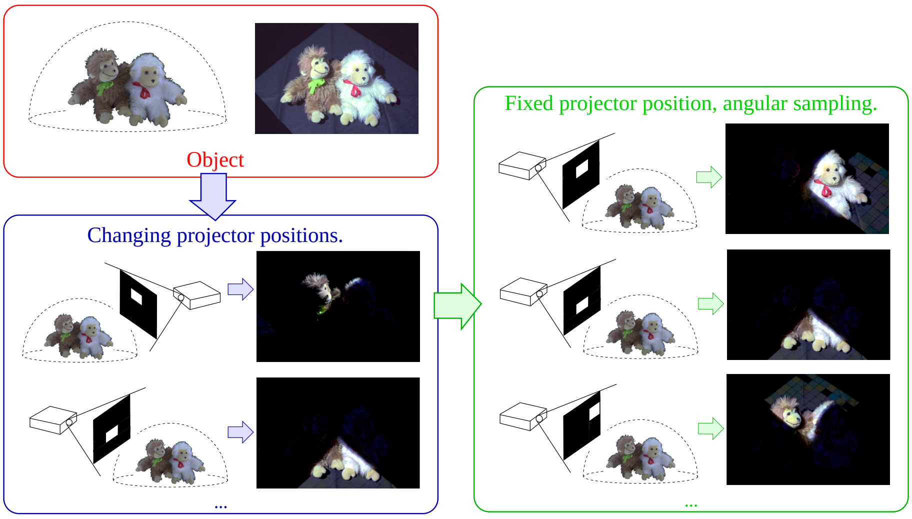

Object
Changing projector positions.
...
Fixed projector position, angular sampling.
...

Figure 7.6: An illustration of the data acquisition process. For each projector position (**blue**), the angular component is sampled using the projector (**green**).

and thus a column $\mathbf{T}_{\cdot,i}$ of the transport matrix $\mathbf{T}$. By illuminating each square for each projector position, and recording an HDR photograph for each illumination condition, the matrix $\mathbf{T}$ can be completely constructed. The acquisition process is iconically illustrated in figure 7.6.

Practical considerations. Before capturing reflectance fields using this setup, a few calibrations are required:

- **White balancing.** White balancing is performed by relating the white emitted from the projector to the observed white by the camera. Since it is not possible to directly observe the emitted white from the projector, a GretagMcBeth ColorChecker DC is used. The central patch in this color checker (in figure 3.2 such a color checker chart is shown) is diffuse white, and should, in theory, not alter the color of the reflected light. By aiming the projector, while emitting a solid white pattern, on this white patch, an indirect observation of the projector white is made.
- **Radiometric calibration of the projector.** In case complex, non-binary, filters are used to pre-filter the incident illumination, a radiometric calibration of the projector is required. This is similar to compensating for a CRT monitor's gamma curve (section 6.3), except that, as with white balancing, an indirect observation is made of the emitted illumination.
- **Projector black-level.** A typical LCD projector has a 1 : 300 contrast ratio, and a DLP projector a 1 : 2000 contrast ratio. This contrast ratio is the ratio between the darkest and brightest intensity. This also implies that the darkest intensity (i.e., emitting a completely black pattern) still results in an observed intensity. This has to be corrected in the recorded images. Therefore, for every projector position, an additional photograph is recorded while emitting a completely black pattern. This image is subtracted from each recorded photograph during acquisition.

7.4 Results and Discussion

To illustrate the effect of relighting with $4D$ incident light fields, a scene containing two toy monkeys is used (figure 7.7). This scene is captured from 224 projector positions, 32×7 regularly spaced locations on a hemisphere surrounding the scene. For each projector position, a 16×16 angular sampling is used. This yields a total of 57344 HDR photographs that need to be recorded. In figure 7.7, three different situations are simulated with respect to two light sources: the monkeys are centered between both lights, the monkeys are located before the lights, and, the monkeys are moved closer to the left light source. As can be seen, the effect on the relit scene is significant.

To further illustrate the effect of $4D$ incident illumination, comparisons with $2D$ incident illumination (i.e., environment maps) are made in figures 7.8 and 7.9. Again,

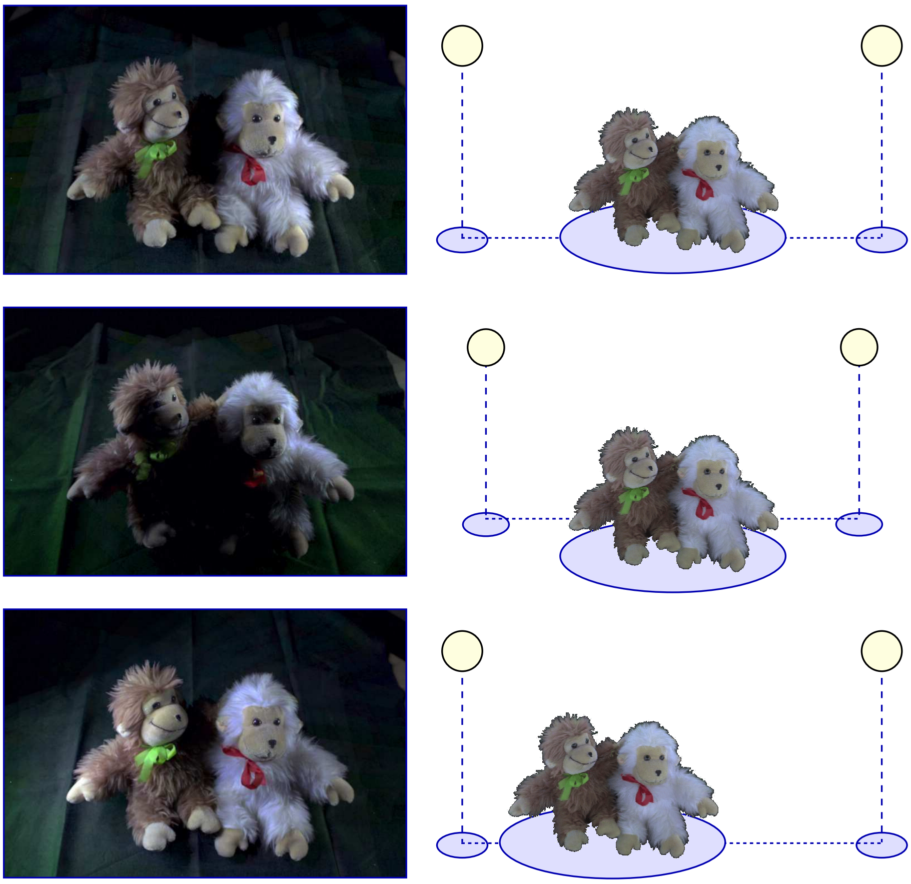


Figure 7.7: An illustration of 4*D* illumination effects on a scene containing two toy monkeys. Top: the monkeys are placed centrally between two light sources. Middle: the monkeys are moved forward with respect to the light sources. Bottom: the monkeys are shifted towards the left light source.

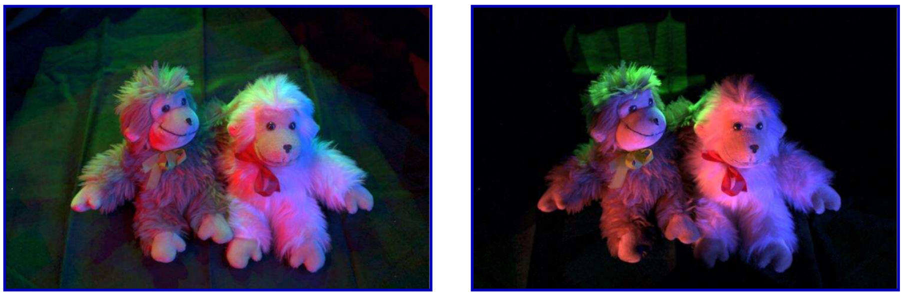


Figure 7.8: A comparison between relighting with 2*D* environment maps and 4*D* incident light fields. The toy monkey scene is illuminated by three colored beams of light aimed at the center of the scene. Left: relit result using 2*D* environment maps. Right: relit result using 4*D* incident light fields.

the scene containing two toy monkeys is used. In figure 7.8, three colored beams of light are aimed at the center of the scene from different locations. The width of these beams is smaller than the size of the scene. An environment map is constructed at the center of the scene. The left relit image in figure 7.8 is computed using this $2D$ incident illumination. The right relit image is computed using $4D$ incident illumination.

Unger et al. [104] presented two devices to capture $4D$ incident light fields. In figure 7.9 such an acquired $4D$ incident light field (courtesy of Unger et al. [104]) is used to relight the toy monkey scene. For comparison, a relit image of the same scene is computed under $2D$ incident illumination. This $2D$ incident illumination is a selected environment map from the $4D$ data set. The results in figure 7.9 illustrate that the acquired reflectance fields are usable when using real-world $4D$ incident light fields.

In figure 7.10 another scene is relit using $4D$ incident light fields. This scene contains an arrangement of different chess pieces, and is captured using 56 projector locations spread over a quarter of the hemisphere (8×7 subdivision). For each projector location, a 32×32 angular sampling is made. The total number of recorded HDR photographs for this scene equals 57344.

Using pre-filtered incident illumination during acquisition eliminates aliasing effects in the angularly sampled component of $4D$ incident illumination. However, since this response is high frequent, the used filter will be visible in the relit results, as can be seen in figure 7.11. In this figure, two details of selected areas are shown. The box filter used during acquisition is particularly visible. Using smoother filters will yield less visible discontinuities. Another, preferred solution is to sample this angular component at a higher resolution. However, this also increases duration of the acquisition.

7.5 Acquisition Speed-up

Capturing the $6D$ reflectance field of a scene or object is an enormous task. A large amount of HDR photographs have to be recorded. If we assume (optimistically) that a single HDR photograph takes 1 second, then for the scenes in the previous section, 16 hours of uninterrupted recording is required. Increasing angular incident resolution, worsens this situation even more. In this section, we propose a practical method to reduce the number of required HDR photographs. For simplicity, we assume that each dimension is discretized in n sample locations. The acquisition method described in section 7.3 has an acquisition complexity of $\mathcal{O}(n^4)$. The presented method is able to reduce this to an acquisition complexity of $\mathcal{O}(n^3)$ without significant loss of accuracy.

Recall the observation that sampling the angular component of incident illumination, yields reflectance functions dominated by high frequency components. Furthermore, the footprint of this response is compact (i.e., a localized influence). This potentially allows to emit more than one stimulus simultaneously. The response of

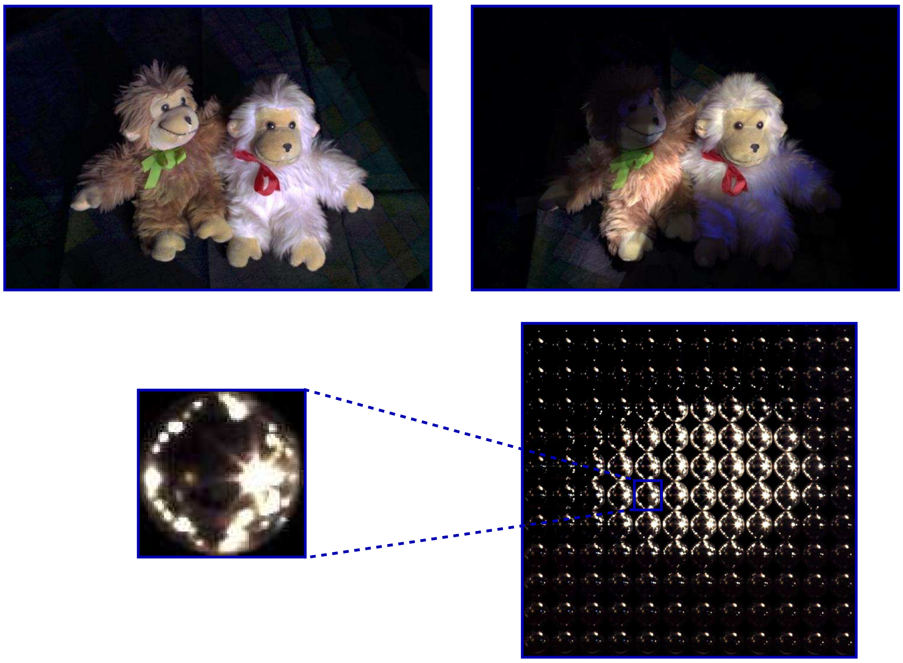


Figure 7.9: A comparison of relighting with 2*D* environment maps and 4*D* incident light fields. In this case, real-world acquired illumination is used. The 4*D* incident light field is captured and provided by Unger et al. [104]. Left: result of relighting with a 2*D* environment map. Right: result of relighting with a 4*D* incident light field. The corresponding incident illumination is shown below each relit result.

