Dissertation Proposal

Greg Coombe
University of North Carolina

August 22, 2005

Contents

1 Introduction

Surface light fields and surface reflectance fields are image-based representations of lighting which are parameterized over geometry. Constructing these representations is a time-consuming and tedious process. The data sizes are quite large, often requiring multiple gigabytes to represent complex reflectance properties. The result can only be viewed after a lengthy post-process is complete, so it can be difficult to determine when the light field is sufficiently sampled. Often, uncertainty about the sampling density leads users to capture many more images than necessary in order to guarantee adequate coverage.

The goal of this work is a "casual capture" system which allows the user to interactively capture and view surface light fields and surface reflectance fields. "Casual capture" refers to two elements of our approach; no specially-constructed capture rigs or tracking devices, and a "human-in-the-loop" interactive feedback system. As each image is captured, it is incorporated into the representation in a streaming fashion and displayed to the user. In this way, the user receives direct feedback about the capture process, and can use this feedback to improve the sampling.

2 Thesis

The incremental construction of surface light fields and surface reflectance fields provides benefits such as accelerated capture, interactive previewing, and data-driven "human-in-the-loop" feedback.

3 Expected Contributions

In defense of this thesis statement, I intend to offer the following contributions:

Incremental capture and display of surface light fields A system for incrementally capturing, constructing, and rendering directionally-varying illumination by building a low rank linear approximation to the surface light field. This system is described in Coombe et al. [CHGL05].

Relighting A system for incrementally capturing, constructing and rendering 6D surface reflectance fields.

Interactive Previewing The surface light field and surface reflectance fields can be viewed interactively using graphics hardware as the model is being constructed. This was demonstrated for surface light fields in Coombe et al. [CHGL05].

User feedback A data-driven heuristic that highlights undersampled areas of the surface light field and directs the user towards effective camera views. This was demonstrated for surface light fields in Coombe et al. [CHGL05].

4 Background

A good overview of the state of the art in material modelling by image acquisition, and potentially inverse rendering, is provided by the recent Siggraph course on Material Modelling [RM02], and the Eurographics State of the Art Report on Acquisition, Synthesis and Rendering of Bidirectional Texture Functions [MMS+04]. Our system captures the exitant radiance of an object from images, and is based on BRDF capture systems [DvGNK99, LFTG97, MWL+99] and view dependent texture maps [LYS01].

Representation of this captured data is crucial for interactive rendering. Lensch [LKG+01] use the Lafortune [LFTG97] representation and clustered BRDFs from acquired data in order to create spatially-varying BRDFs. McAllister [MLH02] describes a device for scanning 6D spatially varying BRDFs and methods for fitting the data to a Lafortune representation. Gardner [GTHD03] describe a BRDF capture device that uses a linear light source (as opposed to a point source), which can also estimate surface normals and a height field.

A data-driven approach to surface light fields can be divided into parametric and non-parametric approaches. The parametric approach assumes a particular model for the lighting (such as the Lafortune [LFTG97] model used by McAllister [MLH02]. These models are incapable of representing the wide variety of objects that occur in real scenes, as observed in Hawkins et al [**?**].

The non-parametric approach uses the captured data to estimate the underlying function, and makes no assumptions about the reflectance. Thus non-parametric models are capable of representing a much larger class of surfaces, which accounts for their recent popularity in image-based modelling [CBCG02, ?, ?]. Our approach uses non-parametric models to represent surface light fields and surface reflectance fields.

Our work has a strong machine learning component and depends on some of the recent algorithms developed in the context of data mining [Bra03, Row97].

4.1 Surface Light Fields

Surface light fields [MRP98] parameterize the exitant radiance directly on the surface of the model. This results in a compact representation that enables the capture and display of complex view-dependent illumination of real-world objects. This category of approaches includes view-dependent texture mapping [DTM96, DYB98, ?], which can be implemented with very sparse and scattered samples, as well as regular parameterizations of radiance [LH96, GGSC96]. Wood et al. [WAA$^+$00] use a generalization of Vector Quantization and Principal Component Analysis to compress surface light fields, and introduce a 2-pass rendering algorithm that displays compressed light fields at interactive rates.