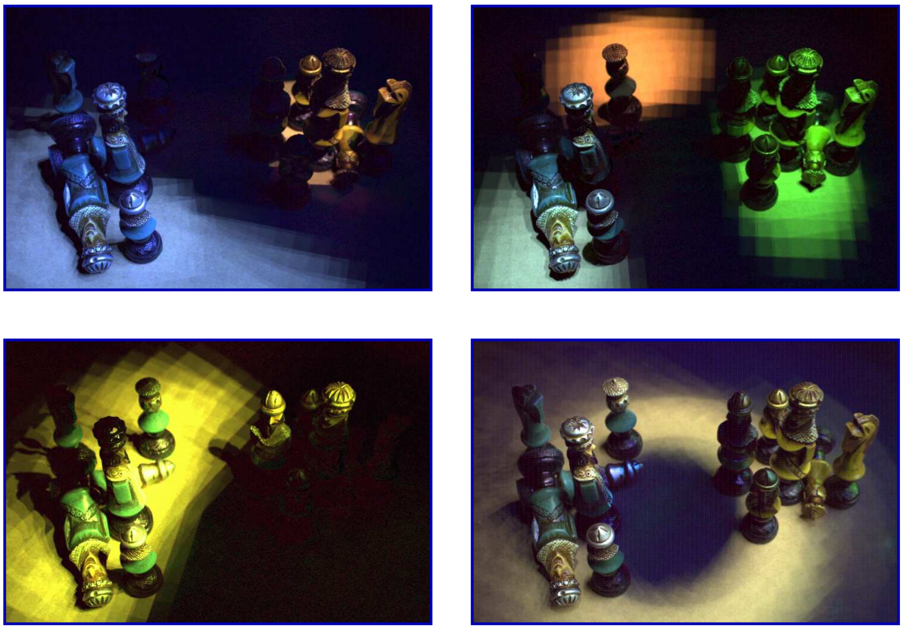


Figure 7.10: Various results of relighting a scene with 4*D* incident illumination containing an arrangement of chess pieces.

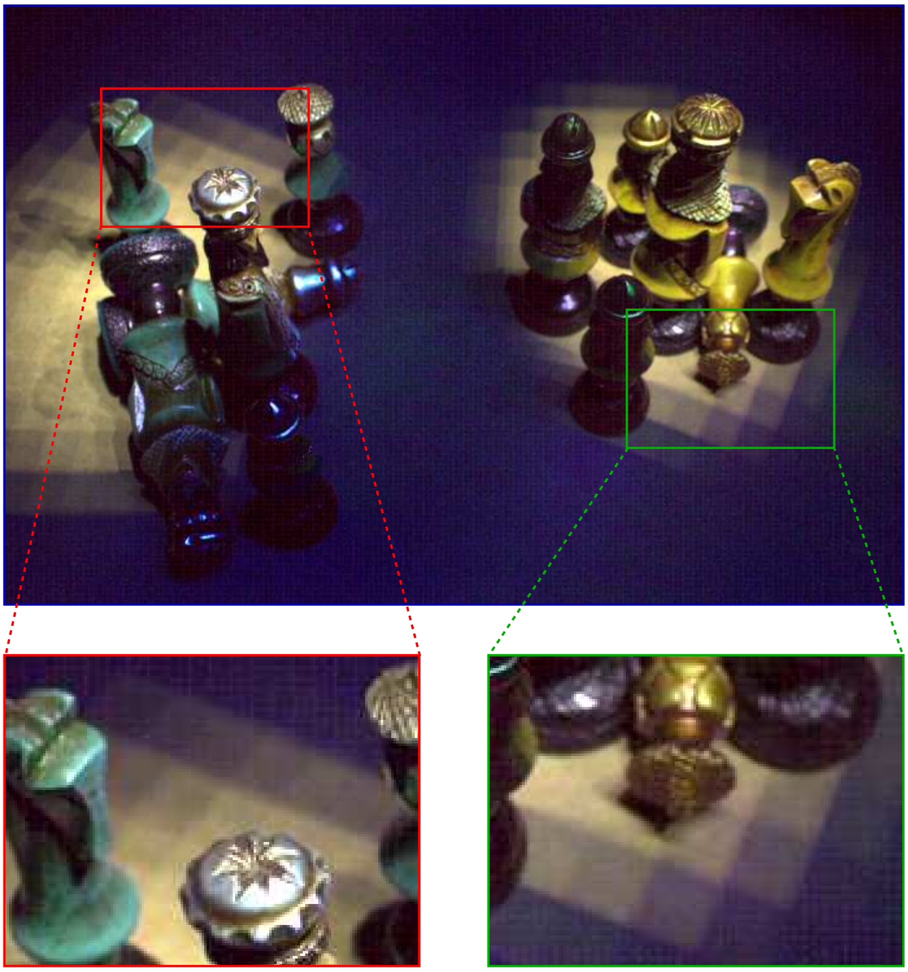


Figure 7.11: Illustrating the effect of using a low angular incident illumination discretization.

each individual stimulus can be reconstructed afterwards.

For this purpose, we create a horizontal set of patterns H as shown on the left of figure 7.12. Each H is constructed in such a way that each individual filtered stimulus does not influence another filtered stimulus in the same horizontal pattern H, once projected. A similar set, the vertical set of patterns V is created as well, as depicted on the right of figure 7.12. The response of the original (single) filtered stimulus $L^{(x,y)}$ occurs twice: once in a horizontal set $H^{(x,o_x)}$ and once in a vertical set $V^{(y,o_y)}$. Each pattern $H^{(x,o_x)}$ can be written as the sum of individual stimuli $L^{(i,j)}$:

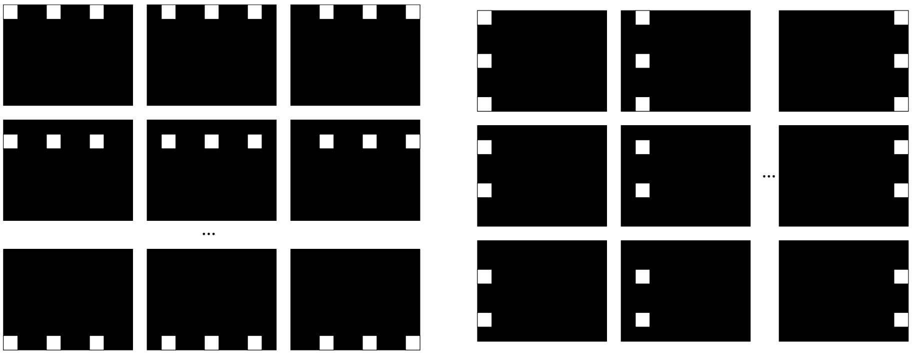


Figure 7.12: On the left, the set of horizontal patterns is shown, on the right, the set of vertical patterns. Each set of patterns is split up into 3 sets, in such a way that the patterns are spaced out evenly (horizontally and vertically respectively).

$$H^{(x,o_x)} = \sum_i L^{(x, i\times d+o_x)},$$

where d is called the *influence* parameter and determines what the distance between individual stimuli in the pattern is, and $o_x < d$. A similar definition exists for $V^{(y,o_y)}$. By construction we know that:

$$\min\left(H^{(x,o_x)}, V^{(y,o_y)}\right) = L^{(x,y)}.$$

The question arises under what conditions/assumptions can we extend this rule to the observed responses of each of the patterns:

$$\min\left(\mathbf{T}H^{(x,o_x)}, \mathbf{T}V^{(y,o_y)}\right) \approx \mathbf{T}L^{(x,y)}. \tag{7.1}$$

If the following conditions hold, then equation (7.1) is exact:

$$\begin{aligned}\min\left(\mathbf{T}L^{(x,i)}, \mathbf{T}L^{(j,y)}\right) &= \mathbf{T}L^{(x,y)} \quad if\,(i=y)\,and\,(j=x),\\ &= 0 \qquad otherwise,\end{aligned} \tag{7.2}$$

where:

$$\begin{aligned} i &\in \{0\times d+o_x, 1\times d+o_x, 2\times d+o_x, \ldots\},\\ j &\in \{0\times d+o_y, 1\times d+o_y, 2\times d+o_y, \ldots\}.\end{aligned}$$

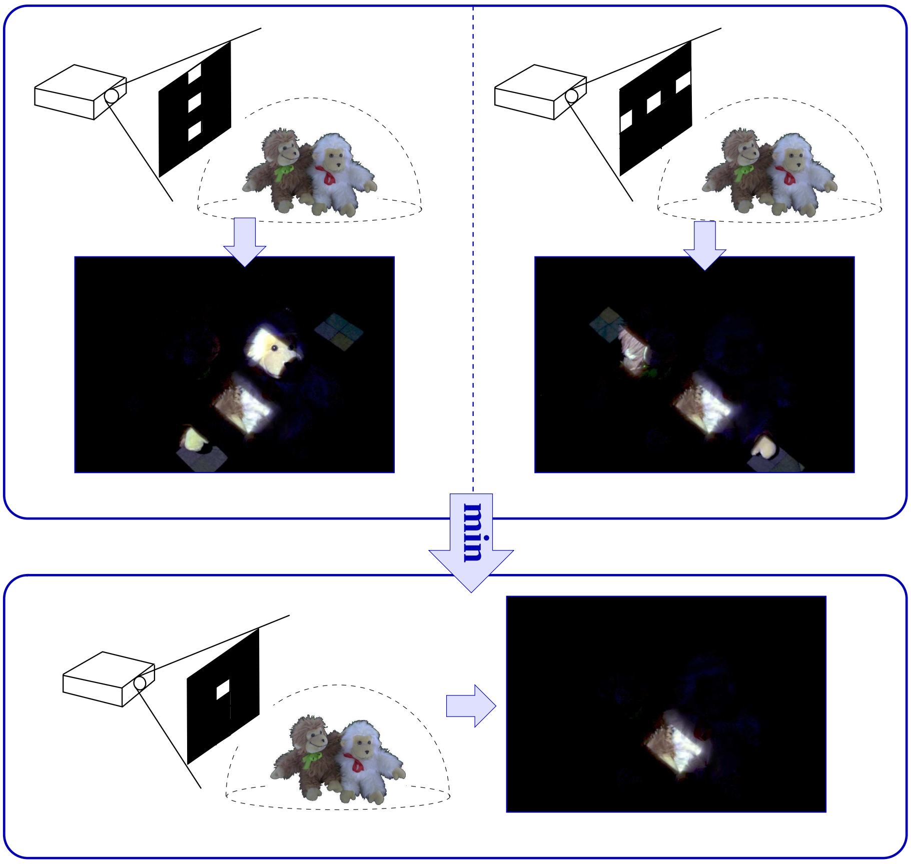


Figure 7.13: An illustration of extracting a single stimulus from the response of a horizontal and vertical pattern. By taking a pixel-wise minimum of the responses of the horizontal pattern and the vertical pattern, a good approximation of a single stimulus is obtained.

Informally: the responses of the other stimuli in each horizontal and vertical pattern do not influence each other, except for the target stimulus.

Thus, the image of projecting an individual square $L^{(x,y)}$ can be reconstructed by taking for each pixel the minimum of the corresponding pixel values in the images resulting from projecting $H^{(x,o_x)}$ and $V^{(y,o_y)}$, where $o_x = (y \mod d)$ and $o_y = (x \mod d)$. This reconstructed image is an approximation of the photograph resulting from emitting $L^{(x,y)}$. In figure 7.13, an illustration of this is given.

If the conditions (7.2) are not met, this acceleration technique will fail. For example, if the scene contains a diffuse concave bowl pointed towards the camera, there will be a lot of inter-reflections in the bowl. Using the minimum on pixel values will in this case not result in a correct approximation of a single stimulus due to this indirect light. In such a case, the acquisition can only be done by projecting each stimulus $L^{(x,y)}$ independently. By fine-tuning the influence parameter d, the amount of allowed inter-reflection can be selected. This acceleration method was also used to capture the reflectance fields of the results shown in the previous section. The patterns for each row or column were split into 3 separate sets (i.e., $d = 3$). This reduced the number of required photographs from 16×16 to $(2 \times 3) \times 16$ for the toy monkey example and from 32×32 to $(2 \times 3) \times 32$ for the chess pieces. The efficiency of the acceleration method improves with increasing resolution. In general we can reduce the complexity of our data acquisition from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^3)$, with n the discretization in each dimension.

7.6 Conclusion

In this chapter an apparatus for the acquisition of $6D$ reflectance fields of real objects, i.e., from a fixed vantage point has been presented. Moreover, an acceleration method has been presented to reduce the acquisition complexity from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^3)$ without significant loss of accuracy.

8

Conclusion

In this final chapter, a summary of this dissertation, and the original contributions of this work are given. Finally, some directions for future research are detailed.

8.1 Summary

This dissertation contributes to the sub-domain of *image-based relighting* in computer graphics. The general goal of image-based relighting is to visualize a scene or an object under novel illumination. The complexity of the illumination can range from a single light source to the observed illumination at a real-world location. More specifically, emphasis is placed on the sampling (acquisition) of reflectance fields, and reflectance functions of objects.

In the first part of this thesis, we explored the principles and constraints of image-based relighting in order to derive a solid mathematical formulation. The driving principles and constraints are: linearity of light transport, restriction to an idealized RGB color space, time invariance of the scene/object, and, restriction to a single vantage point. It turns out that, given these constraints, image-based relighting can be formulated as a linear system of equations, where each pixel is determined by the dot product of its reflectance function and the illumination.

Probably the best known method for acquiring reflectance functions is by using a Light Stage. A Light Stage enables fast and easy acquisition by sampling the reflectance functions from a large set of directions, omitting positional dependence. In order to use the sampled reflectance functions, a reconstruction must be performed. In this work several reconstruction methods are discussed and compared. Due to the properties of the linear system, we show that an appropriate downsampling of the incident illumination has a similar effect as reconstructing (or upsampling) the sampled reflectance functions.

Since image-based relighting is a linear system, any linear operation preserves the integrity of this system. A particularly interesting linear operation is changing the (mathematical) bases, i.e., expressing the incident illumination and the reflectance functions in a different domain (e.g., the frequency domain or the wavelet domain). A change of basis allows to express the reflectance functions in a different domain that has some desired property. In this work we consider the wavelet and frequency domain in order to compactly store the reflectance functions and speed-up the computations of a relit image using linear and non-linear approximations.

A special extension to non-linear wavelet approximations are tree approximations. In this work, we presented a progressive extension to tree approximations that allowed us to develop two new acquisition methods which use a CRT monitor as acquisition device instead of a Light Stage. The advantage of these new acquisition methods is that they are able to sample reflectance fields directly in a different basis. The first method, samples the wavelet represented reflectance field explicitly and selectively during acquisition. The approximation of the reflectance field is refined during acquisition, by selectively sampling parts of the wavelet domain that are considered important for the reconstruction of the reflectance field. The second approach further improves on this idea, by implicitly sampling the wavelet domain during acquisition. Acquisition consists of taking a fixed number of random slices through the wavelet domain. Such a random slice is called wavelet noise. Afterwards, during post-processing, a similar progressive sampling is done to reconstruct a reflectance function per pixel. The difference is that the response of a single wavelet sample is reconstructed from the data obtained from the acquired responses under wavelet noise.

All the methods discussed above are limited to capturing a 4*D* approximation of the reflectance field. In the final chapter of this work, an apparatus, based on the Light Stage, is presented for the acquisition of 6*D* reflectance fields. A brute force acquisition of such a 6*D* reflectance field, however, is cumbersome due to the huge amount of data which needs to be gathered. Therefore, an additional acquisition speed-up technique is presented.

8.2 Original Contributions

In this section we review the original contributions of this work organized per chapter.

Chapter 3. In this chapter, a concise notation based on matrices and linear systems is introduced. Serialization and resampling operators help in hiding the dimensionality of the problem and enable a unified theory for relighting with 2*D* and 4*D* incident illumination.

Chapter 4. This chapter explored signal processing techniques to improve the accuracy of the relighting computations.

- Different reconstruction techniques are investigated. All techniques were qualitatively and quantitatively compared. Linear interpolation and multi-level B-Splines interpolation performed best in terms of relative error. The latter reconstruction technique yielded a perceptually slightly better result in the case of animated incident illumination. This work has been done in collaboration with Vincent Masselus.

- We show that for a large class of reconstruction methods (i.e., upsampling methods), an equivalent downsampling method on the incident illumination can be defined resulting in identical relit results. Downsampling (the incident illumination) has the advantage over upsampling (the reflectance functions) that this operation has to be performed only once, and that individual relit pixel computations require less processing time.

Chapter 5. In this chapter a special case of lossy compression methods are investigated. These lossy compression techniques are based on basis projections, a linear operation. Non-linear wavelet approximation proved to be very effective in reducing the storage requirements and relighting computations.

Chapter 6. The previous chapter showed that representing reflectance functions in the wavelet domain yields a sparse representation. This sparseness was subsequently exploited to achieve compression. In this chapter we further exploit this property to develop faster, sub-linear, acquisition methods.

- Two progressive variants of tree approximation with wavelets are developed. Both methods, greedy progressive tree approximation, and, optimal progressive tree approximation, can yield a very compact representation of a function, without knowing the complete shape beforehand.

- An explicit sampling technique is developed that samples the reflectance field selectively and directly in the wavelet domain. This method is based on greedy progressive tree approximation. A feedback loop is used to decide what part of the wavelet domain is important to sample next in order to have an accurate approximation of the whole reflectance field.

- An implicit sampling technique is introduced that uses a fixed number of wavelet noise illumination patterns. During post-processing, an approximation of the reflectance function for each pixel is computed separately. This algorithm is based on optimal progressive tree approximation and is able to achieve high quality approximations using only a moderate number of illumination conditions.

Chapter 7. In this final chapter, a novel acquisition device, based on the Light Stage, is developed to capture and relight real-world objects with 4*D* incident illumination fields. Additionally, an acquisition speed-up technique is discussed that significantly reduces the acquisition complexity. This work has been done in collaboration with Vincent Masselus.

8.3 Directions for Future Research

Anyone who is looking for new directions for future research in image-based relighting should not despair after reading this dissertation. Let me assure you, there are still a lot of challenges ahead, albeit a bit less than a few years ago (see also chapters 2 to 7). The grand goal, the complete digitalization of real-world objects, is still far away. It is unlikely that this ultimate goal will be reached in the near future. A number of intermediate challenges still lie ahead:

4D incident illumination. In chapter 7 a method for relighting with 4*D* incident light fields is presented. This technique suffers from a number of problems. First of all, since it is based on the Light Stage, it also inherits its problems: limited sampling resolution, point samples, ... Second, because the apparatus uses a projector, a bulky construction is required, greatly taxing the design of a fully automated construction. Third. the resolution of the projected patterns is very limited, resulting in obvious aliasing artefacts.

A promising avenue for future research would be to improve upon the acquisition device presented in this dissertation, or even design a completely new one. The focus of this design can include the capture of performances, higher acquisition resolutions, and, lowering the number of required photographs (i.e., acquisition speed-up).

4D exitant illumination. In this work a scene is captured from only a single vantage point (subsection 3.2.4). The main motivation behind this restriction is that the dimensionality of the problem is reduced by two (i.e., an 8*D* reflectance field is reduced to a 6*D* field). This facilitates acquisition and data processing.

A few researchers have investigated image-based relighting with variable viewpoints [45, 62, 63], or with a restricted degree of freedom in camera movement [36]. None of these methods, however, are able to relight with 4*D* incident light fields. Furthermore, all of these methods use a brute force approach to capture the reflectance field and rely heavily on some form of geometrical proxy.

A true image-based approach, for shape as well as appearance, has yet to be presented. The advantages of image-based methods, outweigh the disadvantages with respect to geometry-based methods. Probably the most limited disadvantage is the enormous amount of data required to accurately represent these reflectance fields.

Dynamic scenes. Recently some research has focused on the acquisition of dynamic scenes [36, 66, 109]. Jones et al. [40] have experimented to extend the system of Wenger et al. [109] to relight performances with 4*D* incident illumination. Of all these methods, the system of Wenger et al. is probably the most successful, but it requires an expensive high-speed camera.

Future work in this direction would include the acquisition and synthesis of several atomic performances, resulting in a *virtual actor*. It is not clear how *atomic* performances can be combined to achieve complex performances, and still maintain a correct appearance under varying illumination. Another avenue for research in this area would be to find a sub-linear acquisition method capable of capturing and relighting a dynamic scene, and this without the use of specialized hardware.

Editing reflectance fields. Probably one of the major disadvantages of image-based techniques, that has received little attention until now, is the editing of acquired models. An image-based model is an exact (or at least a good approximation) model of a real-world object or scene. It is not clear how to "improve" such a model by, for example, changing the reflection properties of a specific part of the model.

In image-based rendering Wood et al. [115] deformed exitant light fields. Lawrence et al. [47] have made some initial steps in this direction, and presented a system for acquiring and editing non-parametric material representations for spatially varying BRDFs. In image-based relighting Wenger et al. [109] changed the reflectance properties of captured human faces. With the exception of Lawrence et al., none of these papers are specifically geared towards providing a complete system for editing.

Digitalization... The complete digitalization of real-world objects is the grand goal and I long for the day that a "digitalization" device is as commonly present in every household as a printer or a scanner is today. I hope that this research brings this dream a bit closer to reality.

A

Wavelets

In this appendix a concise overview of wavelets, and relevant mathematics is presented. It is not the intention to give a “practical guide” into wavelets, nor is it the intention to give an in depth “formal mathematical” overview. This appendix aims at a level of practicality and mathematical detail, such that it supports this dissertation. For a thorough and formal mathematical survey on wavelets, see the excellent book by Mallat [53], and Daubechies’ ten lectures on wavelets [18]. A more practical, computer graphics oriented, introduction into wavelets can be found in the book by Stollnitz et al. [98], and in the original SIGGRAPH ’96 course on which this book is based [90].

First, some important concepts and definitions from linear algebra, such as inner-products, linear bases, and dual bases, are reviewed (section A.1). Next, wavelets are intuitively introduced through the *Haar* wavelet (section A.2). In section A.3, a fast algorithm is discussed to perform a wavelet transform. This is subsequently generalized for arbitrary wavelets in section A.4. Next, we discuss how to generate and use two-dimensional wavelets (section A.5). Finally, some practical issues are detailed in section A.6.

A.1 Concepts and Definitions

An important concept in this dissertation is the notion of a **dot-product**, or **inner-product**. Given two vectors V and W of equal size l, the dot-product is defined by:

$$V \cdot W = \sum_{i=1}^{l} v_i w_i.$$

This can also be denoted by $\langle V | W \rangle$ or by $V^T W$. There are many interpretations possible of such an inner-product. In case both vectors are normalized[1], the dot-product equals the cosine of the angle between both vectors. Another view on inner-products is that they *project* a vector orthogonally onto another vector. Two vectors are considered to be linearly independent, or orthogonal to each other, if:

$$V \cdot W = 0.$$

[1] A vector is normalized if it has a unity length. Any vector V can be normalized by dividing it by its length: $\frac{V}{||V||_2}$.

If each vector has size l, then at most l **normalized** vectors $\mathbf{B}_{\cdot,i}$ can be defined that are linearly independent with respect to each other. By stacking these vectors in a matrix $\mathbf{B} = [\mathbf{B}_{\cdot,1} \,|\, ... \,|\, \mathbf{B}_{\cdot,l}\,]$, this orthonormality constraint can be denoted compactly as:

$$\mathbf{B}\mathbf{B}^T = I. \tag{A.1}$$

Such a set of vectors is called an **orthonormal basis** of the l-dimensional vector space, and completely defines this l-dimensional space. In other words, any vector V can be completely expressed in this basis $\mathbf{B}$, by projecting this vector onto each basis vector $\mathbf{B}_{\cdot,i}$:

$$V_{\mathbf{B}}{}^T = V^T\mathbf{B}, \tag{A.2}$$

or each element of $V_{\mathbf{B}}$ is determined by $(v_{\mathbf{B}})_i = V \cdot \mathbf{B}_{\cdot,i}$. This is also called the *decomposition* of a vector V into a basis $\mathbf{B}$. Furthermore, the basis vectors $\mathbf{B}_{\cdot,i}$ are also called basis functions. Given the decomposed vector $V_{\mathbf{B}}$, the original vector V can be composed again by using equation (A.1):

$$\begin{aligned} V^T &= V^T\left(\mathbf{B}\mathbf{B}^T\right) \\ &= \left(V^T\mathbf{B}\right)\mathbf{B}^T \\ &= V_{\mathbf{B}}{}^T\mathbf{B}^T. \end{aligned}$$

In other words, by doing an inverse transformation (note: $\mathbf{B}^{-1} = \mathbf{B}^T$), the original vector is reconstructed. The orthonormality condition of equation (A.1) is a very restrictive condition. An $l \times l$ matrix $\mathbf{B}$ defines a general (non-orthonormal) basis of an l-dimensional vector space, if and only if there exists an $l \times l$ matrix $\overline{\mathbf{B}}$, for which the following condition holds:

$$\mathbf{B}\overline{\mathbf{B}}^T = \overline{\mathbf{B}}\mathbf{B}^T = I.$$

In such a case, $\mathbf{B}$ is called the **primal** basis, and $\overline{\mathbf{B}}$ the **dual** basis. Again, a vector V can be decomposed in the primal basis: $V_{\mathbf{B}}{}^T = V^T\mathbf{B}$. Following a similar reasoning as before, the composition is defined by:

$$V^T = V_{\mathbf{B}}{}^T\overline{\mathbf{B}}^T.$$

Similarly, a vector V can also be decomposed in the dual basis: $V_{\overline{\mathbf{B}}}$. The composition can be done by: $V^T = V_{\overline{\mathbf{B}}}{}^T\mathbf{B}^T$.