Surface light fields can be represented as the function $f(s, t, \theta_v, \phi_v)$. The variables s and t represented surface location on the mesh, and θ_v and ϕ_v represent view directions. This function can be discretized over surface patches and solid angles and represented as a matrix. The columns of this matrix are the camera views, and the rows are the surface locations. These matrices describe the exitant radiance at every point on the surface from every direction. Since storing these full data matrices would be impractical, several techniques have been developed to compress the data. Factorization approaches represent the 4D surface light field $f(s, t, \theta, \phi)$ as a sum of products of lower-dimensional functions

$$f(s, t, \theta, \phi) \approx \sum_{r=1}^{rank} g(s, t) h(\theta, \phi)$$

The number of terms r is the rank of the approximation. This factorization attempts to decouple the variation in surface texture from the variation in lighting. These functions can be constructed by using Principal Component Analysis [CBCG02, NSI01] or non-linear optimization [HMG03]. The function parameters can be stored in texture maps and rendered in real-time [CBCG02].

4.2 Surface Reflectance Fields

Surface Reflectance Fields [DHT+00] are a generalization of surface light fields. The surface light field can be represented as a 4D function $L(s, t, \theta_v, \phi_v)$ that represents directionally-varying exitant radiance for a fixed illumination. The surface reflectance field removes the restriction of fixed lighting by additionally representing the incident lighting directions $L(s, t, \theta_v, \phi_v, \theta_l, \phi_l)$.

There are several approaches to capturing and rendering surface reflectance fields. Debevec [**?**] acquire radiance samples of a human face using a custom-built light stage, and compute a reflectance function that can be used to generate images from novel locations, and under novel lighting conditions. A similar system was used to capture reflectance fields of cultural artifacts [**?**]. Malzbender [MGW01] fit acquired data to a bi-quadratic polynomial to estimate a reflectance field.

4.3 Online Methods

Most of the research in image-based modelling has focused on *batch-processing* systems. These systems process the set of images over multiple passes, and consequently require that the entire set of images be available. For detailed capture of light fields, this requires significant storage (around 10^6 data samples [HMG03]). In addition, incorporating additional images into these models requires recomputing the model from the beginning. Formulating surface light field construction as an *online processing* approach avoids these problems by incrementally constructing the model as the images become available. Matusik [MLP04] used this approach with a kd-tree basis system to progressively refine a radiance model from a fixed viewpoint. Schirmacher [SHS99] adaptively meshed the *uv* and *st* planes of a light field, and used an error metric along the triangle edges to determine the locations of new camera positions. Hillesland [HMG03] updated a non-linear solution in an online fashion, but required multiple passes over the data.

5 Plan of Action

5.1 Online LFM paper

Paper completed Spring 2005, Presented Summer 2005

Figure 1: *A heart figurine, a marble pestle, and a copper pitcher captured and rendered with our online system.*

Our 4D surface light field system [CHGL05] is an online system for incrementally capturing, constructing, and rendering directionally-varying illumination. It is based upon the technique of Light Field Mapping [CBCG02], which uses the Singular Value Decomposition to represent and render surface light fields in real-time. LFM enables a high level of compression and makes no assumptions about the underlying physical light model. Thus it can be used to capture a wide variety of surfaces. However, the SVD is a batch-processing approach, which means all of the data must be available at once.

Our system builds an incremental low rank linear approximation to the surface light field using a technique known as Online SVD [Bra03]. Each image is incorporated into the lighting model as it is captured, providing the user with real-time feedback. This feedback enables the user to preview the surface light field and direct the image acquisition towards undersampled regions. We also introduce a novel data-driven quality heuristic to highlight these areas. Examples of several models are shown in Figure 1.

In the paper we presented a structured method for dealing with incomplete data. Due to occlusion, many surface patches are only partially visible. Rather than discarding these data, we use the current linear approximation to "impute" these holes.

5.2 Weighted Least Squares Light Fields

Paper Submission Date: Fall 2005

The Online LFM work raised several important questions. One question was the representation of scattered data. Since the core of the Online LFM work was an incremental SVD of the data, this required us to have a fully-resampled set of data. This was sometimes difficult due to missing data from occlusion and meshing errors. We presented a solution to automatically "impute" the missing data, but this had some drawbacks of its own.

To avoid these problems with resampling and missing data, we are trying a different approach and treating the problem as one of *scattered data approximation*. This means that we store the surface light fields as a 2D function $L(\theta, \phi)$ at each surface point (s, t), and use scattered data approximation techniques such as Weighted Least Squares to interpolate. Scattered data approximation is used to construct compressed representations of data values at arbitrary locations (such as camera locations on a hemisphere or surface locations on a model). A more detailed description of the Weighted Least Squares method is presented in Section 7.