Finally, an **orthogonal** basis is a basis for which:

$$\begin{aligned}\mathbf{B}_{\cdot,i} \cdot \mathbf{B}_{\cdot,j} &= ||\mathbf{B}_{\cdot,i}||_2^2 \qquad if\ (i = j)\\ &= 0, \qquad\qquad otherwise,\end{aligned}$$

for each $i, j \in \{1, ..., l\}$, and, $||\mathbf{B}_{\cdot,i}||_2 \neq 0$. In other words, the basis vectors $\mathbf{B}_{\cdot,i}$ are not normalized. The dual basis is defined by:

$$\overline{\mathbf{B}}_{\cdot,i} = \frac{\mathbf{B}_{\cdot,i}}{||\mathbf{B}_{\cdot,i}||_2^2}. \tag{A.3}$$

A.2 Haar Wavelet

In this section, we will first intuitively introduce the Haar wavelet. The Haar wavelet is the simplest of all wavelets, and particularly suited for this purpose. Next, the concepts introduced in the previous section, are used to present a more formal framework for wavelets.

Illustrative example. Consider the vector:

$$V = [8\,;\,2\,;\,1\,;\,5].$$

We will use this vector to illustrate the basis principles of the Haar wavelet. Now, compute the averages of each couple of coefficients (i.e., the first and the second, and, the third and the forth):

$$V_{avg} = [5\,;\,3],$$

or

$$(v_{avg})_i = \frac{v_{2i} + v_{2i+1}}{2}.$$

By averaging, some detail of the original vector V has been lost. When reconstructing the original vector V, this detail needs to be reintroduced. For example, for the first two coefficients (8 and 2) a detail coefficient of 3 is required in order to restore the original values from the average 5, since $5+3=8$, and $5-3=2$. The complete detail vector is:

$$V_{detail} = [3\,;\,-2],$$

or

$$(v_{detail})_i = v_{2i} - (v_{avg})_i.$$

We can repeat this procedure on the obtained averages V_{avg}, yielding a new average $V_{avg'}$ and detail $V_{detail'}$:

$$\begin{aligned} V_{avg'} &= [4], \\ V_{detail'} &= [1]. \end{aligned}$$

Define the vector V_ψ by stacking $V_{avg'}$, $V_{detail'}$, and, V_{detail} in a single vector:

$$V_\psi = [4;1;3;-2].$$

Note, that this can be done, for any vector that has a length l that is a power of 2. The number of times the average and detail are computed (number of recursion levels), equals the logarithm (with base 2) of the vector length: $\log_2 l$.

Haar wavelet basis transformation. The original vector V can be reconstructed from V_ψ, by recursively reconstructing the averages. This raises the question, if non-recursive transformations ψ, and, $\overline{\psi}$ can be found such that:

$$\begin{aligned} {V_\psi}^T &= V^T \psi, \\ V^T &= {V_\psi}^T \overline{\psi}^T. \end{aligned}$$

Indeed, such a basis transformation ψ (in this case a 4 × 4 matrix) exists:

$$\psi = \begin{bmatrix} +\frac{1}{4} & +\frac{1}{4} & +\frac{1}{2} & 0 \\ +\frac{1}{4} & +\frac{1}{4} & -\frac{1}{2} & 0 \\ +\frac{1}{4} & -\frac{1}{4} & 0 & +\frac{1}{2} \\ +\frac{1}{4} & -\frac{1}{4} & 0 & -\frac{1}{2} \end{bmatrix}.$$

This is an orthogonal matrix, and thus the dual transformation can be easily formed using equation (A.3). For the Haar wavelet this transformation can be easily generalized for arbitrary sized (square) transformations by introducing *scale* and *wavelet* functions, $\varphi^{(v,o)}$ and $\psi^{(v,o)}$ respectively:

$$\begin{aligned} (\varphi^{(v,o)})_i &= c(v)\,\varphi(Shift(v,o,i)), \\ (\psi^{(v,o)})_i &= c(v)\,\psi(Shift(v,o,i)), \end{aligned} \tag{A.4}$$

where $Shift(v,o,i) = (2^v \frac{i-1}{l} - o)$ ensures that the correct values are used given a wavelet level v, and offset [2] o from the continuous mother scale and mother wavelet functions, $\varphi(\mathbf{x})$ and $\psi(\mathbf{x})$:

$$\begin{aligned} \varphi(\mathbf{x}) &= +1 \qquad for\ 0 \leq \mathbf{x} < 1, \\ &= 0 \qquad otherwise, \end{aligned}$$

and,

$$\begin{aligned} \psi(\mathbf{x}) &= +1 \qquad for\ 0 \leq \mathbf{x} < \frac{1}{2}, \\ &= -1 \qquad for\ \frac{1}{2} \leq \mathbf{x} < 1, \\ &= 0 \qquad otherwise, \end{aligned}$$

and where $c(v)$ is a weighting function, which in this case is: $c(v) = \frac{2^v}{l}$. Finally, the offset o must be in the interval $[0, 2^v - 1]$. The 4×4 matrix ψ is now defined by:

$$\psi = \left[\varphi^{(0,0)} \,|\, \psi^{(0,0)} \,|\, \psi^{(1,0)} \,|\, \psi^{(1,1)} \right].$$

To obtain an orthonormal Haar wavelet basis, the weighting function $c(v)$ is set to $\frac{\sqrt{2^v}}{l}$. Also note that $[V_{avg} \,|\, V_{detail}] = V \left[\varphi^{(1,0)} \,|\, \varphi^{(1,1)} \,|\, \psi^{(1,0)} \,|\, \psi^{(1,1)} \right]$.

A.3 Fast Wavelet Transform

Computing a wavelet transform by blindly applying equation (A.2) requires, in case of a vector of length l, l inner-product computations. Each inner-product requires an operational count of $O(l)$. Thus, the total time complexity of a basis transformation is $O(l^2)$. For a number of basis functions the complete transformation can be done in less time. For example, the fast Fourier transform can transform a vector with a length a power of 2, in $O(l \log_2(l))$ operations.

Cascading filter banks. In this section, we will show that there exists a fast wavelet transform algorithm, that works with any wavelet, and has a time complexity of $O(l)$. Consider again the illustrative example of section A.2. The Haar wavelet transform was first computed by using a recursive algorithm, that computes in each recursion step the average per coefficient couple of the previous averages, and the detail

[2] Note that this formula is different than the ones found in existing literature. However, this formula is in essence identical, with the difference that we defined the wavelets not on the unit interval $[0,1]$, but directly on a vector of length l.

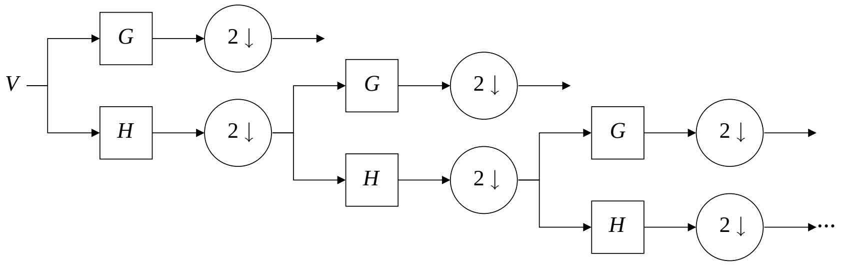


Figure A.1: A cascading filter bank. A signal is filtered by a low-pass filter H, and a high-pass filter G. For both filtered signals, half of the coefficients are thrown out. The filtering is repeated on the low-pass filtered result.

coefficients. At each step in this recursive algorithm, the averages and the corresponding details of half the coefficients are computed. This is repeated $\log_2(l)$ times. Thus the total complexity is:

$$\begin{aligned} complexity &= \sum_{i=0}^{\log_2(l)-1} \frac{l}{2^i}, \\ &= \sum_{i=0}^{\log_2(l)-1} \frac{2^{(\log_2(l)-1-i)}l}{2^{(\log_2(l)-1)}}, \\ &= \sum_{i=0}^{\log_2(l)-1} 2^i, \\ &= 2l-1, \end{aligned}$$

or in other words: $O(l)$.

This recursive algorithm is also called the *butterfly* algorithm, and can be generally described by a set of cascading filter banks. For this purpose, we define a high-pass filter G, and a low-pass filter H. For the Haar wavelet, these are:

$$\begin{aligned} h_{0:1} &= c_H\left[+\frac{1}{\sqrt{2}};+\frac{1}{\sqrt{2}}\right], \\ g_{0:1} &= c_G\left[+\frac{1}{\sqrt{2}};-\frac{1}{\sqrt{2}}\right]. \end{aligned}$$

The normalization constants c_H, and c_G have a similar role as the weighting function $c(v)$ in the previous section.

The working of the cascading filter bank is illustrated in figure A.1. At each cascade, the signal is filter by the low-pass, and high-pass filters. For both filtered signals,

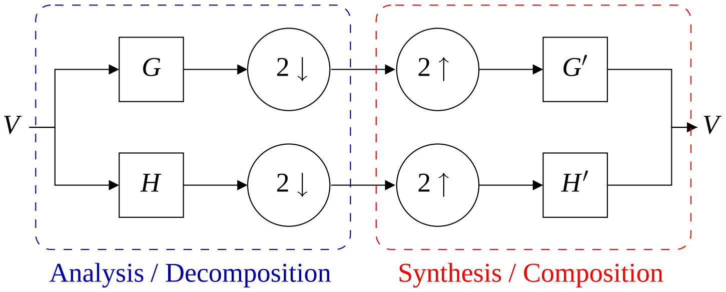


Figure A.2: The analysis filter bank, versus the synthesis filter bank. The analysis filter bank first filters the signal with a high-pass filter G and low-pass filter H, and then drops half of the signal. A synthesis filter bank, first inserts zero coefficients to double the size of the signal, and then performs a low-pass filtering H' and a high-pass filtering G'.

only half of the coefficients are kept. Next, the process is repeated on the low-pass filtered signal. This cascade continues until the length of the to-be-filtered signal is a single element. Such a cascading filter bank is also called an **analysis** filter bank. At the i-th cascading step, a vector analogous to the following transformation is computed:

$$V\left[\varphi^{(\log_2(l)-i)}\,|\ldots|\,\varphi^{(\log_2(l)-i,2^i-1)}\,|\,\psi^{(\log_2(l)-i,0)}\,|\ldots|\,\psi^{(\log_2(l)-1,\frac{l}{2}-1)}\right].$$

The inverse operation, the **synthesis** filter bank, is illustrated in figure A.2. A synthesis filter bank, works in a similar manner as the analysis filter bank. First, zero-coefficients are inserted in the signal. Next, the signal is interpolated by using low-pass, and high-pass synthesis filters H', and G' respectively. The synthesis filters for the Haar wavelet are defined as follows (note that the indices in these filters go from -1 to 0):

$$\begin{aligned} h'_{-1:0} &= \frac{1}{c_H}\left[+\frac{1}{\sqrt{2}};+\frac{1}{\sqrt{2}}\right], \\ g'_{-1:0} &= \frac{1}{c_G}\left[-\frac{1}{\sqrt{2}};+\frac{1}{\sqrt{2}}\right]. \end{aligned}$$

Example: analysis. We illustrate this filtering on the example of the previous section: a vector $V = [8;2;1;5]$. The normalization constants are set to $c_H = c_G = \frac{1}{\sqrt{2}}$, or $H = \left[+\frac{1}{2};+\frac{1}{2}\right]$, and $G = \left[+\frac{1}{2};-\frac{1}{2}\right]$. Then:

$$\begin{aligned} V * H &= \left[5;\frac{3}{2};3;\frac{13}{2}\right], \\ V * G &= \left[3;\frac{1}{2};-2;\frac{-3}{2}\right]. \end{aligned}$$

Note, that we assume that the signal repeats itself at the boundaries in order to compute the last element. Dropping all odd elements yields:

$$\begin{aligned}(V*H)\downarrow &= [5;3],\\ (V*G)\downarrow &= [3;-2].\end{aligned}$$

Note that:

$$[(V*LP)\downarrow;(V*G)\downarrow] = V\left[\varphi^{(1,0)}\,|\,\varphi^{(1,1)}\,|\,\psi^{(1,0)}\,|\,\psi^{(1,1)}\right].$$

Now, the filtering can be repeated on $(V*H)\downarrow$:

$$\begin{aligned}(V*H)\downarrow *H &= [4;4],\\ (V*H)\downarrow *G &= [1;-1].\end{aligned}$$

Dropping the odd elements, yields the transformed vector V_ψ.

Example: synthesis. The composition of the transformed signal V_ψ to the original signal V, can be done is a similar manner by using the synthesis filters:

$$\begin{aligned}(v_{\psi_{1:1}}\uparrow)*H' &= [4;4],\\ (v_{\psi_{2:2}}\uparrow)*G' &= [1;-1],\end{aligned}$$

summing yields:

$$V_{sum} = [5;3].$$

Note that this is equal to: $V_{sum} = V[\varphi^{(1,0)}\,|\,\varphi^{(1,1)}]$. Upsampling the subsequent level:

$$\begin{aligned}(V_{sum}\uparrow)*H' &= [5;5;3;3],\\ (v_{\psi_{3:4}}\uparrow)*G' &= [3;-3;-2;2].\end{aligned}$$

Adding results in the original vector $V = [8;2;1;5]$.

A.4 General Wavelets

In this section, other wavelets than the Haar wavelet are introduced. In many cases, the mother scale $\varphi(\mathbf{x})$ and mother wavelet $\psi(\mathbf{x})$ cannot be described by a simple analytical formula as with the Haar wavelet. In the previous section, however, we demonstrated that a wavelet transform is completely defined by its analysis, and synthesis filters. Most wavelets are specified, and even designed, by their filter coefficients.

Definitions. Before introducing other wavelets, a few characteristics of wavelets are defined.

- **Low-pass DC gain**: $DC_H = \left|\sum_i h_i\right|$,
- **High-pass DC gain**: $DC_G = \left|\sum_i g_i\right|$,
- **Low-pass Nyquist gain**: $NQ_H = \left|\sum_i (-1)^i h_i\right|$,
- **High-pass Nyquist gain**: $NQ_G = \left|\sum_i (-1)^i g_i\right|$,

For any wavelet the high-pass DC gain, DC_G, and the low-pass Nyquist gain, NQ_H, should be equal to zero.

A wavelet is characterized by the following properties:

- The **support** or footprint of a wavelet, is directly related to the filter lengths. The Haar wavelet is the shortest wavelet possible. Note that the length of each filter can be any number greater or equal to two. Furthermore, the low-pass filter length, and the high-pass filter length do not need to be equal. Infinite filter lengths are possible (although impractical). In general, a compact support is preferred to maximize locality in the spatial domain.
- The number of **vanishing moments**, also called the polynomial approximation order, is the lowest order of polynomial that cannot be represented completely by the scaling functions. All polynomials with an order less than the number of vanishing moments can be completely represented. For example, the scale function of the Haar wavelet can only completely represent piece-wise constant functions. These have a polynomial order of 0, thus the number of vanishing moments of the Haar wavelet equals 1.
- The **smoothness** of a wavelet is important, because it ensures an as smooth as possible approximation of the original signal.
- **Symmetric** wavelets are preferred above non-symmetric wavelets, because they have more signal extension possibilities. In section A.6, different strategies are discussed for signal extension at boundaries.
- Finally, **orthogonality** is a desired property.

Orthogonal Wavelets. Given a low-pass filter H of length n, a possible orthogonal high-pass filter can be computed by:

$$g_i = (-1)^i h_{n-i}.$$

The synthesis filters H', and G' can be computed from the analysis filter by:

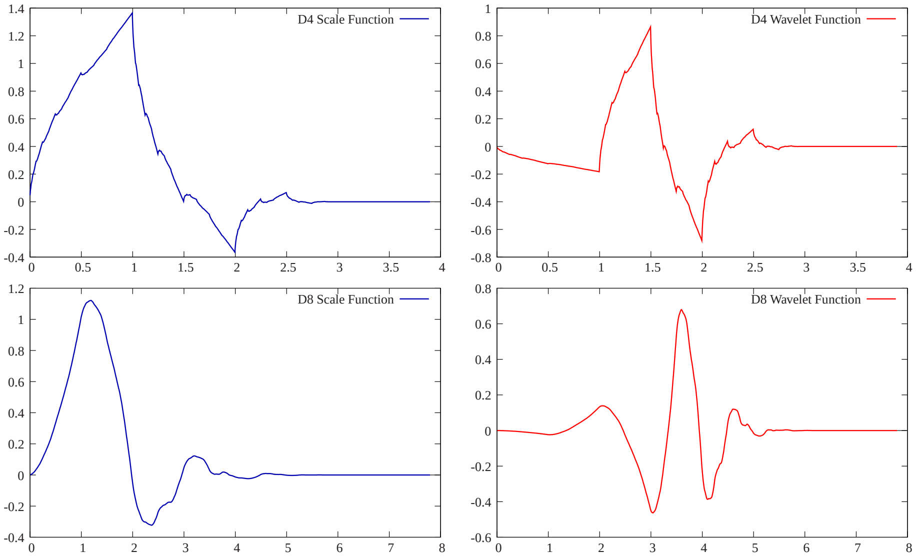


Figure A.3: Illustrations of two selected Daubechies orthogonal wavelets. The plotted wavelets are the *D*4, and *D*8 wavelet.

$$\begin{aligned} h'_i &= \alpha h_{-i}, \\ g'_i &= \alpha g_{-i}, \end{aligned}$$

where:

$$\alpha = \frac{\sqrt{2}}{||H||_2}.$$

If the analysis filter is defined on the interval $[0,n]$, then the synthesis filter is defined on the interval $[-n,0]$.

An orthogonal wavelet is orthonormal if and only if $DC_H = \sqrt{2}$, and $NQ_G = \sqrt{2}$. Thus, by introducing normalization parameters c_H and c_G, any orthogonal wavelet can be normalized.

A well-known example of a family of orthonormal wavelets is by Daubechies [17] (also in [18]). All filters have an even length, and are designed to have a maximum number of vanishing moments given their length (length $= 2 \times$ #vanishing moments). Furthermore, these wavelets are in general *not* smooth, nor symmetrical. These wavelets are denoted by their length: e.g., Daubechies 4 tap wavelet, or *D*4, has a wavelet filter size of 4 coefficients. The Haar wavelet is also contained in this family (i.e., *D*2). In figure A.3 two examples, Daubechies tap 4 and Daubechies tap

8 wavelets, are shown. For each wavelet, the scale and wavelet functions are shown. The low-pass synthesis filters (rounded to 5 significant digits) for these examples are:

$$\begin{aligned}
H_{D4} &= [0.48296\,;0.83652\,;0.22414\,;-0.12941],\\
H_{D8} &= [0.23038\,;0.71485\,;0.63088\,;-0.02798\,;\\
&\quad -0.18703\,;0.03084\,;0.03288\,;-0.01060].
\end{aligned}$$

For a derivation we refer the reader to [18], chapter 6. Also, in that work, an extended list of filter coefficients can be found in table 6.1, p 195.

Bi-orthogonal Wavelets. The orthonormal wavelets discussed above, have a number of excellent properties, such as a compact support versus the number of vanishing moments, and orthogonality. However, these wavelets are not smooth, nor symmetric. When designing a wavelet, additional properties can also be important for specific goals. In order to have more flexibility, the orthogonality constraint is relaxed to a bi-orthogonality constraint.

A wavelet is bi-orthogonal if the following conditions are met:

$$\begin{aligned}
\varphi^{(v,i)} \cdot \overline{\varphi}^{(v,j)} &= \delta_{i,j},\\
\psi^{(v,i)} \cdot \overline{\psi}^{(w,j)} &= \delta_{v,w}\delta_{i,j},\\
\psi^{(v,i)} \cdot \overline{\varphi}^{(v,j)} &= 0,\\
\varphi^{(v,i)} \cdot \overline{\psi}^{(v,j)} &= 0.
\end{aligned}$$

Or in terms of filter coefficients: H is orthogonal to G', and H' is orthogonal to G.

In this thesis we consider only two bi-orthogonal wavelets: the $5/3$ LeGall wavelet, and the $9/7$ Daubechies wavelet. Both are part of a larger family of bi-orthogonal wavelets that adheres to the following rules:

- These wavelets are symmetrical.
- Unlike their orthogonal counterparts, the analysis filter coefficients start with a negative index $-\frac{n-1}{2}$, where n is the length of the filter.
- Given the low-pass synthesis, and analysis filter, the high-pass filters are related through the "aliasing cancellation" conditions:

  $$\begin{gathered}
  g_i = \alpha(-1)^i h'_{-i},\\
  g'_i = \alpha(-1)^i h_{-i},
  \end{gathered}$$

  where α is a normalization constant given by:

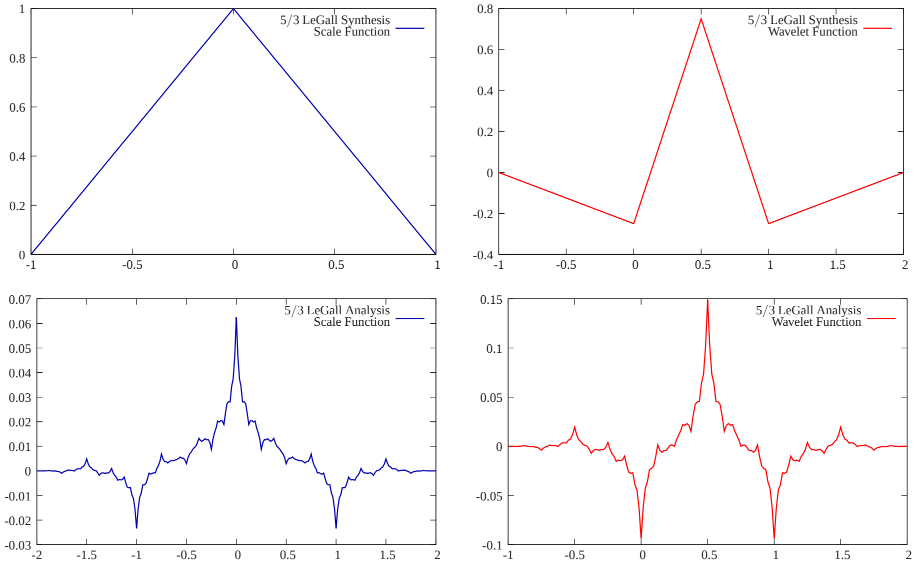


Figure A.4: A plot of the scale and wavelet functions for the 5/3 LeGall wavelet (synthesis and analysis).

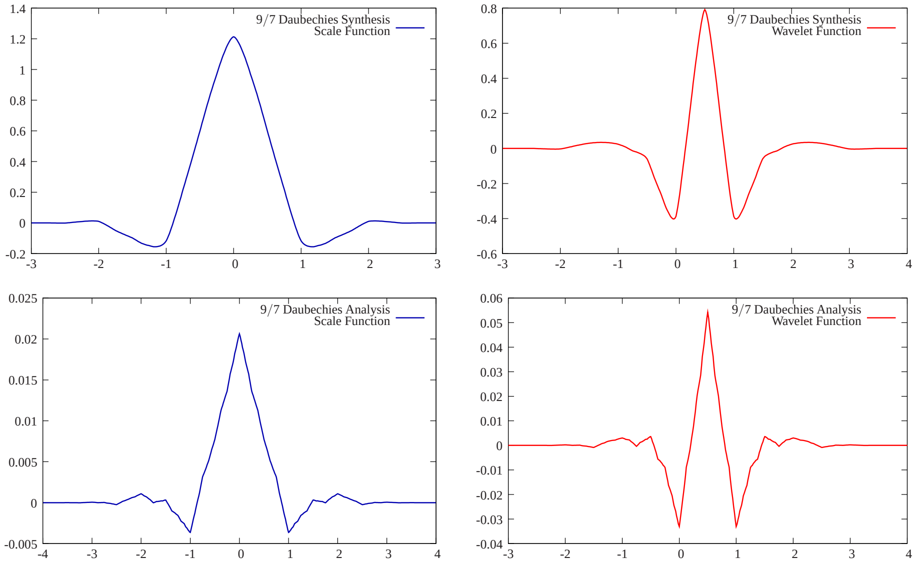


Figure A.5: A plot of the scale and wavelet functions for the 9/7 Daubechies wavelet (synthesis and analysis).

$$\alpha = \frac{2}{\left(\sum_i h_i\right)\left(\sum_i (-1)^i h'_i\right) + \left(\sum_i h'_i\right)\left(\sum_i (-1)^i h_i\right)}.$$

- Both (high-pass and low-pass) filters have either an even, or an odd length. In case of the two discussed wavelets, these are odd in length. Furthermore, both filters do not have an equal length.

In figure A.4, the $5/3$ LeGall wavelet (also called Integer $5/3$, or bi-orthogonal $5/3$ Daubechies) is shown. This wavelet has two vanishing moments, and its (analysis) scaling function is a linear B-Spline. This wavelet is especially designed for compression purposes. To ensure a smooth decompressed signal, the analysis filters are designed such that they are smooth, while maintaining an as small as possible footprint. The synthesis filters, however, are not smooth. The filter coefficients are:

$$\begin{aligned} H_{53} &= \left[-\frac{1}{8}; \frac{2}{8}; \frac{6}{8}; \frac{2}{8}; -\frac{1}{8}\right], \\ G_{53} &= \left[-\frac{1}{2}; 1; -\frac{1}{2}\right]. \end{aligned}$$

In figure A.5, the $9/7$ Daubechies wavelet is shown. This wavelet has four vanishing moments, and its (analysis) scaling function is a cubic B-Spline. Again, the analysis filters are designed to be as smooth as possible given the number of vanishing moments, and have a minimal support size. The filter coefficients are:

$$\begin{aligned} H_{97} &= [0.02675; -0.01686; -0.07822; 0.26686; 0.60295; \\ &\qquad 0.26686; -0.07822; -0.01686; 0.02675], \\ G_{97} &= [0.09127; -0.05754; -0.59127; 1.11509; \\ &\qquad -0.59127; -0.05754; 0.09127]. \end{aligned}$$

For a detailed comparison regarding both these wavelets, we refer the reader to [105]. For a more detailed mathematical background, see [18], chapter 8.

A.5 Two-dimensional Wavelets

Two-, and, multiple-dimensional wavelet basis functions are defined by combining multiple $1D$ wavelet basis functions. In this section we specifically target $2D$ wavelet basis functions, and discuss two construction methods.