In Figures 2, 3, and 4, we show results from preliminary experiments that we have generated with a prototype Matlab implementation of Weighted Least Squares. The first two are from Dave McAllister's GiftWrap128 dataset, and the third is from the buddha dataset provided by Intel. These renderings were computed using 169 node points for each texel. At each node, a 4-term polynomial was fit to the data of the form

$$F = a_1 + a_2 x + a_3 y + a_4 xy$$

This means each color channel of the 64x64 image is 4x169, which is 1107558 bytes per channel using 32-bit floats (total RGB storage about 3.3MB). The uncompressed version of these pictures is 64x64x164x3x4 = 8.3MB, which is approximately compression ration of 2.5x. Obviously, the real models that we use will have significantly more texels, which will increase the compression.

This research does not require rebuilding the light capture stage from the earlier paper, as we can reuse some of the captured datasets. We also have several datasets provided by Intel.

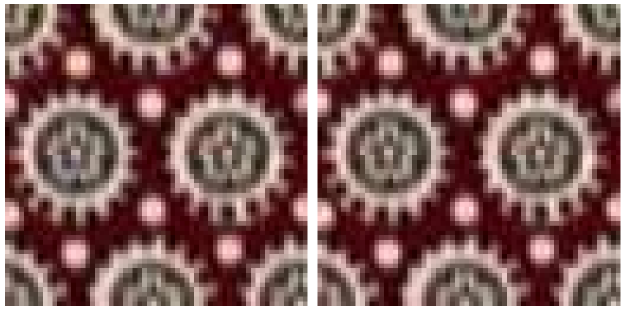

Figure 2: *GiftWrap128 SBRDF dataset. The reference version is on the left, and the Weighted Least Squares reconstruction is on the right.*

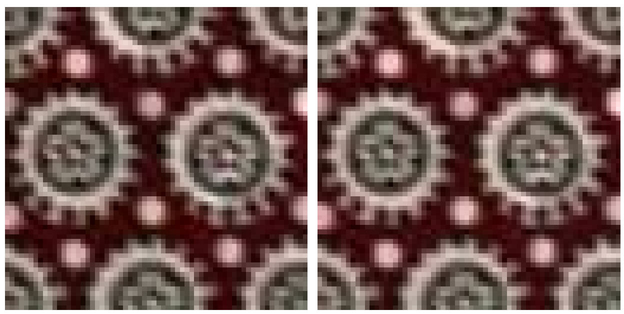

Figure 3: *Another view of the GiftWrap128 SBRDF dataset. As above, the reference version is on the left, and the Weighted Least Squares reconstruction is on the right.*

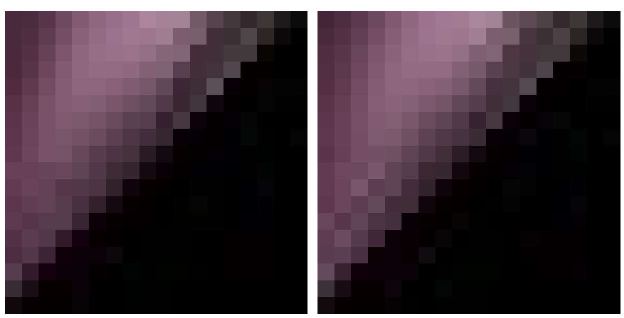

Figure 4: *A view of a single triangle from the buddha dataset. As above, the reference version is on the left, and the Weighted Least Squares reconstruction is on the right.*

5.3 Weighted Least Squares Reflectance Fields

Paper Submission Date: Spring 2006

Surface reflectance fields remove the restriction of fixed lighting, which increases the dimensionality of the problem from 4D to 6D. The real advantage of the scattered data approximation representation will become evident when the scope of the problem is expanded to include surface reflectance fields. Our attempts to use the Higher-Order (or Tensor) Singular Value Decomposition to represent surface reflectance fields were uniformly negative. Not only are the data sizes prohibitive, but the sampling requirements are even more stringent. We were also unable to develop an incremental approach to construct Higher-Order SVDs.