Standard construction. The standard construction of $2D$ wavelet basis functions consists of all possible tensor products of one-dimensional basis functions. This is illustrated for the Haar wavelet in figure A.6. It is important to note that, a standard construction of orthonormal wavelets, yields a $2D$ orthonormal wavelet basis.

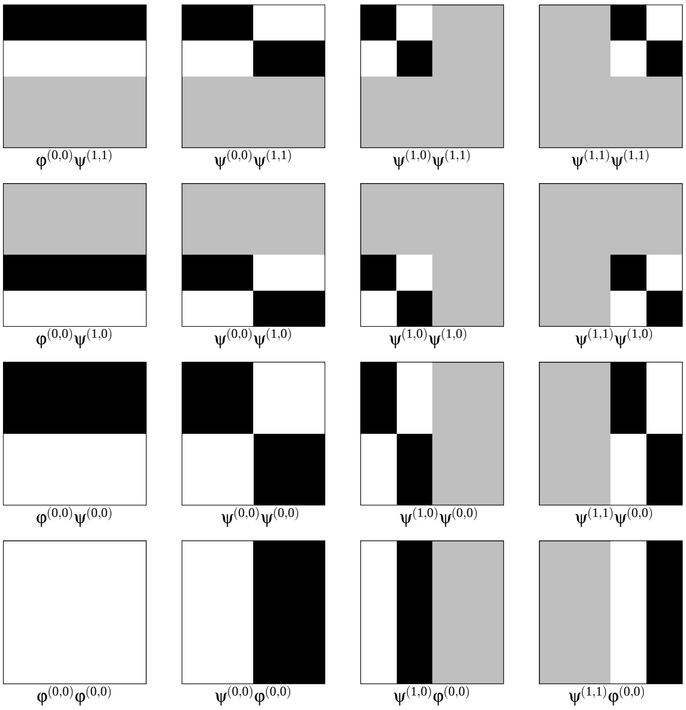

$\varphi^{(0,0)}\psi^{(1,1)}$
$\psi^{(0,0)}\psi^{(1,1)}$
$\psi^{(1,0)}\psi^{(1,1)}$
$\psi^{(1,1)}\psi^{(1,1)}$
$\varphi^{(0,0)}\psi^{(1,0)}$
$\psi^{(0,0)}\psi^{(1,0)}$
$\psi^{(1,0)}\psi^{(1,0)}$
$\psi^{(1,1)}\psi^{(1,0)}$
$\varphi^{(0,0)}\psi^{(0,0)}$
$\psi^{(0,0)}\psi^{(0,0)}$
$\psi^{(1,0)}\psi^{(0,0)}$
$\psi^{(1,1)}\psi^{(0,0)}$
$\varphi^{(0,0)}\varphi^{(0,0)}$
$\psi^{(0,0)}\varphi^{(0,0)}$
$\psi^{(1,0)}\varphi^{(0,0)}$
$\psi^{(1,1)}\varphi^{(0,0)}$

Figure A.6: A standard construction of 2*D* Haar wavelet basis functions (4 × 4 resolution).

Non-standard construction. The non-standard construction is probably the most commonly used 2*D* wavelet basis function construction method. This construction method starts by defining two-dimensional wavelet and scale functions:

$$\begin{aligned}
\varphi\varphi(\mathbf{x},\mathbf{y}) &= \varphi(\mathbf{x})\varphi(\mathbf{y}),\\
\varphi\psi(\mathbf{x},\mathbf{y}) &= \varphi(\mathbf{x})\psi(\mathbf{y}),\\
\psi\varphi(\mathbf{x},\mathbf{y}) &= \psi(\mathbf{x})\varphi(\mathbf{y}),\\
\psi\psi(\mathbf{x},\mathbf{y}) &= \psi(\mathbf{x})\psi(\mathbf{y}).
\end{aligned}$$

Equation (A.4), can now be naturally extended to 2*D*:

$$\begin{aligned}
\left(\varphi\varphi^{(v,o_x,o_y)}\right)_{j,i} &= c(v)\,\varphi\varphi(Shift(v,o_x,i),Shift(v,o_y,j)),\\
\left(\varphi\psi^{(v,o_x,o_y)}\right)_{j,i} &= c(v)\,\varphi\psi(Shift(v,o_x,i),Shift(v,o_y,j)),\\
\left(\psi\varphi^{(v,o_x,o_y)}\right)_{j,i} &= c(v)\,\psi\varphi(Shift(v,o_x,i),Shift(v,o_y,j)),\\
\left(\psi\psi^{(v,o_x,o_y)}\right)_{j,i} &= c(v)\,\psi\psi(Shift(v,o_x,i),Shift(v,o_y,j)),
\end{aligned}$$

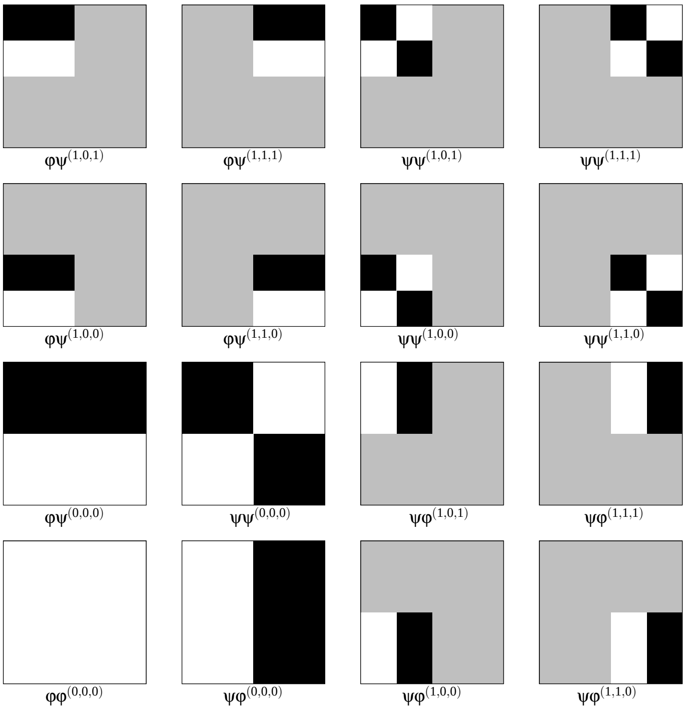


Figure A.7: A non-standard construction of 2*D* Haar wavelet basis functions (4×4 resolution).

where $Shift(\cdot,\cdot,\cdot)$ is defined similarly as before. The non-standard construction is illustrated in figure A.7 for the Haar wavelet. The non-standard construction can be seen as a single wavelet that is scaled and shifted depending on the level and inter-level location. Additionally, for each wavelet different “orientations”, or “directions” exist (i.e., 3 directions: $\varphi\psi$, $\psi\varphi$, and, $\psi\psi$). In this thesis, the non-standard construction is used for 2*D* wavelet basis functions.

A.6 Practical Issues

In this final section, a number of important practical issues are discussed. Until now, we assumed a cyclic signal, but more advanced edge handling techniques exist. Finally, an alternative to the butterfly algorithm is briefly discussed.

Edge extension. Except for the Haar wavelet, all other wavelets have a support size larger than two, and thus appropriate action has to be taken to handle filtering at the edges of the signal. The general solution is to **repeat** the signal:

$$v_{i+1} = v_{(i \; mod \; l)+1}.$$

Repeating the signal can introduce high frequency discontinuities in an otherwise smooth signal. For example, suppose the vector V is defined by: $V = [\frac{i}{10}]_i$. At any point the difference between subsequent elements is $\frac{1}{10}$. However, at the (repeating) edge this will be $\frac{l}{10}$, and depending on l, this can be significant.

In case the low-pass, and high-pass filters are *symmetrical* another edge extension method can be used: **mirroring** the signal. Depending on the filter length being *even*, or *odd*, a different mirroring scheme is used.

If both filters are odd, then the signal is extended to a cyclic signal of length $2l-2$:

$$\begin{aligned} v_{i+1} &= v_{(i \ mod \ (2l-2))+1} && if(i \ mod \ (2l-2))+1 \leq l, \\ &= v_{(2l-(i \ mod \ (2l-2)))+1} && otherwise, \end{aligned}$$

or in other words, the edge elements are mirrored on themselves, and thus not repeated. For example, a signal $[1;2;3;4]$ is mirrored to $[1;2;3;4;3;2]$, and then repeated.

If both filters are even, then the signal is extended to a cyclic signal of length $2l$:

$$\begin{aligned} v_{i+1} &= v_{(i \ mod \ 2l)+1} && if(i \ mod \ 2l)+1 \leq l, \\ &= v_{(2l-(i \ mod \ 2l)+1} && otherwise, \end{aligned}$$

or in other words, the whole signal is mirrored, and thus the edge elements are repeated. For instance, a signal $[1;2;3;4]$ is mirrored to $[1;2;3;4;4;3;2;1]$, and then repeated.

Both these approaches will not introduce additional high-frequency discontinuities at the mirrored, or repeated edges, and thus yield a sparser wavelet transformed signal.

Lifting. Lifting is an alternative algorithm to decompose and compose a signal into a wavelet basis. The lifting scheme was introduced by Sweldens [102], and is a fairly recent advancement in wavelets. In general, lifting is much easier to implement, and works two times as fast as the butterfly algorithm. A disadvantage of lifting is that a set of *lifting coefficients* are needed. The derivation of the lifting coefficients from filter coefficients is not trivial.

Figure A.8 illustrates the lifting scheme. The lifting scheme begins by splitting the signal in odd, and even coefficients. The next step is called the *predict step*, which tries to predict the odd values using the even values. The even values are left unchanged, while the odd elements are changed by the predicted values. Next, the odd elements are used to *update* the even elements. The even elements are the resulting low-pass filtered, and downsampled signal, while the odd elements are the high-pass filtered, and downsampled signal.

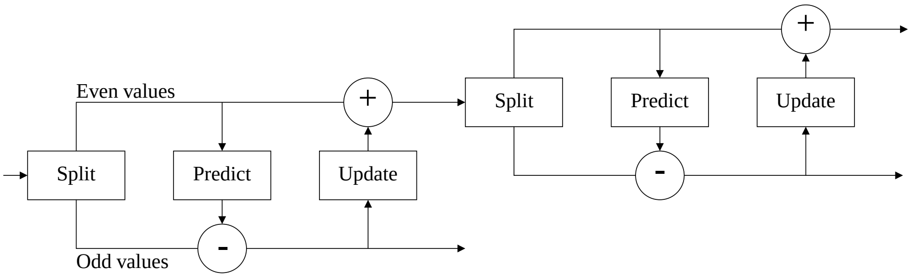


Figure A.8: The lifting scheme: forward wavelet transform.

The predict and update step are also described by one or more sets of coefficients. For the Haar wavelet these are:

$$\begin{aligned} P &= [1], \\ U &= [\frac{1}{2}]. \end{aligned}$$

To illustrate the lifting scheme, we use the example used in this appendix: a vector $V = [8;2;1;5]$. First, the signal is split in odd and even coefficients:

$$\begin{aligned} V_{odd} &= [8;1], \\ V_{even} &= [2;5]. \end{aligned}$$

Next the odd elements are updated by the prediction coefficients:

$$V'_{odd} = V_{odd} - P * V_{even},$$

or in the case of the Haar wavelet:

$$V'_{odd} = V_{odd} - V_{even}.$$

Thus:

$$V'_{odd} = [6;-4].$$

The update step, updates the even elements:

$$V'_{even} = V_{even} + U * V_{odd},$$

or specifically for the Haar wavelet:

$$V'_{even} \quad = \quad V_{even} + \frac{V_{odd}}{2}.$$

This yields:

$$V'_{even} \quad = \quad [5\,;3].$$

At this point, any normalization can be applied to the odd and even signal. In our case we use: $c_{even} = 1$, and $c_{odd} = \frac{1}{2}$:

$$\begin{aligned} V'_{odd} &= [3\,;-2], \\ V'_{even} &= [5\,;3]. \end{aligned}$$

Next this process is repeated on the obtained low-pass filter signal (i.e., V'_{even}), yielding:

$$\begin{aligned} V''_{odd} &= [1], \\ V''_{even} &= [4]. \end{aligned}$$

This results in the final transformed signal: $V_\psi = [3\,;1\,;-2\,;4]$. Note, that this is in a different order, which can be changed by a simple permutation to the same order as in section A.2. Synthesis is achieved by running the procedure above in a reverse order.

An interesting aspect of the lifting scheme is that everything can be done in place, whereas with filtering the signal needs to be copied to ensure a correct filtering. The reason why lifting can be done in place is because a new value only depends on subsequent values that are not yet changed during the current iteration.

The lifting coefficients for the $5/3$ LeGall wavelet are:

$$\begin{aligned} P &= \left[\frac{1}{2}\,;\frac{1}{2}\right], \\ U &= \left[\frac{1}{4}\,;\frac{1}{4}\right]. \end{aligned}$$

The lifting coefficients for the $9/7$ Daubechies wavelet are:

$$\begin{aligned} P^0 &= [1.58613\,;1.58613], \\ P^1 &= [-0.88291\,;-0.88291], \\ U^0 &= [-0.05298\,;-0.05298], \\ U^1 &= [0.44351\,;0.44351]. \end{aligned}$$

Note, that this wavelet has two sets of update and predict coefficients. A single lifting step, first uses the prediction coefficients P^0, and update coefficients U^0. Next, the same procedure is repeated on the obtained *odd* and *even* signals, but with the prediction coefficients P^1, and update coefficients U^1. Then, any normalization can be done, before going to the next lifting iteration step.

Publications

Articles in International Journals:

[P1] Vincent Masselus, **Pieter Peers**, Philip Dutré, and Yves D. Willems “Relighting with 4D Incident Light Fields” *ACM Transactions on Graphics (TOG) (Proceedings of ACM SIGGRAPH 2003)*, Volume 22 , Issue 3, pages 613–620, San Diego, USA, July 2003.

[P2] Muath Sabha, **Pieter Peers**, and Philip Dutré “Texture Synthesis using Exact Neighborhood Matching” *Computer Graphics Forum*, (in press), 2006.

[P3] **Pieter Peers**, Karl vom Berge, Wojciech Matusik, Ravi Ramamoorthi, Jason Lawrence, Szymon Rusinkiewicz, and Philip Dutré “A Compact Factored Representation of Heterogeneous Subsurface Scattering” *ACM Transactions on Graphics (TOG) (Proceedings of ACM SIGGRAPH 2006)*, (in press), Boston, USA, July 2006.

Articles in Proceedings of International Conferences:

[P4] **Pieter Peers** and Philip Dutré “Wavelet Environment Matting” *Rendering Tedchniques 2003 (Eurographics Symposium on Rendering 2003)*, pages 157–166, Leuven, Belgium, June 2003.

[P5] Vincent Masselus, **Pieter Peers**, Philip Dutré, and Yves D. Willems “Smooth Reconstruction and Compact Representation of Reflectance Functions for Image-based Relighting” *Rendering Techniques 2004 (Eurographics Symposium on Rendering 2004)*, pages 287–298, Nörrkoping, Sweden, June 2004.

[P6] Thomas Koninckx, **Pieter Peers**, Philip Dutré, and Luc Van Gool “Scene-adapted Structured Light” *IEEE International Conference on Computer Vision and Pattern Recognition (CVRP2005)*, San Diego, USA, June 20-25, 2005.

[P7] **Pieter Peers** and Philip Dutré “Inferring Reflectance Functions from Wavelet Noise” *Rendering Techniques 2005 (Eurographics Symposium on Rendering 2005)*, pages 173–182, Konstanz, Germany, June 2005.

Posters and Abstracts:

[8] **Pieter Peers** and Philip Dutré “Accurate Image Based Re-lighting through optimization” Poster presentation, *13th Eurographics Workshop on Rendering*, Pisa, Italy, June 2002.

[P9] **Pieter Peers** and Philip Dutré “Accurate Image Based Re-lighting through optimization” Technical Sketch, *SIGGRAPH 2002*, San Antonio, USA, July 2002.

[P10] Philip Dutré and **Pieter Peers** “Wavelet Environment Matting” Hierarchical Methods in Computer Graphics, Dagstuhl Seminar 03271, Dagstuhl, Germany, June 29–July 4, 2003.

Technical Reports:

[P11] **Pieter Peers** and Philip Dutré “Accurate Image Based Re-lighting through optimization” Technical Report CW336, Department of Computer Science, K.U. Leuven, April 2002. 17 pages.

[P12] **Pieter Peers**, Vincent Masselus, and Philip Dutré “Free-form Acquisition of Shape and Appearance” Technical Report CW403, Department of Computer Science, K.U.Leuven, February 2005. 18 pages.

[P13] **Pieter Peers** and Philip Dutré “Update Rules for a Weighted Non-negative FH*G Factorization” Technical Report CW440, Department of Computer Science, K.U.Leuven, April 2006. 7 pages.

Curriculum Vitae

Pieter Peers was born on June 24th, 1977 in Lommel, Belgium. In 2000 he received a Master's degree in Computer Science (Licentiaat Informatica) from the Katholieke Universiteit Leuven (K.U.Leuven) in Belgium. Since 2000 he is a member of the Computer Graphics Research Group at the Department of Computer Science of the Katholieke Universiteit Leuven. He started working as a Ph.D. student from October 2001 under the supervision of prof. Philip Dutré, and prof. Yves D. Willems. His research interests include image-based relighting, matting, novel acquisition methods, and the use of compression techniques in computer graphics. The results of his research are reported in this thesis and have appeared in various international peer reviewed journals and proceedings.

Bibliography

[1] Edward H. Adelson and James R. Bergen. The Plenoptic Function and the Elements of Early Vision. In *Computational Models of Visual Processing*, pages 3–20. MIT Press, 1991.

[2] Michael Ashikhmin and Peter Shirley. An Anisotropic Phong BRDF Model. *Journal of Graphics Tools: JGT*, 5(2):25–32, 2000.

[3] Richard Baraniuk. Optimal Tree Approximation using Wavelets. In *SPIE Technical Conference on Wavelet Applications in Signal Processing*, July 1999.

[4] Joan Batlle, El Mustapha Mouaddib, and Joaquim Salvi. A Survey: Recent Progress in Coded Structured Light as a Technique to Solve the Correspondence Problem. *Pattern Recognition*, 31(7):963–982, July 1998.

[5] Moshe Ben-Ezra. Segmentation with Invisible Keying Signal. In *2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, pages 568–575, June 2000.

[6] Roy S. Berns, Fransisco H. Imai, Peter D. Burns, and Di-Y. Tzeng. Multi-spectral-based Color Reproduction Research at the Munsell Color Science Labratory. *Electronic Imaging*, pages 14–25, 1998.

[7] Paul J. Besl. Active, Optical Range Imaging Sensors. *Machine Vision and Applications*, 1(2):127–152, 1988.

[8] James F. Blinn. Models of Light Reflection for Computer Synthesized Pictures. In *SIGGRAPH '77: Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, 1977.

[9] Robert W. Boyd. *Nonlinear Optics, Second Edition*. Accademic Press, 2002.

[10] Matthew Brand. Incremental Singular Value Decomposition of Uncertain Data with Missing Values. In *Computer Vision - ECCV 2002, 6th European Conference on Computer Vision Proceedings, Part I*, pages 707–720, 2002.

[11] Yung-Yu Chuang, Brain Curless, David H. Salesin, and Richard Szeliski. A Bayesian Approach to Digital Matting. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Vol II*, pages 264–271, 2001.

[12] Yung-Yu Chuang, Douglas E. Zongker, Joel Hindorff, Brian Curless, David H. Salesin, and Richard Szeliski. Environment Matting Extensions: Towards Higher Accuracy and Real-Time Capture. In *SIGGRAPH 2000, Computer Graphics Proceedings*, Annual Conference Series, 2000.

[13] Albert Cohen, Wolfgang Dahmen, Ingrid Daubechies, and Ronald DeVore. Tree Approximation and Optimal Encoding. *Applied Computer Harmonics Analysis*, 11(2):192–226, 2001.

[14] Rob L. Cook and Kenneth E. Torrance. A Reflectance Model for Computer Graphics. *ACM Transactions on Graphics*, 1(1):7–24, 1982.

[15] Brian Curless and Marc Levoy. Better Optical Triangulation Through Spacetime Analysis. In *IEEE International Conference on Computer Vision*, pages 987–994, 1995.

[16] J. Daniel, W. Gragg, L. Kaufman, and G. W. Stewart. Reorthogonalization and Stable Algorithms for Updating the Gram-Schmidt QR Factorization. *Mathematics of Computations*, pages 772–795, October 1976.

[17] Ingrid Daubechies. Orthonormal Bases of Compactly Supported Wavelets. *Communications of Pure Applied Mathematics*, 41:909–996, 1988.

[18] Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

[19] Paul Debevec. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 189–198, 1998.

[20] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the Reflectance Field of a Human Face. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 145–156, 2000.

[21] Paul Debevec, Andreas Wenger, Chris Tchou, Andrew Gardner, Jamie Waese, and Tim Hawkins. A Lighting Reproduction Approach to Live-action Compositing. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 547–556, 2002.

[22] Paul E. Debevec and Jitendra Malik. Recovering High Dynamic Range Radiance Maps from Photographs. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 369–378, 1997.

[23] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and Rendering Architecture from Photographs: a Hybrid Geometry- and Image-based Approach. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20, 1996.

[24] Ronald A. DeVore. Nonlinear Approximation. *Acta Numerica*, 7:51–150, 1998.

[25] Ronald A. DeVore, Bjorn Jawerth, and Bradley J. Lucier. Image Compression through Wavelet Transform Coding. *IEEE Transactions on Information Theory*, 38(2):719–746, 1992.

[26] Julie Dorsey, James Arvo, and Donald Greenberg. Interactive Design of Complex Time-Dependent Lighting. *IEEE Computer Graphics and Applications*, 15(2):26–36, 1995.

[27] Julie Dorsey, François Sillion, and Donald P. Greenberg. Design and Simulation of Opera Lighting and Projection Effects. In *SIGGRAPH'91 conference proceedings*, pages 41–50, 1991.

[28] Ron O. Dror, Thomas K. Leung, Edward H. Adelson, and Alan S. Willsky. Statistics of Real-World Illumination. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Vol I*, pages 164–171, 2001.

[29] Philip Dutré, Philippe Bekaert, and Kavita Bala. *Advanced Global Illumination*. A.K. Peters, 2003.

[30] Martin Fuchs, Volker Blanz, and Hans-Peter Seidel. Bayesian Relighting. In *EGSR '05: Proceedings of the 16th Eurographics workshop on Rendering*, pages 157–164, July 2005.

[31] Andrew Gardner, Chris Tchou, Tim Hawkins, and Paul Debevec. Linear Light Source Reflectometry. *ACM Transactions on Graphics*, 22(3):749–758, 2003.

[32] Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, 1995.

[33] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The Lumigraph. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996.

[34] Tim Hawkins, Jonathan Cohen, and Paul Debevec. A Photometric Approach to Digitizing Cultural Artifacts. In *VAST '01: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*, pages 333–342, 2001.

[35] Tim Hawkins, Per Einarsson, and Paul E. Debevec. A Dual Light Stage. In *EGSR '05: Proceedings of the 16th Eurographics Workshop on Rendering*, pages 91–98, 2005.

[36] Tim Hawkins, Andreas Wenger, Chris Tchou, Andrew Gardner, Fredrik Göransson, and Paul E. Debevec. Animatable Facial Reflectance Fields. In *EGSR '04: Proceedings of the 15th Eurographics workshop on Rendering*, pages 309–321, 2004.

[37] Xiao D. He, Kenneth E. Torrance, François X. Sillion, and Donald P. Greenberg. A Comprehensive Physical Model for Light Reflection. *SIGGRAPH Computer Graphics*, 25(4):175–186, 1991.

[38] Wolfgang Heidrich and Hans-Peter Seidel. Realistic, Hardware-accelerated Shading and Lighting. In *SIGGRAPH 1999 Conference Proceedings*, pages 171–178, 1999.

[39] David A. Huffman. A Method for the Construction of Minimum-Redundancy Codes. In *Proceedings of the I.R.E.*, pages 1098–1102, Sept 1952.

[40] Andrew Jones, Andrew Gardner, Mark Bolas, Ian McDowall, and Paul Debevec. Performance Geometry Capture for Spatially varying Relighting, SIGGRAPH 2005, Sktech.

[41] James T. Kajiya. Anisotropic Reflection Models. In *SIGGRAPH 85*, pages 15–21, 1985.

[42] James T. Kajiya. The Rendering Equation. In *Computer Graphics (Proceedings of SIGGRAPH 86) 20 (4)*, pages 143–150, 1986.

[43] Takeo Kanade, Andrew Gruss, and L. Carley. A Very Fast VLSI Rangefinder. In *IEEE International Conference on Robotics and Automation*, pages 1322–1329, 1991.

[44] Jan Kautz, Peter-Pike Sloan, and John Snyder. Fast, Arbitrary BRDF Shading for Low-frequency Lighting using Spherical Harmonics. In *EGWR '02: Proceedings of the 13th Eurographics workshop on Rendering*, pages 291–296, 2002.

[45] Melisa L. Koudelka, Sebastian Magda, Peter N. Belhumeur, and David Kriegman. Image-based Modeling and Rendering of Surfaces with Arbitrary BRDFs. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Vol II*, pages 568–575, 2001.

[46] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-Linear Approximation of Reflectance Functions. *Computer Graphics*, 31(Annual Conference Series):117–126, 1997.

[47] Jason Lawrence, Aner Ben-Artzi, Chris DeCoro, Wojciech Matusik, Hans-Peter Pfister, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Inverse Shade Trees for Non-Parametric Material Representation and Editing. *ACM Transactions on Graphics*, 2006.

[48] Seungyoung Lee, George Wolberg, and Sung Yong Shin. Scattered Data Interpolation with Multilevel B-Splines. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):228–244, 1997.