Extending the scattered data approximation technique to multiple dimensions is simply a matter of changing the dimensionality of the basis functions. This will allow us to reuse much of the structure that will be developed in the Weighted Least Squares Light Field paper. We will then be able to focus on what we view as the most serious drawback to reflectance fields, which is the capture process. Most surface reflectance capture algorithms require the construction of a lighting/camera rig to track both the light positions and the camera positions. This requires a fair amount of money as well as mechanical ability. We would like to simplify this process by using light probes to capture the incident lighting. A light probe can be thought of as a 2D plane in the 4D surface reflectance field. We may be able to use this 2D plane directly in the scattered data approximation, or we may importance sample the plane and use these points as weighted samples. Either way, by capturing a light probe image and then a series of camera images, we should be able to dramatically reduce the amount of effort required.

There are several 6D datasets available, including Dave McAllister's SBRDF datasets and a couple of real and synthetic datasets from Intel. We will most likely rebuild the light field stage that we built for the EGSR paper, so that we can capture surface reflectance fields.

5.4 Department Requirements

Teaching: Fall 2005
Orals: Spring 2006
Defense: Late Summer 2006

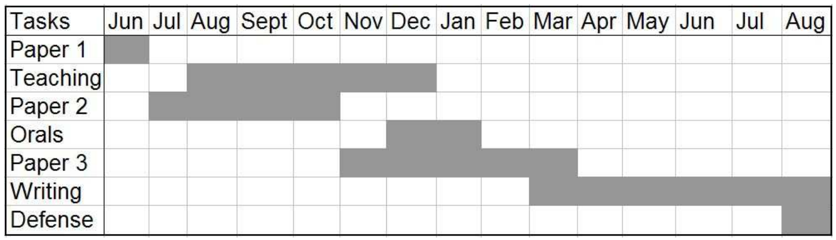

Figure 5: *My proposed schedule.*

6 Proposed Oral Examination Topics

Surface Light fields, Surface Modeling [CBCG02, HMG03, **?**, **?**, MLH02, LKG+01]

Image-based Modeling [DTM96, GGSC96]

Programming Graphics Hardware [**?**, **?**]

Data Mining, Dimensionality Reduction [**?**, Bra03]

7 Details about Methods

Least Squares methods are linear approximation techniques for scattered data. Given a set of N scalar samples $f_i \in \mathbb{R}$ at points $x_i \in \mathbb{R}^d$, we want a globally-defined function $f(x)$ that best approximates the f_i samples. The goal is to generate this function $f(x)$ such that the distance between the scalar data values f_i and the function evaluated at the data points $f(x_i)$ is as small as possible. This is written as:

$$min\{\sum_i \|f(x_i) - f_i\|\}$$

(Note: this discussion follows the notation of Nealen [**?**]). The coefficients of the polynomial are determined by minimizing this sum. Typically, $f(x)$ is a polynomial of degree m in d spatial dimensions. Thus $f(x)$ can be written as

$$f(x) = b(x)^T c$$

where $b(x) = [b_1(x)...b_k(x)]^T$ is the polynomial basis vector and $c = [c_1...c_k]$ is the unknown coefficient vector. The set of basis functions $b(t)$ is chosen that best approximates the properties of the data. In our case, we chose the 2D quadratic basis set

$$b(t) = [1 \;\; x \;\; y \;\; x^2 \;\; xy \;\; y^2]$$

To determine the coefficient vector c, the minimization problem is solved by taking partial derivatives, setting them to zero, and solving the resulting system of linear equations. After rearranging the terms, the solution is:

$$c = [\sum_i b(x_i)b(x_i)^T]^{-1} \sum_i b(x_i) f_i$$

This can be solved using any common matrix inversion package such as BLAS [**?**] or TGT [**?**]. The size of the matrix to be inverted depends upon the dimensionality d of the data and the degree k of the polynomial.

The function is reconstructed by applying the coefficients to the basis set

$$f(x) = b(x)^T c$$

7.1 Weighted Least Squares

One of the problems with Least-Squares fitting is that every point in the solution influences every other point. This global complexity makes it difficult for large point sets. We would prefer a local method which constructs an approximation that considers points nearby as more important than points far away. This can be accomplished by adding a distance-weighting term $\theta(d)$ to the Least Squares minimization. We are now trying to minimize the function

$$min\{\sum_i \theta(\|x - x_i\|)\|f(x_i) - f_i\|\}$$

Now, instead of evaluating a single global approximation for all of the data points, we pick a set of node points $\bar{x}$ and evaluate the distance-weighted contribution of each data point x_i to these node points. This means that the coefficients are now a function of $\bar{x}$, and only defined locally around this point. A set of piecewise quadratic local approximations are computed at each of these node points.