[49] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based Reconstruction of Spatial Appearance and Geometric Detail. *ACM Transactions on Graphics*, 22(2):234–257, 2003.

[50] Marc Levoy and Pat Hanrahan. Light Field Rendering. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996.

[51] Zhoulin Lin, Tien-Tsin Wong, and Heung-Yeung Shum. Relighting with the Reflected Irradiance field: Representation, Sampling and Reconstruction. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Vol I*, pages 561–567, December 2001.

[52] Céline Loscos, Marie-Claude Frasson, George Drettakis, Bruce Walter, Xavier Granier, and Pierre Poulin. Interactive Virtual Relighting and Remodeling of Real Scenes. In *EGWR '99 :Proceedings of the 10th Eurographics workshop on Rendering*, volume 10, pages 235–246, Jun 1999.

[53] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[54] Tom Malzbender, Dan Gelb, and Hans Wolters. Polynomial Texture Maps. In *SIGGRAPH 2001, Computer Graphics Proceedings*, Annual Conference Series, pages 519–528, 2001.

[55] Wai man Pang, Tien-Tsin Wong, and Pheng-Ann Heng. Estimating Light Vectors in Real Time. *IEEE Computer Graphics and Applications*, 24(3):36–43, 2004.

[56] Vincent Masselus. *A Practical Framework for Fixed Viewpoint Image-based Relighting (PhD thesis)*. Katholieke Universiteit Leuven, Belgium, 2004.

[57] Vincent Masselus, Philip Dutré, and Frederik Anrys. The Free-form Light Stage. In *EGWR '02: Proceedings of the 13th Eurographics workshop on Rendering*, pages 247–256, 2002.

[58] Vincent Masselus, Pieter Peers, Philip Dutré, and Yves D. Willems. Relighting with 4D Incident Light Fields. *ACM Transactions on Graphics*, 22(3):613–620, 2003.

[59] Vincent Masselus, Pieter Peers, Philip Dutré, and Yves D. Willems. Smooth Reconstruction and Compact Representation of Reflectance Functions for Image-based Relighting. In *EGSR '04: Proceedings of the 15th Eurographics workshop on Rendering*, pages 287–298, 2004.

[60] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J. Gortler, and Leonard McMillan. Image-based Visual Hulls. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 369–374, 2000.

[61] Wojciech Matusik, Matthew Loper, and Hanspeter Pfister. Progressively-Refined Reflectance Functions from Natural Illumination. In *EGSR '04: Proceedings of the 15th Eurographics Workshop on Rendering*, pages 299–308, 2004.

[62] Wojciech Matusik, Hanspeter Pfister, Addy Ngan, Paul Beardsley, Remo Ziegler, and Leonard McMillan. Image-based 3D Photography using Opacity Hulls. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 427–437, 2002.

[63] Wojciech Matusik, Hanspeter Pfister, Remo Ziegler, Addy Ngan, and Leonard McMillan. Acquisition and Rendering of Transparant and Refractive Objects. In *EGWR '02: Proceedings of the 13th Eurographics workshop on Rendering*, pages 267–277, 2002.

[64] Morgan McGuire, Wojciech Matusik, Hanspeter Pfister, John F. Hughes, and Frédo Durand. Defocus Video Matting. *ACM Transactions on Graphics*, 24(3):567–576, 2005.

[65] Ankit Mohan, Jack Tumblin, Bobby Bodenheimer, Cindy Grimm, and Reynold J. Bailey. Table-top Computed Lighting for Practical Digital Photography. In *EGSR '05: Proceedings of the 16th Eurographics workshop on Rendering*, pages 165–172, 2005.

[66] Francesc Moreno-Noguer, Shree K. Nayar, and Peter N. Belhumeur. Optimal Illumination for Image and Video Relighting. In *Proceeding of IEEE European Conference on Visual Media Production (CVMP)*, 2005.

[67] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-Frequency Shadows Using Non-Linear Wavelet Lighting Approximation. *ACM Transactions on Graphics*, 22(3):376–381, 2003.

[68] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Geometric Considerations and Nomenclature for Reflectance. Technical Report 161, National Bureau of Standards (US), 1977.

[69] Jeffry Nimeroff, Eero Simoncelli, Julie Dorsey, and Norman I. Badler. Rendering Spaces for Architectural Environments. *Presence, the Journal of Virtual Reality and Teleoperators*, 4(3), 1995.

[70] Jeffry S. Nimeroff, Eero Simoncelli, and Julie Dorsey. Efficient Re-rendering of Naturally Illuminated Environments. In *Fifth Eurographics Workshop on Rendering*, pages 359–373, 1994.

[71] Harry Nyquist. Certain Topics in Telegraph Transmission Theory. *Transactions of AIEE*, 47:617–644, 1928.

[72] Michael Oren and Shree K. Nayar. Generalization of Lambert's Reflectance Model. *Computer Graphics*, 28(Annual Conference Series):239–246, 1994.

[73] Pieter Peers and Philip Dutré. Accurate Image Based Re-lighting through Optimization. Technical report, Department of Computer Science, K.U.Leuven, April 2002.

[74] Pieter Peers and Philip Dutré. Wavelet Environment Matting. In *EGSR ’03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 157–166, 2003.

[75] Pieter Peers and Philip Dutré. Inferring Reflectance Functions from Wavelet Noise. In *EGSR ’05: Proceedings of the 16th Eurographics workshop on Rendering*, pages 173–182, 2005.

[76] Pieter Peers, Vincent Masselus, and Philip Dutré. Free-form Acquisition of Shape and Appearance. Technical report, Department of Computer Science, K.U.Leuven, February 2005.

[77] Hanspeter Pfister, Matthias Zwicker, Jeroen van Baar, and Markus Gross. Surfels: Surface Elements as Rendering Primitives. In *SIGGRAPH ’00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342, 2000.

[78] Bui Tuong Phong. Illumination for Computer Generated Pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[79] Thomas Porter and Tom Duff. Compositing Digital Images. In *Computer Graphics (SIGGRAPH ’84 Proceedings)*, volume 18, pages 253–259, July 1984.

[80] Pierre Poulin and Alain Fournier. A Model for Anisotropic Reflection. *Computer Graphics*, 24(4):273–282, 1990.

[81] Ravi Ramamoorthi and Pat Hanrahan. An Efficient Representation for Irradiance Environment Maps. In *SIGGRAPH 2001, Computer Graphics Proceedings*, Annual Conference Series, pages 497–500, 2001.

[82] Ravi Ramamoorthi, Melissa L. Koudelka, and Peter N. Belhumeur. A Fourier Theory for Cast Shadows. In *Computer Vision - ECCV 2004, 8th European Conference on Computer Vision Proceedings, Part I*, volume 3021, pages 146–162, 2004.

[83] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*. Morgan Kaufmann Publishers, December 2005.

[84] Howard Rheingold. *Virtual Reality*. Summit Books, 1991.

[85] Mark A. Ruzon and Carlo Tomasi. Alpha Estimation in Natural Images. In *2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, pages 18–25, 2000.

[86] Joaquim Salvi, Jordi Pagés, and Joan Batlle. Pattern Codification Strategies in Structured Light Systems. *Pattern Recognition*, 37(4):827–849, 2004.

[87] Yoav Y. Schechner, Shree K. Nayar, and Peter N. Belhumeur. A Theory of Multiplexed Illumination. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 808, 2003.

[88] Christophe Schlick. A Customizable Reflectance Model for Everyday Rendering. In *EGWR '93: Proceedings of the 4th Eurographics workshop on Rendering*, pages 73–84, 1993.

[89] Peter Schröder and Wim Sweldens. Spherical Wavelets: Efficiently Representing Functions of the Sphere. In *SIGGRAPH 95, Computer Graphics Proceedings*, Annual Conference Series, pages 161–172, 1995.

[90] Peter Schröder, Wim Sweldens, Michael Cohen, Tony DeRose, and David Salesin. Wavelets in Computer Graphics, SIGGRAPH 96 Course Notes, 1996.

[91] Pradeep Sen, Billy Chen, Gaurav Garg, Stephen R. Marschner, Mark Horowitz, Marc Levoy, and Hendrik P. A. Lensch. Dual Photography. *ACM Transactions on Graphics*, 24(3):745–755, 2005.

[92] Jonathan Shade, Steven Gortler, Li wei He, and Richard Szeliski. Layered Depth Images. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242, 1998.

[93] Claude E. Shannon. Communication in the Presence of Noise. *Proceedings of Institute of Radio Engineers*, 37(1):10–21, 1949.

[94] Heung-Yeung Shum and Sing Bing Kang. A Review of Image-based Rendering Techniques. In *IEEE/SPIE Visual Communications and Image Processing (VCIP)*, pages 2–13, 2000.

[95] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed Radiance Transfer for Real-time Rendering in Dynamic, Low-frequency Lighting Environments. *ACM Transactions on Graphics*, 21(3):527–536, July 2002.

[96] Alvy Ray Smith and James F. Blinn. Blue Screen Matting. *Computer Graphics*, 30(Annual Conference Series):259–268, 1996.

[97] Gilbert W. Stewart. *Matrix Algorithms: Volume 1: Basic Decompositions*. Society for Industrial and Applied Mathematics (SIAM), 1998.

[98] Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann Publishers, Inc., 1996.

[99] Maureen Stone. *A Field Guide to Digital Color*. A.K.Peters, 2003.

[100] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson Matting. *ACM Transactions on Graphics*, 23(3):315–321, 2004.

[101] Jian Sun, Yin Li, Sing-Bing Kang, and Heung-Yeung Shum. Flash Matting. *ACM Transaction on Graphics*, 2006.

[102] Wim Sweldens. The Lifting Scheme: a Construction of Second Generation Wavelets. *Siam Journal on Mathimatical Analysis*, 29(2), 1997.

[103] Kenneth E. Torrance and Ephraim M. Sparrow. Theory for Off-specular Reflection from Roughened Surfaces. *Journal of the Optical Society of America*, 57, 1967.

[104] Jonas Unger, Andreas Wenger, Tim Hawkins, Andrew Gardner, and Paul Debevec. Capturing and Rendering with Incident Light Fields. In *EGSR '03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 141–149, 2003.

[105] Michael Unser and Thierry Blu. Mathematical Properties of the JPEG2000 Wavelet Filters. *IEEE Transactions on Image Processing*, 12(9):1080–1090, September 2003.

[106] Z. Vidor. An Infrared Self-matting Process. In *Society of Motion Picture and Television Engineers 69*, pages 425–427, June 1969.

[107] Gregory J. Ward. Measuring and Modeling Anisotropic Reflection. *Journal of Computer Graphics*, 26(2):265–272, 1992.

[108] Terry Welch. A Technique for High-Performance Data Compression. *Computer*, June 1984.

[109] Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec. Performance Relighting and Reflectance Transformation with Time-multiplexed Illumination. *ACM Transactions on Graphics*, 24(3):756–764, 2005.

[110] Andreas Wenger, Tim Hawkins, and Paul Debevec. Optimizing Color Matching in a Lighting Reproduction System for Complex Subject and Illuminant Spectra. In *EGSR '03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 249–259, 2003.

[111] Yonatna Wexler, Andrew W. Fitzgibbon, and Andrew Zisserman. Image-based Environment Matting. In *EGWR '02: Proceedings of the 13th Eurographics workshop on Rendering*, 2002.

[112] Ian H. Witten, Radford M. Neal, and John G. Cleary. Arithmetic Coding for Data Compression. *Communications of the ACM*, 30(6):520–540, 1987.

[113] Tien-Tsin Wong, Pheng-Ann Heng, and Chi-Wing Fu. Interactive Relighting of Panoramas. *IEEE Computer Graphics and Applications*, 21(2):32–41, 2001.

[114] Tien-Tsin Wong, Pheng-Ann Heng, Siu-Hang Or, and Wai-Yin Ng. Image-based Rendering with Controllable Illumination. In *EGWR '97: Proceedings of the 8th Eurographics workshop on Rendering*, pages 13–22, June 1997.

[115] Daniel N. Wood, Daniel I. Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H. Salesin, and Werner Stuetzle. Surface Light Fields for 3D Photography. In *SIGGRAPH 2000*, *Computer Graphics Proceedings*, Annual Conference Series, pages 287–296, 2000.

[116] Cha Zhang and Tsuhan Chen. A Survey on Image-based Rendering - Representation, Sampling and Compression. *EURASIP Signal Processing: Image Communication*, 19:1–28, Jan 2004.

[117] Jiayuan Zhu and Yee-Hong Yang. Frequency-Based Environment Matting. In *12th Pacific Conference on Computer Graphics and Applications (PG 2004)*, pages 402–410, 2004.

[118] Jacob Ziv and Abraham Zempel. A Universal Algorithm for Sequential Data Compression. *IEEE Transactions on Information Theory*, May 1977.

[119] Douglas E. Zongker, Dawn M. Werner, Brian Curless, and David H. Salesin. Environment Matting and Compositing. In *SIGGRAPH 1999, Computer Graphics Proceedings*, Annual Conference Series, pages 205–214, 1999.