$$c(\bar{x}) = [\sum_i \theta(\|\bar{x} - x_i\|) b(x_i) b(x_i)^T]^{-1} \sum_i \theta(\|\bar{x} - x_i\|) b(x_i) f_i$$

To reconstruct a global approximation from this set of local approximations, we first determine the m nearby local approximations that overlap this point. Unfortunately, we cannot just simply weight these functions by distance and add them together, since these weights may not sum to 1. To get the proper weighting of the local approximations, we use a technique known as *Partition of Unity* [**?**], which allows us to extend the local approximations to cover the entire domain. A new set of weights ϕ_j are computed by considering all of the m local approximations that overlap this point

$$\phi_j(x) = \frac{\theta_j(x)}{\sum_{k=1}^m \theta_k(x)}$$

The global approximation of this function is computed by summing the weighted local approximations.

$$f(x) = \sum_{j=1}^m \phi_j(x) b(x)^T c(\bar{x}_j)$$

This representation allows us to place the node points $\bar{x}$ at the most effective locations. This maps well to a GPU-rendering framework since the points can be regularly-spaced across the domain. In general, the GPU is most efficient when the node points are regularly-spaced, as this minimizes special cases and maximizes parallelism. The data structure and camera capture are managed on the CPU and function reconstruction and rendering is handled by the GPU.

7.1.1 Distance Function

Common choices for $\theta(d)$ are the Wendland function [Wen95]

$$\theta(d) = (1 - d/h)^4 (4d/h + 1)$$

and the interpolation function

$$\theta(d) = 1 + 2(d/h)^3 - 3(d/h)^2$$

Note that both of these functions include a term h, which modifies the support of the distance function. These functions are illustrated in Figure 6.

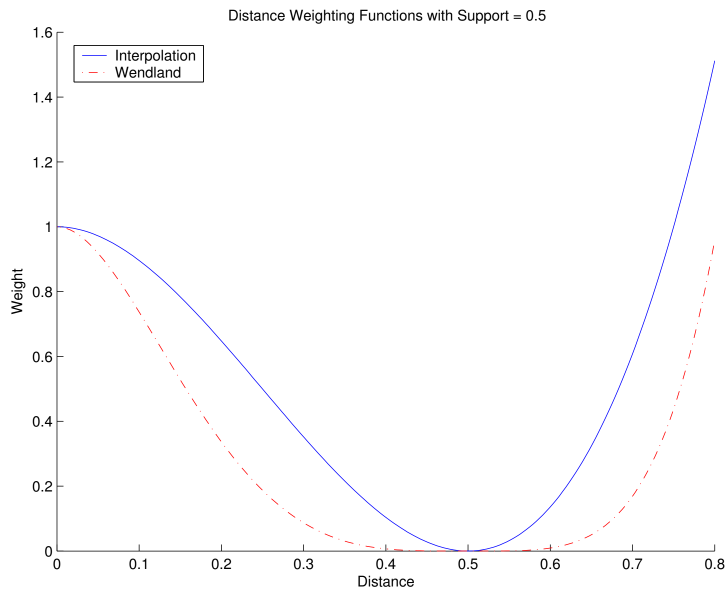

Figure 6: *Two possible distance-weighting functions, with a variable support. Both functions are 1 at the center, and taper off to zero as the distance increases. While they are well-behaved within the support interval, they must be clamped outside this interval to avoid spurious weighting.*

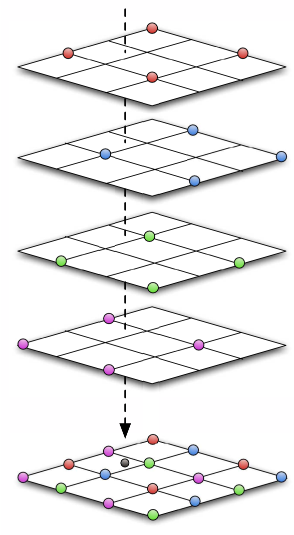

Figure 7: *The staggered grid gives us a higher resolution.*

7.1.2 Staggered Grids

We can also use this approximation to minimize discontinuities between patches on the surface by incorporating data points from adjacent surface patches. This requires adjusting the distance-weighting function to incorporate these values. We can also go one step further and use this framework to get a higher-resolution approximation by staggering the node locations at adjacent patches. This increases the effective resolution of the surface patches without increasing the storage costs.

This is illustrated in Figure 7.

References

[Bra03] Matthew Brand. Fast online svd revisions for lightweight recommender systems. In *SIAM International Conference on Data Mining*, 2003.

[CBCG02] Wei-Chao Chen, Jean-Yves Bouguet, Michael Chu, and Radek Grzeszczuk. Light field mapping: Efficient representation and hardware rendering of surface light fields. In *SIGGRAPH*, 2002.

[CHGL05] Greg Coombe, Chad Hantak, Radek Grzezschuk, and Anselmo Lastra. Online construction of surface lightfields. In *Eurographics Rendering Symposium*, 2005.

[DHT+00] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *SIGGRAPH*, 2000.

[DTM96] Paul E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH*, pages 11–20, 1996.

[DvGNK99] K.J. Dana, B. van Ginneken, S.K. Nayar, and J. J. Koenderink. Reflectance and texture of real world surfaces. In *ACM TOG*, 1999.

[DYB98] Paul E. Debevec, Yizhou Yu, and George D. Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. In *Eurographics Rendering Workshop*, pages 105–116, 1998.

[GGSC96] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The Lumigraph. In *SIGGRAPH*, pages 43–54, 1996.

[GTHD03] A. Gardner, C. Tchou, T. Hawkins, and P. Debevec. Linear light source reflectometry. In *SIGGRAPH*, 2003.

[HMG03] Karl Hillesland, Sergey Molinov, and Radek Grzeszczuk. Nonlinear optimization framework for image-based modeling on programmable graphics hardware. In *SIGGRAPH*, 2003.

[LFTG97] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear approximation of reflectance functions. In *SIGGRAPH*, pages 117–126, 1997.

[LH96] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH*, 1996.

[LKG+01] H. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H. Seidel. Image-based reconstruction of spatially varying materials. In *Eurographics Rendering Workshop*, 2001.

[LYS01] Xinguo Liu, Yizhou Yu, and Heung-Yeung Shum. Synthesizing bidirectional texture functions for real-world surfaces. In *SIGGRAPH*, 2001.

[MGW01] Tom Malzbender, Dan Gelb, and Hans Wolters. Polynomial texture maps. In *SIGGRAPH*, 2001.

[MLH02] D. McAllister, A. Lastra, and W. Heidrich. Efficient rendering of spatial bi-directional reflectance distribution functions. In *Graphics Hardware*, 2002.

[MLP04] Wojciech Matusik, Matthew Loper, and Hanspeter Pfister. Progressively-refined reflectance functions from natural illumination. In *Eurographics Symposium on Rendering*, 2004.

[MMS+04] G. Mueller, J. Meseth, M. Sattler, R. Sarlette, and R. Klein. Acquisition, synthesis and rendering of bidirectional texture functions. In *Eurographics*. State of the Art Reports, 2004.

[MRP98] Gavin S. P. Miller, Steven M. Rubin, and Dulce Ponceleon. Lazy decompression of surface light fields for precomputed global illumination. In *Eurographics Workshop on Rendering*, 1998.

[MWL+99] Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, Kenneth E. Torrance, and Donald P. Greenberg. Image-based BRDF measurement including human skin. In *Eurographics Workshop on Rendering*, 1999.

[NSI01] Ko Nishino, Yoichi Sato, and Katsushi Ikeuchi. Eigen-texture method: Appearance compression and synthesis based on a 3d model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1257–1265, 2001.

[RM02] Ravi Ramamoorthi and Steve Marschner. Acquiring material models using inverse rendering. In *SIGGRAPH Course 39*, 2002.

[Row97] Sam Roweis. EM Algorithms for PCA and SPCA. In *Neural Information Processing Systems*, pages 626–632, 1997.

[SHS99] Hartmut Schirmacher, Wolfgang Heidrich, and Hans-Peter Seidel. Adaptive acquisition of lumigraphs from synthetic scenes.

In *Computer Graphics Forum*, volume 18(3), pages 151–160, 1999.

[WAA+00] Daniel Wood, Daniel Azuma, Wyvern Aldinger, Brian Curless, Tom Duchamp, David Salesin, and Werner Stuetzle. Surface light fields for 3d photography. In *SIGGRAPH*, 2000.

[Wen95] Holger Wendland. Piecewise polynomial,positive definite and compactly supported radial basis functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396, 1995.